Electrical Power and Energy Systems 33 (2011) 131–136

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

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

Artificial immune system for dynamic economic dispatch

M. Basu

Department of Power Engineering, Jadavpur University, Kolkata 700098, India

ARTICLE INFO

Article history:
Received 16 July 2008
Received in revised form 25 February 2010
Accepted 7 June 2010

Keywords:
Dynamic economic dispatch
Artificial immune system
Clonal selection principle

ABSTRACT

Dynamic economic dispatch determines the optimal scheduling of online generator outputs with predicted load demands over a certain period of time taking into consideration the ramp rate limits of the generators. This paper proposes artificial immune system based on the clonal selection principle for solving dynamic economic dispatch problem. This approach implements adaptive cloning, hyper-mutation, aging operator and tournament selection. Numerical results of a ten-unit system with nonsmooth fuel cost function have been presented to validate the performance of the proposed algorithm. The results obtained from the proposed algorithm are compared with those obtained from particle swarm optimization and evolutionary programming. From numerical results, it is found that the proposed artificial immune system based approach is able to provide better solution than particle swarm optimization and evolutionary programming in terms of minimum cost and computation time.

1. Introduction

Static economic dispatch (SED) allocates the load demand which is constant for a given interval of time, among the online generators economically while satisfying various constraints including static behavior of the generators. Dynamic economic dispatch (DED) is an extension of static economic dispatch problem. It schedules the online generator outputs with the predicted load demands over a certain period of time so as to operate an electric power system most economically. In order to avoid shortening the life of the equipment, plant operators try to keep gradients for temperature and pressure inside the boiler and turbine within safe limits. This mechanical constraint is transformed into a limit on the rate of increase or decrease of the electrical power output. This limit is called ramp rate limit which distinguishes DED from SED problem. Thus, the dispatch decision at one time period affects those at later time periods. DED is the most accurate formulation of the economic dispatch problem but it is the most difficult to solve because of its large dimensionality. Further, due to increasing competition into the wholesale generation markets, there is a need to understand the incremental cost burden imposed on the system by the ramp rate limits of the generators.

Since the DED was introduced, several classical methods [1–7] have been employed for solving this problem. However, all of these methods may not be able to find an optimal solution and usually stuck at a local optimum solution. Classical calculus-based methods address DED problem with convex cost function. But in reality large steam turbines have a number of steam admission valves, which contribute nonconvexity in the fuel cost function of the generating units. Dynamic programming (DP) can solve such type of problems but it suffers from the curse of dimensionality.

Recently, stochastic search algorithms [8–16] such as simulated annealing (SA), Genetic algorithm (GA), evolutionary programming (EP) and particle swarm optimization (PSO) have been successfully used to solve power system optimization problems due to their ability to find the near global solution of a nonconvex optimization problem. The SA is a powerful optimization technique but in practice, the annealing schedule of SA should be carefully tuned otherwise achieved solution will still be of locally optimal. Nevertheless, an appropriate annealing schedule often requires tremendous computation time. Both GA and EP based on the metaphor of natural biological evolution can provide near global solution. EP differs from GA in aspect that EP relies primarily on mutation and selection but not crossover as in GA. Hence, considerable computation time may be saved in EP. In spite of their successful implementation, both GA and EP posses some weakness leading to more computation time and less guaranteed convergence in case of highly epistatic objective functions i.e. the parameters being optimized are highly correlated. Although PSO can be used to solve nonlinear and noncontinuous optimization problem, it suffers from premature convergence especially while handling problems with more local optima.

Artificial immune system (AIS) [17–22] has emerged in the 1990s as a new branch in computational intelligence. AIS is inspired by immunology, immune function and principles observed in nature. It is now interest of many researchers and has been successfully used in power system optimization problems [23–26].

