Electrical Power and Energy Systems 33 (2011) 115–123

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch

Rajesh Kumar *, Devendra Sharma, Abhinav Sadu

Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur 302 017, India

ARTICLE INFO

Article history:
Received 25 January 2009
Received in revised form 9 December 2009
Accepted 7 June 2010

Keywords:
Economic power dispatch
PSO
Valve-point effect
Multi-agent system

ABSTRACT

This paper presents a new multi-agent based hybrid particle swarm optimization technique (HMAPSO) applied to the economic power dispatch. The earlier PSO suffers from tuning of variables, randomness and uniqueness of solution. The algorithm integrates the deterministic search, the Multi-agent system (MAS), the particle swarm optimization (PSO) algorithm and the bee decision-making process. Thus making use of deterministic search, multi-agent and bee PSO, the HMAPSO realizes the purpose of optimization. The economic power dispatch problem is a non-linear constrained optimization problem. Classical optimization techniques like direct search and gradient methods fails to give the global optimum solution. Other Evolutionary algorithms provide only a good enough solution. To show the capability, the proposed algorithm is applied to two cases 13 and 40 generators, respectively. The results show that this algorithm is more accurate and robust in finding the global optimum than its counterparts.

1. Introduction

Economic power dispatch (EPD) is the scheduling of the committed generating unit outputs so as to meet the load demand at minimum operating costs while satisfying all units and system equality and inequality constraints. The main aim in the economic dispatch problem is to minimize the total cost of generating real power (production cost) at various stations while satisfying the loads and the losses in the transmission links [1,2]. EPD is thus one of the most important problems to be solved in the operation of power system. Since modern unit's input–output characteristics are highly non-linear due to valve-point loading, multiple-fuel effects and other constraints, a continuous search for better solver is going on [3–5].

A lot of classical methods have been developed and are being used for optimization problem. Golden section search, Fibonacci search, Newton's method and Secant method are some one dimension search method. Gradient methods, Newton's method, conjugate direction method and neural networks are commonly used for unconstrained optimization [2]. These methods are problem specific and use gradients. Consequently they are applicable to a much smaller classes of optimization problem.

A genetic algorithm (GA) is a probabilistic search technique that has its roots in the principles of genetics. It gives more emphasis on natural selection of surviving species and process of reproduction of new offspring. The algorithm works on process of mutation and crossover to create new population [6]. Since its conception, genetic algorithm has been used widely as a tool in computer programming, artificial intelligence and optimization.

Mimicking the behavior of intelligence available in various swarms a new intelligence comes into existence which is known as swarm intelligence (SI). Swarm intelligence is artificial intelligence which based on the collective behavior of decentralized, self-organized systems which mimics natural behavior of organisms. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local interactions between such agents lead to the emergence of complex global behavior [7]. A natural example of SI includes ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. Various algorithms derive from SI are Ant Colony Optimization (ACO), GA and particle swarm optimization (PSO) [6–8].

Particle swarm optimization (PSO) algorithm is based on social behavior of groups like flocking of birds or schooling of fish. It is a stochastic, population-based evolutionary computer algorithm for problem solving. It is a kind of swarm intelligence that predicts each individual solution as "particles" which evolve or change their positions with time. Each particle modifies its position in search space in accordance with its own experience and also that of neighbouring particle by remembering the best position visited by itself and its neighbours, then calculating local and global positions. These techniques are free from use of

* Corresponding author. Tel.: +91 141 2713372; fax: +91 141 2529092.
E-mail addresses: rk.rlab@gmail.com (R. Kumar), devendra.sharma11@gmail.com (D. Sharma), abhinavsadu@gmail.com (A. Sadu).

0142-0615/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijepes.2010.06.021

gradients hence can be applicable to a wider class of optimization problems [8,9].

The bees algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution for food as well as next site selection [10]. This algorithm performs a kind of neighborhood search combined with random search and can be used for both combinatorial optimization and functional optimization. Bee Colony Optimization (BCO), Bee System (BS) algorithms are some of the examples where algorithms are based on *Waggle dance* perform by scouts' bee to inform other foraging bees about the nectar site [11].

The practical EPD problems with valve-point effects is represented as a nonsmooth optimization problem having complex and non-convex features with heavy equality and inequality constraints [2]. This kind of optimization problem is hard, if not impossible, to solve using traditionally deterministic optimization algorithms. Recently, as an alternative to the conventional mathematical approaches, modern stochastic optimization techniques, evolutionary algorithms, Tabu search, neural networks, genetic algorithms, particle swarm optimization and other heuristic approaches algorithms have been given much attention by many researchers due to their ability to find potential solutions [2–6,13].

In this paper, we use our own developed a new algorithm, which is hybrid version on PSO which mimics its search algorithm from PSO and modify Nelder–Mead method [12] to find optimal solution. The decision making technique is mimicked from Bee decision-making process. The decision-making process is based on the algorithm used by bees for finding a suitable place for establishing new colony. The experimental results show the robustness and accuracy of hybrid PSO over genetic algorithm and PSO. Due to its hybrid nature this algorithm provides only deterministic solutions. Making use of these agent–agent interactions and evolution mechanism of PSO in a lattice-like environment, the proposed method can find high-quality solutions reliably with the faster convergence characteristics in a reasonably good computation time.

This paper is organized as follows. The hybrid algorithm is comprises of two parts search algorithm and other as decision-making process. The Section 2 details the economic dispatch problem formulation with valve-point effect. The Section 3 details the standard PSO and the related issues about accuracy and convergence to optimal solutions. Section 4 describes the basic requirements of MAS. The development and working of the Hybrid PSO is elaborated in the Section 5. The decision-making process in the honey bees make them an interesting swarm research area to work. Section 5 also discusses the decision making method used by the bees in the proposed algorithm. Section 6 discusses simulation and experimental results made on some standard test systems and draws inferences on the convergence characteristics from the results obtained. Finally, Section 7 concludes the paper.

