3rd National Conference on Electrical & Computer Engineering
Distributed systems & Smart Grids
سومین کنفرانس ملی برق و کامپیوتر
سیستم های توزیع شده و شبکه های هوشمند
۱۸ آذر ماه ۱۳۹۵

Elitist Imperialist Competitive Algorithm: An Improved Performance version of Imperialist Competitive Algorithm

Mahdi Ghorbani*
Department of Computer science
Kashan branch Islamic Azad University
Kashan, Iran
m.ghorbani@iaukashan.ac.ir

Mahdi Esmaeili
Department of Computer science
Kashan branch Islamic Azad University
Kashan, Iran
m.esmaeili@iaukashan.ac.ir

Mohammad Asadi
Department of Computer science
Kashan branch Islamic Azad University
Kashan, Iran
Mo.asadi@gmail.com

***Abstract*—Recently, meta-heuristic optimization algorithms are used to find optimal solutions in huge search spaces. One of the most recent is Imperialist Competitive Algorithm (ICA) which is widely used in many optimization problems and has successful results. We add some elitism to ICA and introduced Elitist Imperialist Competitive Algorithm (EICA) as a new version of ICA.**

One of the most important application of optimization techniques is in data mining where clustering and its most popular algorithm, k-means, is a challenging problem. Its performance depends on the initial state of centroid and may trap in local optima. It is shown that the combination of EICA and k-means have better performance in terms of clustering and experimental results are discussed on k-means clustering. The goal of this research is to improve ICA for any optimization problem.

Keywords-Optimization Techniques, Evolutionary Computation, Meta-heuristics Algorithm, Imperialist Competitive Algorithm, ICA, EICA, Data Mining, K-means Clustering.

I. Introduction

Meta-heuristic algorithms are the most widely used algorithms for optimization which are commonly nature-inspired. As we can see from many case studies presented in this paper, they have many advantages compared to conventional algorithms. There are a few recent books which are solely dedicated to meta-heuristic algorithms [1,2,3]. Meta-heuristic algorithms are very diverse, including ant colony optimization(ACO) [4], bee's algorithm (BA) [5,6,7], cultural algorithm(CA) [8,9], genetic algorithm(GA) [10,11,12], particle swarm optimization (PSO) [13,14], imperialist competitive algorithm(ICA) [15], etc.

Two important characteristics of meta-heuristic are intensification and diversification [16]. Intensification intends to search locally and more intensively, while diversification makes sure the algorithm to explore the whole search space. A fine balance between these two is very important to the overall efficiency and performance of an algorithm. Too little exploration and too much exploitation could cause the system to be trapped in local optima, which makes it very difficult or even impossible to find the global optimum. On the other hand, if there is too much exploration and too little exploitation, it may be difficult for the system to converge and will slow down the overall search performance. A proper balance itself is an optimization problem, and one of the main tasks of designing new algorithms is to find an optimal balance by trade-off.

This paper is organized as follow: In Section 2 we explain imperialist competitive algorithm and describe its structure. In Section 4 we introduce our proposed method to improve the performance of ICA. In Section 3 we look at clustering problems in k-means algorithm. In Section 5 we evaluate our method by k-means clustering using different data sets and compare the results with some of the most popular optimization algorithms. Finally, in Section 6 we represent conclusions and some suggestions for future works.

II. IMPERIALIST COMPETITIVE ALGORITHM (ICA)

Inspired by the nature optimization algorithms, have succeed among other classic methods as intelligent optimization methods. Some of the most famous methods are Genetic Algorithms (GA) [5-7] (Inspired by biological evolution of human and other species), Ant Colony Optimization (ACO) [8] (based on optimized movement of ants) and Simulated Annealing (SA) [9] (inspired by annealing process for metal logy). These methods are used to resolve optimization issues in different fields such as determination of optimized path for automatic agents, designing optimized controllers for industry, resolving queue problems and clustering.