E-mail address: mousumibasu@yahoo.com

0142-0615/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijepes.2010.06.019

Nomenclature

P_{it}	real power output of ith unit during time interval t	B_{ij}	loss coefficient
$P_i^{\min}$, $P_i^{\max}$	lower and upper generation limits for ith unit	N	number of generating units
P_{Dt}	load demand at the time interval t	N_P	population size
P_{Lt}	transmission line losses at time t	N_c	number of clones
a_i, b_i, c_i, d_i, e_i	cost coefficients of ith unit	T	number of intervals in the scheduled horizon
$F_{it}(P_{it})$	cost of producing real power output P_{it} at time t		
UR_i, DR_i	ramp-up and ramp-down rate limits of the ith generator		

In this paper AIS algorithm is developed for solving the DED problem. The proposed approach is based on the clonal selection principle and implements adaptive cloning, hyper-mutation, aging operator and tournament selection. In order to show the validity of the proposed approach the developed algorithm is illustrated on a ten-unit system [10] with nonsmooth fuel cost function. Results obtained from the proposed approach are compared with those obtained using particle swarm optimization and evolutionary programming. The comparison shows that the proposed AIS based approach performs the best amongst three in terms of minimum production cost and computation time.

2. Problem formulation

Normally, the DED problem minimizes the following total production cost of committed units:

$$F = \sum_{t=1}^{T}\sum_{i=1}^{N} F_{it}(P_{it}) \tag{1}$$

The fuel cost function of each unit considering valve-point effect [11] can be expressed as

$$F_{it}(P_{it}) = a_i + b_i P_{it} + c_i P_{it}^2 + |d_i \sin\{e_i(P_i^{\min} - P_{it})\}| \tag{2}$$

Subject to the following equality and inequality constraints for the tth interval in the scheduled horizon

(i) Real power balance

$$\sum_{i=1}^{N} P_{it} - P_{Dt} - P_{Lt} = 0 \quad t \in T \tag{3}$$

(ii) Real power operating limits

$$P_i^{\min} \leqslant P_{it} \leqslant P_i^{\max} \quad i \in N,\ t \in T \tag{4}$$

(iii) Generating unit ramp rate limits

$$\begin{aligned} P_{it} - P_{i(t-1)} &\leqslant UR_i, \quad i \in N,\ t = 2, \ldots, T \\ P_{i(t-1)} - P_{it} &\leqslant DR_i, \quad i \in N,\ t = 2, \ldots, T \end{aligned} \tag{5}$$

3. Determination of generation levels

In this approach, the power loading of first $(N-1)$ generators are specified. From the equality constraints in Eq. (3) the power level of the Nth generator (i.e. the remaining generator) is given by

$$P_{Nt} = P_{Dt} + P_{Lt} - \sum_{i=1}^{N-1} P_{it} \quad t \in T \tag{6}$$

The transmission loss P_{Lt} is a function of all the generators including that of the dependent generator and it is given by

$$P_{Lt} = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} P_{it}B_{ij}P_{jt} + 2P_{Nt}\left(\sum_{i=1}^{N-1} B_{Ni}P_{it}\right) + B_{NN}P_{Nt}^2 \quad t \in T \tag{7}$$

Expanding and rearranging, Eq. (6) becomes

$$B_{NN}P_{Nt}^2 + \left(2\sum_{i=1}^{N-1} B_{Ni}P_{it} - 1\right)P_{Nt} + \left(P_{Dt} + \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} P_{it}B_{ij}P_{jt} - \sum_{i=1}^{N-1} P_{it}\right) = 0 \quad t \in T \tag{8}$$

The loading of the dependent generator (i.e. Nth) can then be found by solving Eq. (8) using standard algebraic method.

4. Immune system

The immune system of vertebrates including human is composed of cells, molecules and organs in the body which protect the body against infectious diseases caused by foreign pathogens such as viruses, bacteria, etc. To perform these functions, the immune system has to be able to distinguish between the body's own cells as the self cells and foreign pathogens as the non-self cells or antigens. After distinguishing between self and non-self cells, the immune system has to perform an immune response in order to eliminate non-self cell or antigen. Antigens are further categorized in order to activate the suitable defense mechanism and at the same time, the immune system also developed a memory to enable more efficient responses in case of further infection by the similar antigen.

Clonal selection theory explains how the immune system fights against an antigen. It establishes the idea that only those cells which recognize the antigen, are selected to proliferate. The selected cells are subjected to an affinity maturation process which improves their affinity to the selected antigens.