2. Economic dispatch problem formulation

2.1. Basic economic dispatch formulation

The economic dispatch problem is to simultaneously minimize the overall cost rate and meet the load demand of a power system while satisfying an equality and inequality constraints [2,13]. Assuming the power system includes N generating units. The aim of economic power dispatch is to determine the optimal share of load demand for each unit in the range of 3–5 min. Generally, the economic power dispatch problem can be expressed as minimizing the cost of production of the real power which is given by objective function F_T

$$F_T = \sum_{i=1}^{n} F_i(P_i) \tag{1}$$

which is subjected to the constraints of equality in real and reactive power balance

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{2}$$

where a_i, b_i and c_i are the cost coefficients of the ith generator and N is the number of generators committed to the operating system. P_i is the power output of the ith generator.

2.1.1. Real power balance equation

For power balance, an equality constraint should be satisfied. The generation-demand balance including losses is given by the following equation

$$\sum_{i=1}^{N} P_i - P_l - P_d = 0 \tag{3}$$

where P_d is the total system demand and is the total line loss. However, in the case study presented here,we disregarded the transmission losses (i.e. $P_l = 0$).

2.1.2. Minimum and maximum power limits

Generation output of each generator should lie between maximum and minimum limits. The inequalities of real power limits on the generator output are:

$$P_{\min,i} \leqslant P_i \leqslant P_{\max,i} \quad \text{where} \quad i = 1, 2, \ldots, N \tag{4}$$

where $P_{\min i}$ and $P_{\max i}$ are the minimum and maximum real power limits of ith generator output in the system.

2.2. Valve-point effects

The generator costs are usually approximated using quadratic functions. However, it is more practical to consider the valve-point loading for fossil–fuel-based plants. In this context, a cost function is obtained based on the ripple curve for more accurate modeling. This curve contains higher order nonlinearity and discontinuity due to the valve-point effect as shown in Fig. 1. One way of representing this effect is to use a rectified sinusoidal function to represent the valve-point loading in the cost function [13]. In this case Eq. (2) can be written as

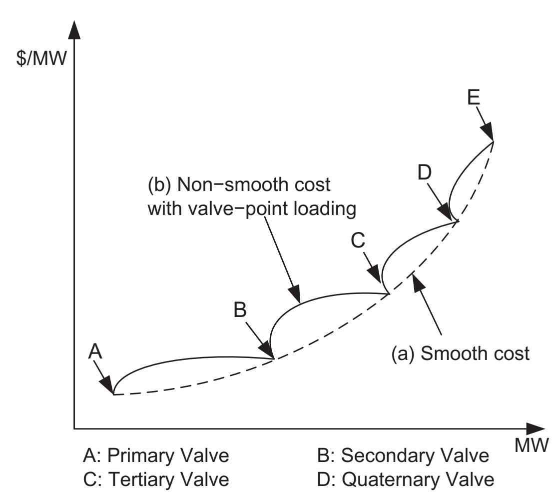

Fig. 1. Incremental fuel cost versus power output for a five valve steam turbine unit.

Table 1
Symbols and their meanings.

Symbol	Quantity
F_T	Objective function
F_i	Cost of generation of *i*th generator
N	Number of generating units
P_i	Generating unit real power output
a_i, b_i, c_i	Cost function coefficients of *i*th generating unit
P_{min}, P_{max}	Minimum and maximum power output limit of *i*th generating unit
P_l	Overall system real power losses
P_d	Total system real power demand
B_{ij}	Element of loss function coefficient
e_i, ρ_i	Fuel cost coefficients of *i*th generator considering valve-point effect
p	Total number of parameters
h_j	*j*th parameter (j = 1 to p)
W_{ij}	Initial value of *j*th parameter
W_{fj}	Final value of *j*th parameter
n_j	Number of steps for *j*th parameter
S_j	Step length for *j*th parameter
N_v	Total number of volumes
X_l	Optimized point found by *l*th agent ($l = 1–N_v$)
X_G	Global optimized point

$$f(P) = c_i + b_iP + a_iP^2 + |e_i \ \sin[\rho_i(P_{\min} - P)]| \quad (5)$$

where e_i and ρ_i are fuel cost coefficients of *i*th generator considering the valve-point effect and other variables are defined in Table 1.

3. Standard particle swarm optimization and its analysis

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart in 1995. It is popularly used in the complicated problem with non-linear and multi peak values. It is a population-based search algorithm that exploits a population of individuals to probe promising regions of the search space. The population here is called a swarm, and the individuals are called particles. PSO follows a stochastic optimization method based on swarm intelligence. The fundamental idea is that the optimal solution can be found through cooperation and information sharing among individuals in the swarm [14].

Each particle moves with a given random speed and moves within the search space and retains in its memory the best position it ever encountered. The standard PSO can be described here. Let $X_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ an n dimensional vector represents the current position of particle i in a search space S, $X_i \in S$. The current velocity of this particle is $V_i = [V_{i1}, V_{i2}, \ldots, V_{in}] \in S$. The past optional position encountered by the *i*th particle is denoted as $P_i = [P_{i1}, P_{i2}, \ldots, P_{in}] \in S$. Assume g to be the index of the particle that attains the best of all particles taken as global best of swarms. At last, the modified velocity and position of each particle can be calculated as follows:

$$V_{in}(t+1) = wV_{in}(t) + c_1r_1() \cdot (p_{in}(t) - x_{in}(t)) + c_2r_2() \cdot (p_{gn}(t) - x_{in}(t)) \quad (6)$$

$$x_{in}(t+1) = x_{in}(t) + V_{in}(t+1) \quad (7)$$

where C_1 and C_2 are constants of acceleration, $i = 1, 2, \ldots, N_p$ is particle index, $n = 1, 2, \ldots, N$ is the dimension index and $t = 1, 2\ldots$ indicates the iteration number, w is the weight of inertia and r_1 and r_2 are random numbers in [0, 1].