ICA is one of the relatively new meta-heuristics optimization algorithms which propose a method to resolve optimization by mathematical modeling of socio-politically evolution process. [10] Same as all algorithms in this category, ICA provides initial population and evaluates them by Eq. (1). This population is known as "Chromosome" in GA, "Particle" in Particle Swarm Optimization (PSO) and "country" in ICA. Basic principles of this algorithm are assimilation, imperialist competition and revolution. Simulating social, economic and political evolution of countries and providing operators as algorithms, ICA helps us to resolve complicated optimization. In fact, this algorithm constructs empires based on countries, calculates costs by Eq. (2) and finally tries to reach optimum result by a recursive process and optimizing the population gradually.

$$country = \left[p_1.p_2.p_3.\dots.p_{N_{imp}}\right]$$
$$cost = f(country) = f\left(p_1.p_2.p_3.\dots.p_{N_{imp}}\right)$$
$$C_n = c_n - max_i\{c_i\}P_n = \left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right| \quad (1)$$

$$TC_n = Cost(imperialist_n) + \zeta.mean\{Cost(colonies\ of\ empire_n)\} \quad (2)$$

A. *Assimilation: Moving Colonies toward the Imperialist*

According to the algorithm, countries are divided to imperialists and colonies. Considering its power, every imperialist absorbs some of colonies and take them under control. Assimilation is one of the main two principals of this algorithm. Studying the history of grate? imperialists like France and England, they usually tried to wipe out traditions and cultures of colonies by some methods such as constituting schools which uses their languages. This process represented in the algorithm by moving colonies of an empire based on a special equation. Fig.1 and Fig.2 show this movement and variables are defined by Eq.(3), where β is a number greater than 1 and d is the distance between the colony and the imperialist state. Setting $\beta > 1$ causes colonies to get closer to the imperialist state, γ is a parameter that adjusts the deviation from the original direction. Nevertheless, the values of β and γ are arbitrary, in most of implementations setting about 2 for β and about $\pi/4$ (Rad) for γ results in good convergence of countries to the global minimum.

$$x \sim U(0.\beta * d). \quad \theta \sim U(-\gamma.\gamma) \quad (3)$$

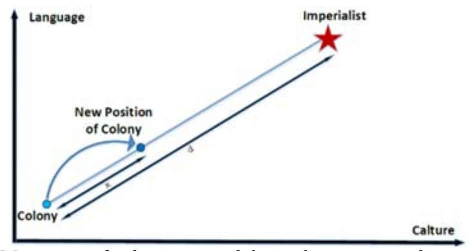

Fig 1: Movement of colonies toward their relevant imperialist

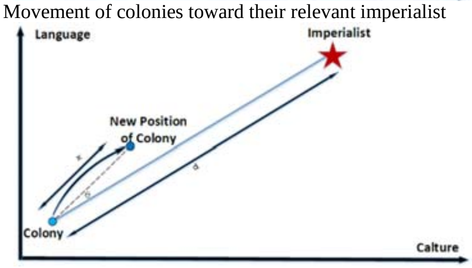

Fig 2: Movement of colonies toward their relevant imperialist in a randomly deviated direction

B. *Imperialist Competition*

Imperialist competition is the other important issue of this algorithm. Though competition weak empires gradually lost their power and eventually will be eliminated. This competition leads to a state in which single empire rules the world. This state happens when algorithm reaches optimum solution and stops. Eq. (4) shows calculation method of this process and imperialist competition diagram is shown in Fig.3.

$$P = \left|\frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_i}\right|$$
$$P = \left[P_{p_1}.P_{p_2}.\dots.P_{p_{N_{imp}}}\right]$$
$$R = \left[r_1.r_2.\dots.r_{N_{imp}}\right].where\ r_i \approx U(0.1)\ and\ 1 \le i \le N_{imp}$$
$$D = P - R = \left[D_1.D_2.\dots.D_{N_{imp}}\right]$$
$$= \left[P_{p_1} - r_1.\dots.P_{p_{N_{imp}}} - r_{N_{imp}}\right] \quad (4)$$

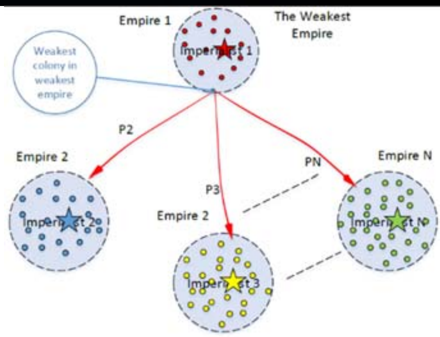

Fig 3: imperialist competition diagram

C. *Revolution*

Revolution causes radical social and political changes in a country. In ICA, revolution is modeled with random movement of a colony to a new position. Revolution saves movements from trapping in local optimums and in some cases improves the position of the country and moves it to a better area. This action is shown in Fig.4.