Clonal selection operates both on B-lymphocytes or B cells produced by the bone marrow and T-lymphocytes or T cells produced by the thymus. When the body is exposed to an antigen, B cells would respond to secrete specific antibodies to the particular antigen. Thereafter, a second signal from the T-helper cells, a subclass of T cells, would then stimulate the B cell to proliferate and mature into terminal (non-dividing) antibody secreting cells called plasma cells. In proliferation, clones are generated in order to achieve the state of plasma cells as they are the most active secretors of the antibodies at a larger rate than rate of antibody secretion by the B cells. The proliferation rate is directly proportional to the affinity level i.e. higher the affinity level of B cells more clones is generated. Clones are mutated at a rate inversely proportional to the antigen affinity i.e. clones of higher affinity are subjected to less mutation compared to those which exhibit lower affinity.

This process of selection and mutation of B cells is known as affinity maturation.

T cells do not secrete antibodies but play a central role in the regulation of the B cell response and are the most excellent in cell mediated immune responses. Lymphocytes, in addition to proliferating into plasma cells, can differentiate into long-lived B memory cells. These memory cells circulate through the blood, lymph and tissues, so that when exposed to a second antigenic stimulus, they commence to differentiate into plasma cells capable of producing high affinity antibody, preselected for the specific antigen that had stimulated the primary response.

5. Artificial immune system

Artificial immune system (AIS) mimics these biological principles of clone generation, proliferation and maturation. The main steps of AIS based on clonal selection principle are activation of antibodies, proliferation and differentiation on the encounter of cells with antigens, maturation by carrying out affinity maturation process, eliminating old antibodies to maintain the diversity of antibodies and to avoid premature convergence, selection of those antibodies whose affinities with the antigen are greater.

In order to emulate AIS in optimization, the antibodies and affinity are taken as the feasible solutions and the objective function respectively. Real number is used to represent the attributes of the antibodies.

Initially, a population of random solutions is generated which represent a pool of antibodies. These antibodies undergo proliferation and maturation. The proliferation of antibodies is realized by cloning each member of the initial pool depending on their affinity. In minimization problem, a pool member with lower objective value is considered to have higher affinity. The proliferation rate is directly proportional to the affinity of the antibodies. The maturation process is carried through hyper-mutation which is inversely proportional to the antigenic affinity of the antibodies. The next step is the application of the aging operator. This aging operator eliminates old antibodies in order to maintain the diversity of the population and to avoid the premature convergence. In this operator an antibody is allowed to remain in the population for at most τ_B generations. After this period, it is assumed that this antibody corresponds to local optima and must be eliminated from the current population, no matter what its affinity may be. During the cloning expansion, a clone inherits the age of its parent and is assigned an age equal to zero when it is successfully hyper-mutated i.e. when hyper-mutation improves its affinity.

Fig. 1 shows the flowchart of artificial immune system algorithm.

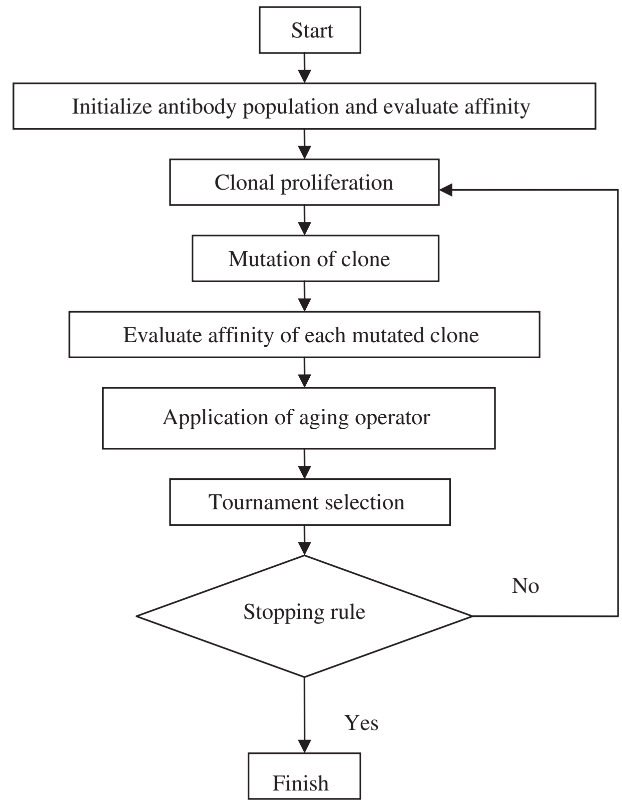

Fig. 1. Flowchart of artificial immune system.