The inertia weight w plays a role of balancing the local and global search. Proper tuning of C_1 and C_2 results to an improved performance. Generalized models and techniques for tuning these parameters are analyzed in [14]. Since PSO is one of probability optimizer and hence it impossible for PSO to be guaranteed to converge the global optimization solution. Many methods have been introduced to solve this problem, introducing immunity and heredity result in loosing character, diversity result to more complex [15]. In this paper, we use an entirely different approach to improve PSO performance for overcoming curse of probability and uniqueness in solution to solve EPD problem.

4. Multi-agent system

Multi-agent system (MMS) is computational system in which several agent works, interact with each other and in unison take decision to achieve goals. According to [16,17] agent must have following properties: agents live and act in a given environment, agents are able to sense its local environment, and to interact with other agents in its local environment, agents attempt to achieve particular goals or perform particular tasks and agents are able to respond to changes that if occur in them.

The agents develop a society with collaboration to achieve their own individual as well as the common goal. The group decision-making process matches the basic nature of a particle in PSO and hence MAS provides an opportunity to compute and optimize complex problems. Some issues like the environment of agents, method of interaction, starting point of search, behavioral rules are to be addressed when used for optimization problems. These requirements and related issues to HMAPSO are addressed in the coming sections with experimental results and justifications.

5. Hybrid multiagent-based particle swarm optimization approach (HMAPSO)

In the proposed algorithm different agents are being sent in the whole search area which is divided into different fragments. The best solution in each fragment is being searched by its respective agent through modified Nelder–Mead method (NM method). For this purpose total range of the independent parameters are divided into smaller volumes, each of which determines the starting point for the exploration for each agent. The agent then finds its own optimized point by a developed optimization technique NM method. Each agent then passes the information regarding the optimized point by bee waggle dance. When all the information of optimized points is obtained then the best among these is chosen by consensus method as in case of honey bee swarms [19].

5.1. Particle search methodology

For optimization of the given objective function we have modified a very popular optimization technique usually known as Nelder–Mead method. The methodology used is deterministic search methodology but in a sense similar to swarm local search. Let $z = f(x, y)$ be the function that is to be minimized. For agents this is food function. To start, we assume that agent considers three vertices of a triangle as food points for a two variables problem as z_1, z_2 and z_3. $z_1 = (x_1, y_1)$ represents the initial position of agent $z_2 = (x_2, y_2)$ and $z_3 = (x_3, y_3)$ are the positions of probable food points i.e. local optimal points. The movement of agent from its initial position towards the food position, i.e. optimization point is as follows. Here we have considered the problem as to generate the minima of a function $z_i = f(x_i, y_i)$. The function $z_i = f(x_i, y_i)$ for i = 1, 2, 3 is evaluated at each of these three points. The obtained values of z_i are recorded in a way that $z_1 \leqslant z_2 \leqslant z_3$ with corresponding agents positions and food points as from the best to worst position. The construction process uses the midpoint of the line segment joining the two best food positions z_1 and z_2 as shown in Fig. 2a.

The value of function decreases as bee moves along z_3 to z_1 or z_3 to z_2. Hence it is feasible that $f(x, y)$ takes smaller value if bee

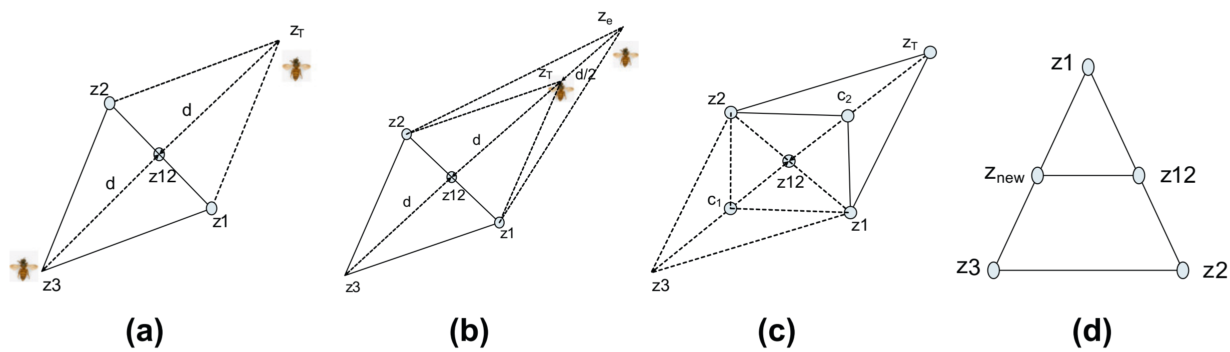

Fig. 2. Agents search movements with the proposed optimization algorithm. (a) Starting of the motion in search of solution, (b) extension in the direction of good optimal point, (c) contraction of the movement in case optimal point quality is not good, and (d) shrinking of the space towards optimistic solution.

moves towards z_{12}. For the further movement of the bee a test point z_T is chosen in such a way that it is reflection of the worst food point i.e. z_3 as shown in Fig. 2a. The vector formula for z_T is

$$z_T = 2 \times z_{12} - z_3 \tag{8}$$

If the function value at z_T is smaller than the function value at z_3, then the bee has moved in the correct direction towards minimum. Perhaps the minimum is just a bit further than the point z_T. So the line segment is extended further to z_e through z_T and z_{12} .The point z_e is found by moving as additional distance $d/2$ along the line as shown in Fig. 2b. If the function value at z_e is less than the function value at z_T, then the agent has found a better food point than z_T.

$$z_e = 2 \times z_T - z_{12} \tag{9}$$

If the function value at z_{12} and z_3 are the same, another point must be tested. Two test points are considered by the bee on the both sides of z_{12} at distance $d/2$ as shown in Fig. 2c.