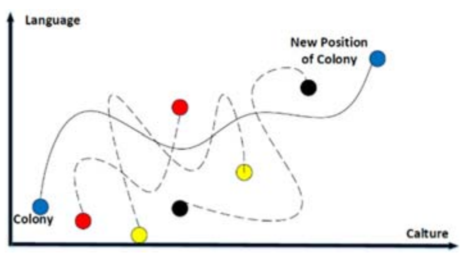

Fig 4: Revolution; radical change in socio-political characteristics of a country

Please be sure your sentences are complete and that there is continuity within your paragraphs. Check the numbering of your graphics and make sure that all proper references are included.

III. Clustering and K-means Algorithm

Clustering can be considered as the most important task in unsupervised learning. It's about finding a structure inside an unlabeled data set. Cluster is a set of similar data. Clustering process tries to put data with maximum similarity inside one cluster and to minimize the similarity between data in different clusters. Fig. 5 shows a sample of data clustering.

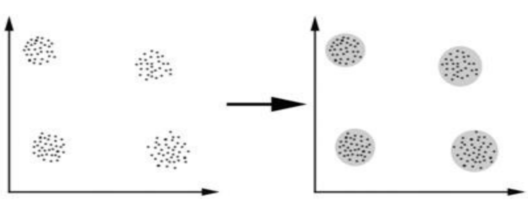

Fig. 5 This figure shows a sample of clustering which uses distance as a factor for dissimilarity.

Clustering algorithms can be grouped into two main classes of algorithms, namely supervised and unsupervised. Most of these algorithms group data into clusters independent of the topology of data space. One of the most famous one is K-means [22,23]. K-means is a clustering method which uses Loyd's algorithm. Some of the improved variations of the K-means algorithm can be found in [24,25]. Despite of implicitly, this method is the basis for many other clustering methods (like fuzzy clustering) and is exclusive and at. There are many different structures for this algorithm but they all have the same routine
which estimates the followings:

Centers of clusters: These points mostly are the mean points of clusters. A point belongs to a cluster if it has the minimum distance from its center. In simple implementations of this algorithm, for n data points, first a certain
number of points (k) are selected as centers randomly. Then other $n-k$ points join to centers based on their similarity and consequently new clusters are created. One can calculate new mean as new centers and construct new clusters
for each iteration. This process continues until no changes are made to the centers. The following function is considered as the goal function.

The number of clusters: The best clustering method is one which maximizes the similarity of intra-cluster points and minimizes it between clusters central points. To have the best clusters first a range is proposed for k based on experience. Then for each selected k, $p(k)$ is calculated. The optimum value for k is one which has the maximum value of $p(k)$.
Eq. 5 describes the quality of clustering for k points, where O is a set of central points of clusters, C^n is the central point of a cluster, O^n is a set of non-central points, Tc is a set of data which clustering is performing on, η_n is the mean similarity for centers in C^n and those in O^n, η_m is the mean similarity for centers in C^m and those in O^m and finally $\delta_{nm}$is the similarity
if C^n and O^n.

$$O \;=\{C^n|n=1.\dots.k\}$$
$$O^n=\{C_i|i=1.\dots.||T^C-O||\}$$
$$p(k)=\frac{1}{k}\sum_{n=1}^{k}\left(\min\left\{\frac{\eta_n+\eta_m}{\delta_{nm}}\right\}\right)$$
$$\eta_n=\frac{1}{||O^n||}\sum_{C_i\in O^n} sim(C_i.O^n)$$
$$\eta_m \;=\frac{1}{||O^m||}\sum_{C_j\in O^m} sim(C_j.O^m)$$
$$\eta_{nm}=\;sim(C^n.O^m) \qquad (5)$$

In general, there are two considerations in K-means clustering method, first is to set the value of k properly as the number of clusters, and second is to specify exactly which points in the search space is belong to k clusters. In this paper we focus on second challenge, finding the best points. To do this, we use our proposed method (EICA) and compare the results with the best meta-heuristic optimization algorithms.