6. Clonal selection-based AIS for solving DED problem

AIS based on clonal selection principle is presented here to solve DED problem. The algorithmic steps are as follows:

Step 1. Let

$$p_k = [(P_{11}, P_{21}, \ldots, P_{i1}, \ldots, P_{N1}), \ldots, (P_{1t}, P_{2t}, \ldots, P_{it}, \ldots, P_{Nt}), \ldots, (P_{1T}, P_{2T}, \ldots, P_{iT}, \ldots, P_{NT})]^{/}$$

be the kth antibody of a population to be evolved and $k = 1, 2, \ldots, N_P$. The elements of p_k are real power outputs of the committed N generating units over T number of intervals. The real power output of the ith unit at the tth interval is determined by setting $P_{it} \sim U(P_i^{\min}, P_i^{\max})$, where $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$. $U(P_i^{\min}, P_i^{\max})$ denotes a uniform random variable ranging over $[P_i^{\min}, P_i^{\max}]$. Each antibody should satisfy the constraints given by Eqs. (3)–(5).

Step 2. As DED is a minimization problem, affinity is the inverse of the objective function and it is given by the following equation.

$$\text{Affinity} = \frac{1}{\sum_{t=1}^{T}\sum_{i=1}^{N} a_i + b_i P_{it} + c_i P_{it}^2 + |d_i \sin\{e_i(P_i^{\min} - P_{it})\}|} \quad (9)$$

Step 3. The antibodies are cloned directly proportional to their affinities.

Step 4. The clones undergo maturation process through hyper-mutation mechanism and are given by the following equation

$$P_{ijt}^{/} = P_{kjt} + mul \times \frac{F_k}{F_{\min}} \times N(0,1) \times (P_j^{\max} - P_j^{\min}),$$
$$k = 1, \ldots, N_P, \; i = 1, \ldots, N_c, \; j = 1, \ldots, N \quad (10)$$

where $F_{\min}$ is the minimum value of F among the N_P solutions, mul is a scaling factor, F_k is the value of the function associated with p_k and $N(0, 1)$ represents a Gaussian random variable with mean 0 and standard deviation 1. $P_{ijt}^{/}$ is the real power output of jth unit of ith clone during time interval t. The term $\frac{F_k}{F_{\min}}$ makes the mutation more intensive in antibodies with a high production cost and smooth in antibodies with low production cost.

Each mutated clone must satisfy the constraints given by Eqs. (3)–(5).

Step 5. The affinities of the mutated clones are evaluated.

Step 6. The aging operator eliminates old antibodies in order to maintain the diversity of the population and to avoid the premature convergence. For this reason, an antibody is allowed to survive for at most τ_B generations. The aging operator eliminates those individuals which have more than τ_B generations

from the current population. When an individual is $\tau_B + 1$ old it is erased from the current population, no matter what its fitness value may be.
Step 7. Tournament selection is done to select a new population of the same size as the initial from the antibodies and mutated clones which are remained after application of aging operator.

Each of the antibodies and mutated clones which are remained after the application of aging operator undergoes a series of N_t tournaments with randomly selected opponents. The score for each population after a stochastic competition is given by

$$S_{p_k} = \sum_{l=1}^{N_t} S_l \qquad S_l = 1 \quad \text{if } F_{p_k} < F_{p_r} \qquad = 0 \text{ otherwise} \tag{11}$$

The competitor p_r is selected at random from among the antibodies and mutated clones. After competing the antibodies and mutated clones are ranked in descending order of the score obtained in (10). The first N_P population is selected for the next generation.

Step 8. If the maximum number of generations is reached, output the optimal solution i.e. the highest affinity value obtained so far and terminates the proposed algorithm. Otherwise, go back to Step 3.

7. Simulation results

In order to demonstrate the performance of the proposed AIS algorithm, a ten-unit test system with nonsmooth fuel cost function is used. The demand of the system has been divided into 24

Table 1
Hourly generation (MW) schedule, cost ($\times 10^6$ \$) and CPU time (second) of dynamic economic dispatch obtained from AIS.