The point of smaller value will frame a new triangle with other two best points. If the function value at the two test points is not less than the value at z_3, the points z_2 and z_3 must be shrunk towards z_1 as shown in Fig. 2d. The point z_2 is replaced with z_{12}, and z_3 is replaced with the midpoint of the line segment joining z_1 and z_3. Fig. 3 shows the path trace by the agents (bees) and the sequences of triangles $\{T_k\}$ converging to the optimal point for the objective function

$$f(x,y) = x^2 - 4x + y^2 - y - xy \tag{10}$$

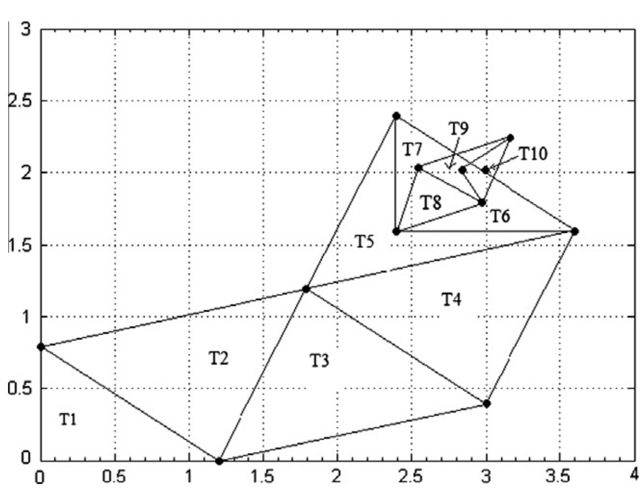

Fig. 3. Movement of the agents for a given problem.

5.1.1. Choice of starting point of searching in a volume

The solution of the Nelder–Mead method depends upon the starting location of the search in any volume. The algorithm is deterministic metaheuristic algorithm and leads to find out unique solution as compared to stochastic algorithms. The starting point is an important factor in such algorithms to find out global solution and experiment has been made to find the effect over optimal solution with change in starting point of exploration of agents in a volume. We have tested the algorithm on many standard functions and found the centre as the best point as starting point [19].

5.1.2. Choice of number of agents for searching

During the experiments it is found that small number of agents gives fast and give accurate result for simple problem having lower number of parameters whereas for more number of parameters more number of agents should go for exploration which in turn gives result with high accuracy but on the cost of time [19]. It is also observed that the centre of the lattice is a good starting point to get better optimal solution and 30–50 agents in number are sufficient to generate optimal solution.

5.2. Exploration

In MAS, all agents live in an environment [7]. An environment is organized in a structure as shown in Figs. 3 and 4. In the environment, each agent is fixed on a lattice-point and each circle represents an agent; the data in the circle represent the position of agent and the evaluated value of the function. The size and dimension of the lattice depends upon the variables and PSO.

The value of the objective function depends on p number of independent parameters. Let the range of jth parameter ∊ $[W_{ij}, W_{fj}]$, -where W_{ij} and W_{fj} represent the initial and final value of the parameter. Thus the complete domain of the objective function can be represented by a set of p number of axis. Each axis will be in a different dimension and will contain the total range of one parameter.

The next step is to divide each axis into smaller parts. Each of these parts is known as a step. Let the jth axis be divided in n_j number of step each of length S_j where j = 1 to p. This length S_j is known as step size for the jth parameter. The relationship between n_j and S_j can be given as

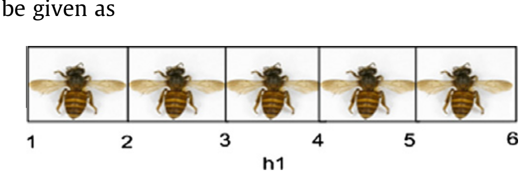

Fig. 4. Domain of the objective function with one independent parameter.

$$n_j = \frac{W_{fj} - W_{ij}}{S_j} \tag{11}$$

Hence each axis is divided into their corresponding branches. If we take one branch from each axis then these p number of branches will constitute a p dimensional volume. Total number of such volumes can be calculated as

$$\text{Number of volumes, } N_v = \prod_{j=1}^{p} n_j \tag{12}$$

The number of volumes indicates the number of scout bees going out for exploration. One point inside each volume is chosen as the starting point for the optimization, which in our approach is the midpoint of that volume, the reason for same is also being discussed here. The midpoint of total cluster can be calculated as follows

$$\left[\frac{W_{i1} + W_{f1}}{2}, \frac{W_{i2} + W_{f2}}{2}, \ldots, \frac{W_{ip} + W_{fp}}{2}\right] \tag{13}$$

For an objective function having one independent parameter, the complete domain will be given by single axis represented as h_1. Here each step will give us one volume. Let us take the following values

$p = 1, W_{1i} = 1, W_f = 6, S_1 = 1$

Therefore $n_1 = 5$ and $N_v = 5$. Thus five agents are sent for exploration. The starting point for each agent is the midpoint of each step as shown in Fig. 4.

For an objective function having two independent parameters, the complete domain will be given by a set of two axis represented as h_1 and h_2. Let us take the following values

$p = 2,\ W_{1i} = 1,\ W_{1f} = 5,\ S_1 = 1$ and $W_{2i} = 1,\ W_{2f} = 5,\ S_2 = 1$

Therefore $n_1 = 4, n_2 = 4$ and $N_v = 16$. Thus 16 agents are sent for exploration as shown in Fig. 5a. The starting point of each bee is the midpoint of each volume which is two dimensional rectangles in this case.

For an objective function with three independent parameters, the complete domain will be given by set of three axis represented as h_1, h_2 and h_3. Let us take the following values

$p = 3,\ W_{1i} = 1,\ W_{1f} = 5,\ S_1 = 1,\ W_{2i} = 1,\ W_{2f} = 4,\ S_2 = 1$ and $W_{3i} = 1,\ W_{3f} = 4,\ S_3 = 1$

Therefore $n_1 = 4, n_2 = 3, n_3 = 3$ and $N_v = 36$. Thus 36 agents are sent for exploration. The starting point for each bee is the midpoint of corresponding volume which is 3-dimensional cuboid in this case as shown in Fig. 5b. Objective functions with more than three independent parameters can also be solved in the similar manner.