Hour	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	Cost	CPU time
1	151.7188	142.7248	155.2973	96.9809	118.9239	98.0044	99.5246	70.0395	79.4591	43.1083		
2	152.0119	135.0000	185.8322	106.2452	165.6634	118.2397	124.2103	68.3935	52.5093	24.3062		
3	150.0000	147.4581	264.1075	152.0469	165.6587	113.9966	124.7598	76.0711	48.4324	44.4734		
4	165.5986	138.7404	309.8241	201.3245	181.8031	149.5624	119.1335	82.8205	59.3074	33.9469		
5	150.0000	135.9797	286.1487	239.0691	226.5914	142.8417	127.9076	110.7802	55.3247	44.9427		
6	155.6444	185.5964	333.6620	286.2148	230.1123	154.6898	125.6117	96.9938	66.8544	41.2362		
7	220.6328	173.4209	340.0000	288.6863	235.1433	160.0000	130.0000	111.4642	78.3817	49.5989		
8	191.8259	252.3653	335.6473	300.0000	239.2651	153.0154	128.4027	100.7748	79.3723	54.2562		
9	270.1137	327.4564	341.6645	296.9106	242.4084	160.0000	129.6346	119.1165	59.3356	48.3666		
10	300.0550	401.5677	335.5574	300.0000	241.7840	159.3614	128.0621	117.4973	64.2570	53.9388		
11	309.5846	470.0000	340.0000	298.4852	243.0000	158.2159	130.0000	120.0000	78.7250	46.3176	2.5197	53.56
12	387.3360	464.7866	338.5846	299.6854	237.2337	157.4834	128.2107	118.2156	59.6021	52.1431		
13	310.2941	438.8216	339.2683	296.5652	242.0924	159.0110	128.3990	118.8761	69.0924	54.5769		
14	233.0006	359.1525	340.0000	296.0900	228.5755	157.8933	128.0513	119.6744	79.9437	52.7125		
15	178.9715	280.3439	336.3661	300.0000	238.6375	154.3277	129.2965	96.5616	70.0446	50.5518		
16	147.0967	201.0938	304.3512	279.2461	236.1371	115.9083	103.4811	119.7890	51.1641	40.2571		
17	152.7787	135.0000	282.9980	257.1466	238.6955	147.6492	128.5761	92.4929	41.5873	42.7073		
18	161.2552	191.8650	302.0309	293.2186	232.3159	149.7215	129.4764	118.1036	60.6432	37.8760		
19	229.4679	244.3134	319.1639	299.0818	236.5163	160.0000	124.7610	116.0010	54.3140	51.3141		
20	307.3244	323.8625	335.8765	298.0084	243.0000	158.3355	130.0000	117.1829	80.0000	53.4780		
21	261.6011	318.6516	332.2783	300.0000	242.5040	160.0000	129.0591	119.7926	76.8193	54.0164		
22	182.3335	239.6949	260.2021	257.4387	235.2103	140.2336	126.5771	118.1959	63.2538	51.7462		
23	150.0000	183.3650	217.7193	247.6629	186.9385	112.7585	111.6819	89.8889	35.4291	29.0366		
24	150.3447	135.0000	155.2479	238.2374	141.5510	124.3791	112.3036	64.0560	48.4250	39.8908		

Table 2
Hourly generation (MW) schedule, cost ($\times 10^6$ \$) and CPU time (second) of dynamic economic dispatch obtained from PSO.

Hour	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	Cost	CPU time
1	178.2974	152.4782	175.9679	152.2693	82.6967	100.2570	55.1840	86.1211	32.0525	40.7531		
2	151.9730	135.0000	213.2147	201.7864	131.9888	118.5195	33.4036	82.8231	35.5465	28.5607		
3	203.1254	163.9726	251.9514	199.1529	138.8356	113.3658	42.5714	70.8768	56.7945	47.1242		
4	150.7934	200.2340	325.5753	226.0880	162.2524	113.6450	58.7423	96.9791	62.9590	45.6968		
5	173.3115	144.6134	285.1757	275.9783	211.5299	135.3282	86.1134	107.2580	52.8395	47.9709		
6	193.0859	180.9466	295.7199	294.4164	243.0000	150.2837	91.6984	119.8831	59.2527	48.5979		
7	199.5730	227.0251	317.7981	285.0641	232.0515	149.8511	121.3454	95.4300	74.6611	53.1300		
8	230.4454	246.3501	318.5516	281.4019	243.0000	160.0000	130.0000	118.9990	55.1984	50.9417		
9	309.2169	325.9173	340.0000	300.0000	240.8720	159.7176	127.3408	107.4298	34.9745	50.1778		
10	319.3131	405.6572	338.3825	291.2641	243.0000	155.7018	130.0000	120.0000	54.6931	44.5437		
11	376.5109	470.0000	293.8462	299.8919	241.4655	145.5909	128.4700	117.7083	76.6091	45.5225	2.5722	68.47
12	410.5210	460.0330	330.1543	290.0606	240.5785	157.8225	130.0000	99.3619	72.5975	52.6241		
13	354.5582	458.7721	325.3415	294.7094	243.0000	158.6978	125.2569	100.0073	61.8662	35.9623		
14	300.6076	379.5915	301.5840	285.1112	215.5048	148.7961	130.0000	119.2818	71.6677	43.9232		
15	229.4357	300.8396	328.6496	276.8651	236.1853	136.8260	118.9267	102.5443	68.2499	37.5047		
16	150.0000	230.9564	290.8325	285.5966	226.2529	129.9050	111.5957	95.6673	39.5550	38.3823		
17	168.5816	155.4401	312.8637	259.7161	197.5561	142.4882	119.8778	84.3126	39.2104	40.0607		
18	150.0000	235.2465	326.6851	295.5718	233.0072	131.9463	121.1258	103.6587	43.9754	36.0901		
19	225.5372	252.2426	308.9457	293.8158	241.5050	160.0000	130.0000	120.0000	72.9211	29.9004		
20	304.6380	331.5404	337.4662	300.0000	243.0000	154.7102	127.7088	118.4984	80.0000	49.6164		
21	290.6766	310.6099	340.0000	299.9669	242.0460	159.8559	130.0000	117.6984	72.4081	32.2045		
22	211.5595	231.5171	260.5104	288.5803	242.1627	130.7513	110.3033	101.3453	55.6514	45.1916		
23	150.0000	165.9392	212.1930	239.0283	196.6465	98.0630	116.2304	119.0883	29.2393	37.8554		
24	180.3674	135.0000	133.2234	190.3317	168.2021	132.2441	104.0532	98.6428	54.6286	12.8466		

Table 3
Hourly generation (MW) schedule, cost ($\times 10^6$ \$) and CPU time (second) of dynamic economic dispatch obtained from EP.

Hour	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	Cost	CPU time
1	150.0000	137.3240	165.8277	130.9594	185.8528	86.6659	43.1268	82.1384	60.5076	13.3651		
2	154.0813	142.0823	111.6599	173.9694	212.5123	127.5569	26.5848	90.0247	58.4067	35.8485		
3	167.6789	175.6266	190.4018	182.6740	243.0000	100.9178	33.9537	104.2804	43.8270	45.1084		
4	151.0869	234.2356	201.1268	192.8167	232.0634	140.5203	58.3984	120.0000	61.3047	51.1215		
5	179.0793	167.5926	241.0310	242.2296	233.1403	132.5917	85.7099	110.1910	76.4297	52.2067		
6	178.1383	246.7190	319.0410	243.9511	222.8359	130.1020	113.9272	114.8017	69.8714	38.1359		
7	231.2156	232.1208	330.9049	293.7810	238.4605	127.7300	101.5284	116.2500	70.6214	13.9564		
8	186.8206	299.0632	336.2252	278.7172	243.0000	152.4734	117.8664	115.2754	79.4536	26.5290		
9	254.2367	344.7199	340.0000	296.9019	235.7257	157.4128	125.5643	120.0000	72.5853	47.9489		
10	323.6424	423.9759	339.7272	282.1438	243.0000	157.9184	128.9391	97.5989	54.7205	51.3773		
11	379.9329	470.0000	333.0634	300.0000	236.4433	160.0000	115.5534	93.1074	54.6007	53.1176	2.5854	72.68
12	413.1634	443.5430	337.5237	289.1344	238.2085	159.5075	124.2170	119.3562	80.0000	38.7139		
13	335.4767	470.0000	334.4274	295.0397	230.2758	136.9080	129.4886	116.0717	73.7319	36.8421		
14	279.7477	390.3287	309.3307	277.8978	236.6148	157.8536	129.2263	93.2559	74.6412	47.1851		
15	225.8492	300.6198	340.0000	271.4831	235.7474	160.0000	113.9795	97.3194	60.2646	30.6238		
16	156.8943	221.9919	274.1830	222.1209	224.4294	145.5529	120.9893	119.0116	71.2232	41.9545		
17	150.0000	142.9721	307.5557	236.0399	229.9259	119.3528	128.4166	92.4584	77.2718	35.9486		
18	213.3822	209.6171	280.5045	285.5809	236.6284	160.0000	118.0363	89.8209	48.4504	35.0462		
19	216.8980	288.5752	330.2142	300.0000	233.9977	126.9242	128.6758	99.3605	75.2536	35.9281		
20	271.3901	367.8558	340.0000	298.0471	243.0000	160.0000	129.7412	108.7347	78.6677	49.8348		
21	285.6118	378.6250	327.2155	300.0000	240.5381	151.3773	127.6958	102.3417	51.2099	31.5443		
22	216.5251	289.8412	318.2870	250.3409	190.6218	112.0639	127.3546	100.7594	46.4267	26.1429		
23	157.4794	210.8063	256.3831	215.0895	181.2377	116.3920	97.4503	82.3338	59.7178	43.7112		
24	150.0000	140.8972	179.9885	221.9082	180.1364	106.8230	87.4166	63.2514	33.6222	45.5223		