5.3. Bee swarms based decision process

The honey bee swarms have a highly distributed decision-making process which they used for finding out their next hive or finding out new source of foods. Few hundreds of bees out of thousands work as *scout bees* to start a search for next possible site. Upon finding the site, scout informs other bees by *waggle dance* [11]. Discovered nest sites of sufficient quality are reported on the cluster via the scouts' waggle dance. Depending on the waggle dance by scout bees quiescent bees get activated and decided to recruit or explore for nest site. If an uncommitted bee is not satisfied with any of the scout sites then she can go for exploring new sites. When a bee advertises a site more than once then in every next turn she decreases the strength of her dance by about 15 dance circuits. Once the quorum threshold reaches for any one of the sites, the bee start *piping signals* that elicit heating by the quiescent bees in preparation for flight. There are two methods used by bee swarms decision for finding out the best nest site as *consensus* and *quorum* [18]. In *consensus* widespread agreement among the group is taken into account whereas in quorum the decision for best site happens when a site crosses the *quorum* (threshold) value. In the present paper, the *consensus* algorithm is used for finding out the optimum solution i.e. best food site.

5.3.1. Waggle dance

As bee after returning from search perform waggle dance to inform other bees about the quality of site or food. Here in the proposed algorithm the agents after collecting their individual optimal solution give to the centralized systems that choose the preferable solution from the searched one. For optimal minimum cases it selects the best optimal solution which can mathematically stated as

$$Wd_i = \min(f_i(X)) \tag{14}$$

where $f_i(X)$ represent the different search value obtained by an agent. Each of these points is recorded in a table known as optimum vector table X. X is a vector containing p number of elements. These elements contain the value of parameters at that point. So both the optimal solution value and the corresponding variable values are recorded. This record is known as *Personal Best* i.e. *Pbest* in PSO. The function value gets change according to the objective function requirement i.e. if objective function is to be minimized then the min function is used and if we have to find maximize in an objective function it will switch over to maximize function.

5.3.2. Consensus

As bee swarms use consensus method to decide the best obtained or search value. The authors mimic this event and behavior by comparing the results obtained. Once exploration and waggle

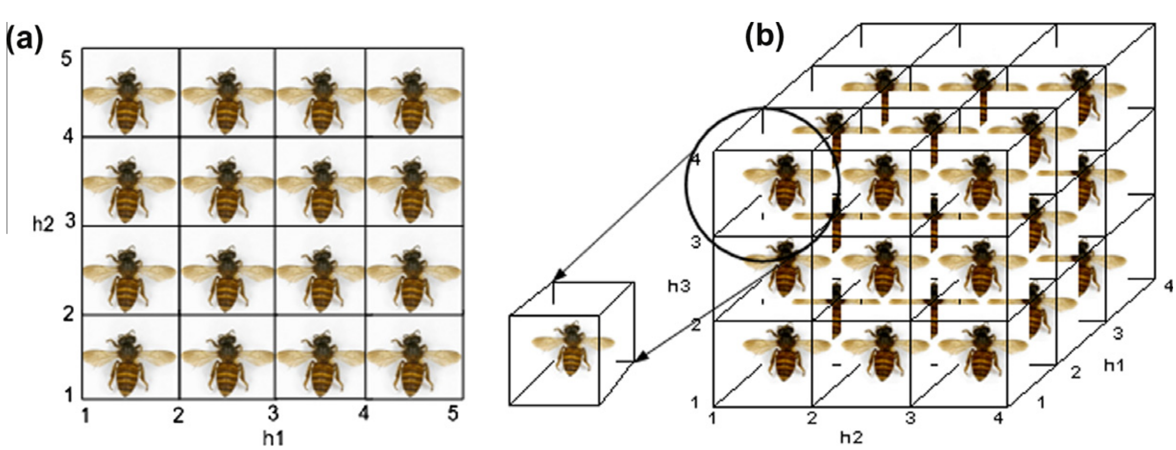

Fig. 5. Domain of the objective function with (a) two and (b) three independent parameters.

dance (transmission of data) is finished the global optimized point is chosen by comparing the fitness values of all the optimized points in the optimum vector table i.e. *global best, gbest* as in case of PSO. For minimization problems the point with the lowest fitness value is selected as the global optimized point. The global optimized point X_G is found by

$$f(X_G) = \min\ [f(X_1),\ f(X_2), \ldots, f(X_{N_v})] \tag{15}$$

Algorithm 1.

Initialize the number of parameters, p initialize the length of steps, S_j (j = 0 to p)
Initialize the range of each parameter as $[W_{ij}, W_{fj}]$ where j = 0, 1, ..., p
Calculate the number of steps $n_j = \frac{W_{fj}-W_{ij}}{S_j}$
Calculate the total number of volumes $N_v = \prod_{j=1}^{p} n_j$
For each volume, take the starting point of the exploration as the midpoint of the volume $\left[\frac{W_{i1}+W_{f1}}{2}, \frac{W_{i2}+W_{f2}}{2}, \ldots, \frac{W_{iP}+W_{fP}}{2}\right]$
Record the value of optimized point obtained corresponding to each volume in optimum vector table in following way $[X_1,\ X_2, \ldots, X_{N_v}]$
After the exploration is being completed, the global optimized point in the following manner
$F(X_G) = \min[F(X_1),\ F(X_2), \ldots, F(X_{Nv})]$

6. Results and discussions

The efficiency and robustness of the proposed HMAPSO has been tested. The algorithm has been applied to both smooth and nonsmooth functions. The objective function with valve-point effects is represented as a nonsmooth NP hard optimization problem with constraints. The results obtained from HMAPSO are compared with other similar methods: particle swarm optimization (PSO), personal best oriented particle swarm optimization (PPSO), mean personal best base oriented particle swarm optimization (MPPSO), adaptive personal best bas oriented particle swarm optimization (APPSO), and decisive personal base oriented particle swarm optimization (DPSO) [13].