intervals. Unit data has been adopted from [10]. Simulations have been carried out on a P-IV, 80 GB, 3.0 GHz personal computer and coding is written using MATLAB.

Results obtained using the proposed AIS algorithm, are shown in Table 1. Here, scaling factor (*mul*), population size (N_P) and maximum iteration number ($N_{\max}$) are taken as 0.1, 50 and 400 respectively for the test system under consideration.

To validate the proposed AIS based approach, the same ten-unit test system is solved by the author using PSO and EP. The PSO control parameters are $c_1 = 2$, $c_2 = 2$, $W_{\max} = 1.2$ and $W_{\min} = 0.7$, $N_P = 100$ and $N_{\max} = 400$. Table 2 presents the results obtained from PSO.

In case of EP, control parameters are scaling factor $\beta = 0.04$, $N_P = 100$ and $N_{\max} = 400$. The results obtained from EP are given in Table 3.

Fig. 2 shows the cost convergence obtained from AIS, PSO and EP. Table 1, Table 2 and Table 3 reveals that AIS has achieved lower minimum production cost and less CPU time than PSO and EP.

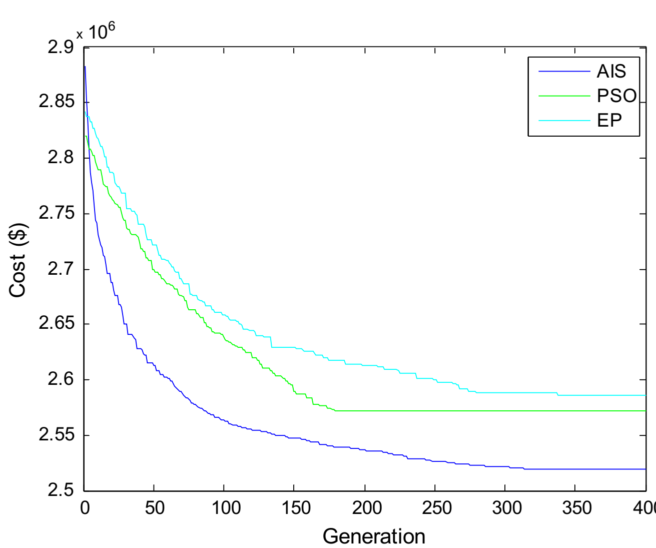

Fig. 2. Cost convergence.

8. Conclusion

This paper has presented a novel approach based on AIS for solving DED problem. AIS algorithm utilizes adaptive cloning, hyper-mutation, aging operator and tournament selection. Real number representation of the antibody attributes is implemented here. The effectiveness of the proposed method is illustrated by using a ten-unit test system and compared with the results obtained from PSO and EP. It is evident from the comparison that the proposed AIS based approach provides better results than PSO and EP in terms of minimum production cost and computation time.

References

[1] Ross DW, Kim S. Dynamic economic dispatch of generation. In: IEEE transactions on power apparatus and systems, vol. PAS-99, No. 6, November/December 1980. p. 2060–8.

[2] Van Den Bosch PPJ. Optimal dynamic dispatch owing to spinning-reserve and power-rate limits. In: IEEE transactions on power apparatus and systems, vol. PAS-104, No. 12, December 1985. p. 3395–401.