(1) *Test Systems:* The HMAPSO is applied to EPD problems, one with 13 generators and another with 40 generators with valve-point effects in both the cases. The first test system has 13 units and details of this test system are listed in Table 2 [11]. The demanded load of this problem is 1800 MW. The second test system consists of 40 generators [13]. The details of this test system are given in Table 5. The load demand is 10,500 MW. The global solution for the 13 and 40 generator system is not discovered yet. The best solution reported until now is 17971.01 and 121,788.22 [$] [13] for 13 and 40 generator system respectively.

(2) *Parameter Settings for the Experimental Setup:* During PSO experiments the constants are set at $c_1 = c_2 = 1.5$ and the velocity of particles are confined in [−20, 20] for 13 generator system. Modifications have been made for 40 generator system as the system is larger than the previous one. The velocity of particle is now confined within [−40, 40]. In the proposed algorithm there is scope only for the step length (S_j) to be tuned. The appropriate selection of the S_j is done according to the sensitivity of the objective function to a particular parameter. For e.g.: Consider an objective function which is more sensitive to p_1 than p_2 then the step length selected for former will be lesser than the latter parameter. The selection is done on a relative basis and accordingly number of agents to be sent to search in the search space is determined according to the Eq. 13. Various simulations have been made to find out the optimal number of agents for HMAPSO as shown in Fig. (6). It is seen that 30 number of agents is sufficient for the problem and the same has been verified the authors earlier [19]. Hence for HMAPSO number of agents kept 30.

(3) *Numerical Results*: The obtained results for the 13-generator and 40-geerator systems are given in Tables 3 and 4 respectively and are compared with particle swarm optimization (PSO), personal best oriented particle swarm optimization (PPSO), mean personal best base oriented particle swarm optimization (MPPSO), adaptive personal best bas oriented particle swarm optimization (APPSO), and decisive personal base oriented particle swarm optimization (DPSO) [13]. It clearly shows that the proposed algorithm has succeeded in finding better solution than its counterparts.

For each experiment the simulation records the mean value, worst value and optimal value obtained in each parameter. In each experiment the simulation run 30 times. Table 3 shows the mean, maximum and minimum cost acquired by different algorithms. Fig. 7 illustrates that all other algorithms generate different solutions on different run whereas HMAPSO generates unique optimum solution as the randomness is removed from the algorithm. The same has been explained and tested on benchmark optimization problem [19]. It is worth mentioning that HMAPSO improved significantly in terms of uniqueness and optimal solution for both the cases. The optimum dispatch of each generator is also recorded in order to see it in permissible limit and presented in Table 4.

The convergence rate and the solution time depend largely on the step size and number of agents employed in HMAPSO. The step size and number of steps are related according to the Eq. (13). Since each step volume is assigned to an agent, number of steps is equal to agents. An increase in number of agents decreases the step length thus decreasing the search space per agent. This improves

Table 2
Generators data for case 1 (13 units).

G	P_{min} (MW)	P_{max} (MW)	a	b	c	e	ρ
1	0	680	0.00028	8.10	550	300	0.035
2	0	360	0.00056	8.10	309	200	0.042
3	0	360	0.00056	8.10	307	200	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

Table 3
Comparison of simulation results for 13 units (load = 1,800 mw).

Method	Minimum cost	Maximum cost	Mean cost
PSO	18014.16	18249.89	18104.65
PPSO	17971.01	18246.70	18106.33
MPPSO	17976.19	18210.59	18087.12
APPSO	17978.89	18291.92	18014.61
DPSO	17976.31	18310.43	18084.99
HMAPSO	**17969.31**	**17969.31**	**17969.31**

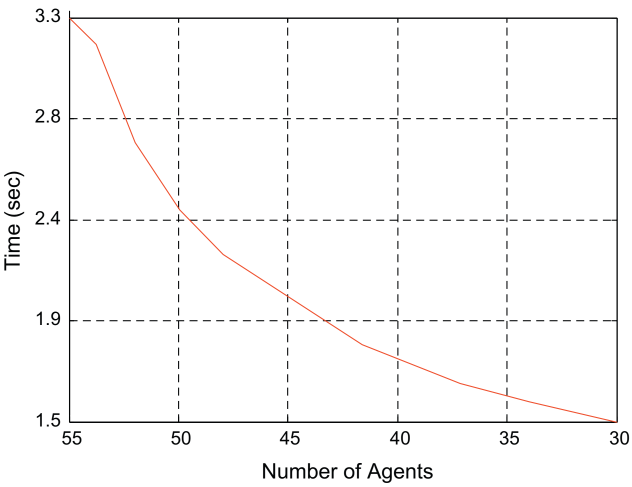

Fig. 6. Convergence time with number of agents.

the convergence characteristics at the expense of computational time. The convergence characteristics have been illustrated in the Fig. 6.

Studies have also been made to analyse the computation time taken to obtain the optimal value. Fig. 8 illustrates the time taken by similar algorithms along with HMAPSO. For more comprehensive analysis both 13 and 40 generator system have been studied. It is seen that the time taken by HMAPSO is less than its counterparts also it is to be noted that in other techniques we take 10–30 runs and then take the average of solution obtained. So if we consider this also in calculations then time taken by HMAPSO is really very small to similar algorithms. From Fig. 8 it can also be concluded that the increase in the complexity and size of the system put an effect on the computation time but again time taken by HMAPSO is less than other algorithms.

The generation dispatch of each generator is also calculated and shown in Table 6. As seen in Tables 3, 6 and 7 the HMAPSO has provided the better solution in comparison with other similar algorithms, exactly satisfying the equality and inequality constraints. It is also concluded that the proposed algorithm is more robust than the other algorithms as it generate a unique and optimal solution.

Table 4
The best solutions found for case 1.