[3] Granelli GP, Marannino P, Montagna M, Silvestri A. Fast and efficient gradient projection algorithm for dynamic generation dispatching. IEE Proc Gener Transm Distrib 1989;136(5):295–302.

[4] Hindi KS, Ab Ghani MR. Dynamic economic dispatch for large scale power systems; a Lagrangian relaxation approach. Electr Power Syst Res 1991;13(1):51–6.

[5] Lee FN, Lemonidis L, Liu Ko-Chih. Price-based ramp-rate model for dynamic dispatch and unit commitment. IEEE Trans Power Syst 1994;9(3):1233–42.

[6] Travers DL, Kaye RJ. Dynamic dispatch by constructive dynamic programming. IEEE Trans Power Syst 1998;13(1).

[7] Han XS, Gooi HB, Kirschen DS. Dynamic economic dispatch: feasible and optimal solutions. IEEE Trans Power Syst 2001;16(1):22–8 [pp. 72–78].

[8] Attavriyanupp P, Kita H, Tanaka T, Hasegawa J. A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. IEEE Trans Power Syst 2002;17(2):411–6.

[9] Victoire TAA, Jeyakumar AE. Reserve constrained dynamic dispatch of units with valve-point effects. IEEE Trans Power Syst 2005;20(3):1273–82.

[10] Basu M. Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II. Int J Electr Power Energy Syst 2008;30(2):140–9.

[11] Wong KP, Fung CC. Simulated annealing based economic dispatch algorithm. IEE Proc Gener Transm Distrib 1993;140(6):509–15.

[12] Walter DC, Sheble GB. Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans Power Syst 1993;8(August):1325–32.

[13] Cheng PH, Chang HC. Large scale economic dispatch by genetic algorithm. IEEE Trans Power Syst 1995;10(4):1919–26.

[14] Yang HT, Yang PC, Huang CL. Evolutionary programming based economic dispatch for units with nonsmooth fuel cost functions. IEEE Trans Power Syst 1996;11(February):112–8.
[15] Gaing Z-L. Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 2003;18(3):1187–95.
[16] Chaturvedi KT, Pandit M, Srivastava L. Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch. Int J Electr Power Energy Syst 2009;31(6):249–57.
[17] de Castro LN, Von Zuben FJ. Artificial immune systems: Part I – basic theory and applications. Technical report, TR-DCA 01/99. December 1999.
[18] de Castro LN, Zuben FJ. Learning and optimization using through the clonal selection principle. IEEE Trans Evol Comput 2002;6(3):239–51.
[19] Nicosia G, Cutello V, Bentley PJ, Timmis J. Artificial immune systems. In: Third international conference (ICARIS 2004), vol. 3239. Catania, Italy: LNCS, Springer-Verlag; 2004.
[20] Cutello V, Morelli G, Nicosia G, Pavone M. Immune algorithms with aging operators for the string folding problem and the protein folding problem. EvoCOP 2005, vol. 3448. Heidelberg: LNCS, Springer; 2005. p. 80–90.
[21] Cutello V, Narzisi G, Nicosia G, Pavone M. Real coded clonal selection algorithm for global numerical optimization using a new inversely proportional hypermutation operator. In: The 21st annual ACM symposium on applied computing, SAC 2006, vol. 2. Dijon, France, April 23–27. New York: ACM Press; 2006. p. 950–4.
[22] Cutello V, Nicosia G, Romeo M, Oliveto PS. On the convergence of immune algorithms. In: The first IEEE symposium on foundations of computational intelligence, FOCI 2007. Los Alamitos: IEEE Computer Society Press; 2007.
[23] Rahman TKA, Suliman SI, Musirin I. Artificial immune – based optimization technique for solving economic dispatch in power systems. Berlin Heidelberg: Springer-Verlag; 2006 [pp. 338–345].
[24] Liao Gwo-Ching. Short-term thermal generation scheduling using improved immune algorithm. Electr Power Syst Res 2006;76:360–73.
[25] Panigrahi BK, Yadav SR, Agrawal S, Tiwari MK. A clonal algorithm to solve economic load dispatch. Electr Power Syst Res 2007;77:1381–9.
[26] Huang TL, Hsiao YT, Chang CH, Jiang JA. Optimal placement of capacitors in distribution systems using an immune multi-objective algorithm. Int J Electr Power Energy Syst 2008;30(3):184–92.