G	PSO	PPSO	MPPSO	APPSO	DPSO	**HMAPSO**
1	538.561	538.618	448.803	538.557	448.799	**538.5611**
2	299.355	149.831	300.211	299.201	224.645	**224.4831**
3	75.037	224.390	300.031	224.460	226.539	**150.0622**
4	159.734	109.951	109.862	60.066	159.733	**109.8862**
5	60.078	109.837	60.033	110.019	109.867	**109.9902**
6	109.864	109.985	109.869	109.866	109.867	**109.8666**
7	109.913	109.998	60.014	60.121	109.867	**109.9903**
8	159.753	109.830	109.882	110.083	109.867	**109.8688**
9	60.069	109.917	109.947	60.075	109.867	**109.8668**
10	40.035	77.412	40.770	77.415	40.418	**40.0000**
11	77.561	40.155	40.461	40.105	40.494	**77.4247**
12	55.042	55.075	55.117	55.029	55.039	**55.0000**
13	55.000	55.000	55.000	55.001	55.000	**55.0000**

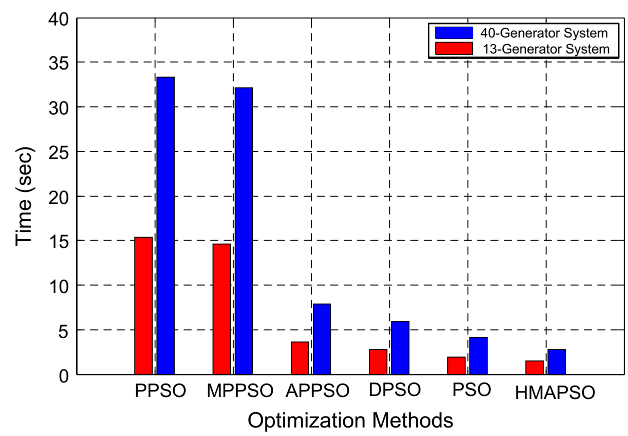

Fig. 8. Comparison of computation time with different algorithms.

7. Conclusions

In this paper a new optimization algorithm known as hybrid multi agent particle swarm optimization is employed to solve the economic power dispatch problem. The paper first analyzed the standard PSO algorithm and discusses its known problems of consistency in solution and of premature phenomenon. The upshot of

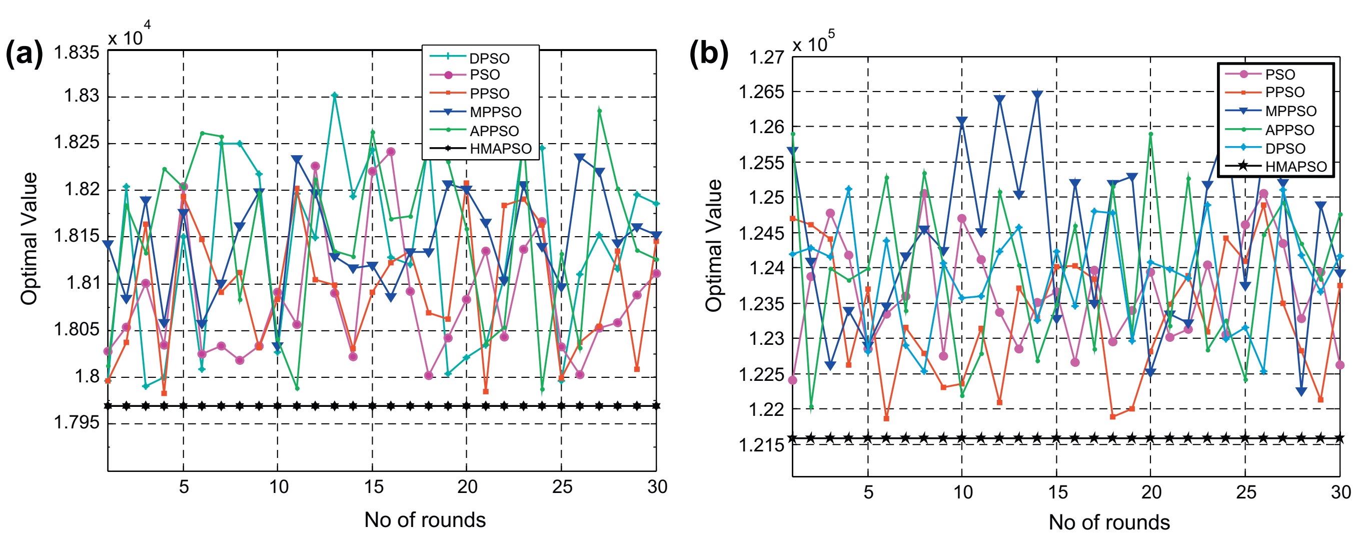

Fig. 7. Optimal solution variations with different numbers of rounds (a) 13-generator system, and (b) 40-generator system.

Table 5
Generators data for case 2 (40 units).

G	P_{min} (MW)	P_{max} (MW)	a	b	c	e	ρ
1	36	114	0.00690	6.73	94.700	100	0.084
2	36	114	0.00690	6.73	94.705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.20	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.40	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

Table 6
Comparison of simulation results for 40 units (load = 10,500 mw).

Method	Minimum cost	Maximum cost	Mean cost
PSO	122323.97	125103.28	123690.62
PPSO	121788.22	124998.23	123639.53
MPPSO	122225.73	126646.46	124723.59
APPSO	122044.63	126259.11	123985.15
DPSO	122159.99	125295.98	123647.81
HMAPSO	**121586.90**	**121586.90**	**121586.90**

the proposed algorithm is that it generates better optimal solutions as compared to its counterparts. The algorithm is based on Multi-agent system with collaboration of natural swarm group decision method by bees to find next site. The proposed algorithm performs well on different objective functions with any number of parameters, LPT equation and also performs well on unconstrained optimization problems. Experimental results prove the robustness and accuracy of HMAPSO over other PSO models. The results also show that HMAPSO removes the randomness in the algorithm and improves significantly in global optimization performance. A study is done on the convergence time with respect to the number of agents was done and it was seen that as the number of agents increased the convergence time increased. Since the number of agents is inversely proportional to the step length it can be inferred that the convergence is poor if the step length is small. The solution of HMAPSO shows consistency in the solution and hence it gives a better option to optimize real-time and on-line optimization problems.

Table 7
optimal dispatch of the 40 generators system.

G	PSO	PPSO	MPPSO	APPSO	DPPSO	**HMAPSO**
1	113.116	111.601	112.903	112.579	111.917	**111.136**
2	113.010	111.781	112.802	111.553	112.338	**111.135**
3	119.702	118.613	117.515	98.751	118.922	**120.000**
4	81.647	179.819	181.442	180.384	179.928	**177.221**
5	95.062	92.443	95.876	94.389	48.998	**088.699**
6	139.209	139.846	139.856	139.943	139.931	**140.000**
7	299.127	296.703	299.452	298.937	299.610	**260.157**
8	287.491	284.566	298.277	285.827	298.206	**284.723**
9	292.316	285.164	299.043	298.381	285.372	**285.523**
10	279.273	203.859	130.886	130.212	130.701	**130.000**
11	169.766	94.283	243.530	94.385	94.849	**168.805**
12	94.344	94.090	94.768	169.583	244.086	**168.689**
13	214.871	304.830	215.033	214.617	214.739	**304.123**
14	304.790	304.173	304.739	304.886	304.504	**304.678**
15	304.563	304.467	304.694	304.547	304.744	**304.317**
16	304.302	304.177	215.146	304.584	304.501	**304.317**
17	489.173	489.544	497.407	489.452	489.515	**489.187**
18	491.336	489.773	489.459	497.472	489.534	**489.455**
19	510.880	511.280	511.867	512.816	511.567	**512.097**
20	511.474	510.904	548.400	548.992	511.374	**511.349**
21	524.814	524.092	523.396	524.652	525.246	**523.247**
22	524.775	523.121	525.206	523.399	523.979	**523.515**
23	525.563	523.242	524.971	548.895	548.599	**523.454**
24	522.712	524.260	523.660	525.871	523.314	**523.453**
25	503.211	523.283	523.624	523.814	523.259	**523.492**
26	524.199	523.074	527.932	523.565	524.360	**523.307**
27	10.082	10.800	10.474	10.575	10.388	**10.000**
28	10.663	10.742	11.074	11.177	10.552	**10.000**
29	10.418	10.799	10.582	11.210	10.082	**10.000**
30	94.244	94.475	96.403	96.178	96.422	**88.691**
31	189.377	189.245	189.338	189.999	189.692	**190.000**
32	189.796	189.995	189.849	189.924	189.820	**190.000**
33	189.813	188.081	189.739	189.714	189.954	**190.000**
34	199.797	198.475	199.808	199.284	199.427	**164.218**
35	199.284	197.528	199.994	199.599	199.905	**200.000**
36	198.165	196.971	199.749	199.751	199.229	**200.000**
37	109.291	109.161	109.917	109.973	109.565	**110.000**
38	109.087	109.900	109.410	109.506	109.741	**110.000**
39	109.909	109.855	109.728	109.363	109.575	**110.000**
40	512.348	510.984	512.053	511.261	511.554	**511.009**

References

[1] Nagrath IJ, Kothari DP. Power system engineering. 1st ed. Tata Mcgraw-Hill Publishing Company Limited; 1995.
[2] Happ HH. Optimal power dispatch – a comprehensive survey. IEEE Trans Power Appl Syst 1997;PAS-96.
[3] Chowdhury BH, Rahman S. A Review of recent advances in economic dispatch. IEEE Trans power syst 1990:1248–59.
[4] Sinha N, Chakrabarti R, Chattopadhyay PK. Evolutionary programming techniques for economic load dispatch. IEEE Trans Evolution Comput 2003;7:83–94.
[5] Park JB, Lee KS, Sin JR, Lee KY. A particle swarm optimization for economic dispatch with non-smooth cost function. IEEE Trans Power Syst 2005;20:34–42.
[6] David E. Goldberg, genetic algorithms in search, optimization, and machine learning. 9th ed. Pearson Education; 2005.
[7] Wooldridge M. An introduction to multiagent system. Wiley: New York; 2002.
[8] Dorigo M, Birattari M, Stutzle T. Ant colony optimization, computational intelligence magazine. IEEE 2006;1(4):28–39.
[9] R.C. Eberhart, J. Kennedy. Particles swarm optimization. In: Proceedings of IEEE international conference on neural network. Perth, Australia; 1995.
[10] Passino Kevin M, Seeley TD. Modeling, analysis of nest-site selection by honey bee swarms: the speed and accuracy trade-off. Behav Ecol Sociobiol 2006;59(3):427–42.
[11] Passino Kevin M, Seeley Thomas D, Kirk Visscher P. Swarm cognition in honey bee. Behav Ecol Sociobiol 2008;62:3.
[12] Nelder JA, Mead R. A simplex method for function minimization. Comput J 1965;7:308–13.
[13] C.H. Chen, S.N. Yeh. Particle swarm optimization for economic power dispatch with valve-point effects. In: IEEE PES transmission and distribution conference; 2006. p. 1–5.

[14] Lei Wang, Qi Kang, Hui Xiao, Qidi Wu. A modified adaptive particle swarm optimization algorithm. In: IEEE international conference on industrial technology, ICIT 2005; 2005. p. 209–14.
[15] Shinn-Ying Ho, SO OP, et al. Orthogonal particle swarm optimization and its application to task assignment problems. IEEE Trans Syst Man Cybern A 2008;38(2):288–98.
[16] Jiao Wenpin, Shi Zhongzhi. A dynamic architecture for multi-agent systems, technology of object-oriented languages and systems. Tools 1999;1:253–60.
[17] Zhong WZ et al. A multiagent genetic algorithm for global numerical optimization. IEEE Trans Syst Man Cybern B 2004;34:1128–41.
[18] Seeley T, Kirk Visscher P, Passino Kevin M. Group decision making in honey bee swarms. Am Sci 2006;94(3):220–9.
[19] Kumar Rajesh, Kumar Anupam, Sharma Devendra. A new hybrid multiagent – based particle swarm optimization technique. Int J Bio-Inspired Comput 2009;1(4):259–69.