IV. Proposed Method Elitist Imperialist Competitive Algorithm(EICA)

In recent years, ICA as an algorithm, has been used for many optimization problems among all similar methods. Referring to applied samples of [17,26,27,28,29,30,31], it can be found that either basic or improved versions have been used.

After proposing base ICA algorithm, there has been different development of this algorithm and each of them improved the performance in their own specific way. As an example [32] or defining an operator for mutation to change the movement of imperialists tries to improve ICA for continuous problems. Also a research by [33] improved ICA to resolve constrained optimization problems by defining a classic penalty function. All these versions try to improve ICA by adding new operators and functions. We will not add any operator to the base algorithm. We will improve the performance of ICA for any application by changing in assimilation and revolution which are two base operations of this algorithm.

Change in Assimilation: In the base algorithm and other versions the assimilation which assigns some colonies to the imperialists is carried out by Eq. 6 but this research shows that if we use Eq. 7 for assimilation it will cause more stability and finally the performance will be improved.

$$\bar{x} = x + \beta * (t - x). 0 \le \beta \le 2 \qquad (6)$$

$$\bar{x} = x + \beta * r * (t - x). 0 \le \beta \le 0.8 \qquad (7)$$

Where x is the current position of a colony, $\bar{x}$ is the new position of a colony, t is the position of imperialist which colony is moving toward, β is assimilation coefficient which considered constant and r is learning coefficient vector which has a random value between 0 and 1.

Change in Revolution Policy: In the base algorithm and other versions, revolution is carried out by normal distribution according to Eq. 8, but in this research we add elitism to this process using normal distribution around an optimized parameter. Eq. 9 shows this method and it is used in any iteration using previous stages.

$$\bar{x} = \sim (X_{min}. X_{max}) \qquad (8)$$

$$X_{min} \le x. \bar{x} \le X_{max}$$
$$\bar{x} \sim N(x. \sigma^2) \sim x + \sigma N(0.1)$$
$$\sigma = \eta(x_{max} - x_{min}). \eta = 0.1 \qquad (9)$$

Where X_{min} and X_{max} are the minimum and maximum values which any country can set its position around, x is the current position, $\bar{x}$ is new position after the revolution, σ specifies the step size and η is the width of search space. Fig. 6 shows the graphical view of revolution by base method and optimized one.

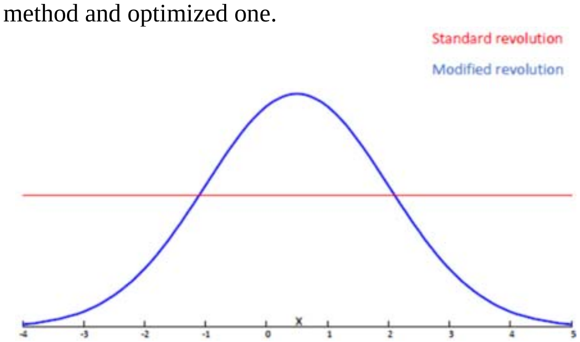

Fig. 6 Normal Revolution.

According to the law of schewefel's $\frac{1}{5}$, at any stage of the algorithm if the percentage of successful evolutions is more than 20%,step length σ will be increased, otherwise it will be decreased. σ is used before in CMA-ES and GA algorithms, therefore the appliance of our method has been proven.

According to the base algorithm, the revolution applies only to colonies and can only change the position of colonies to achieve better exploration and finally a better position, but in this research the revolution is applied to the imperialists as well and according to the law explained in the following we will show that the results are much better than base method. Considering success rate of revolution is the key. If the revolution on an imperialist lead to a better result, it will be accepted and the process will continue, otherwise it will be rejected and the imperialist will return to its previous position.

V. Experimental results

In this section, we will use experimental results to show the clustering performance of EICA. All experiments are implemented on a computer with Intel Pentium® CPU 3.00GHz, 8GB of memory and 64Bit windows 8 operating system. All algorithms and data are implemented by matlab 8.3. To find optimized results the experiment was 10 times on every data sets and results are shown in Table. 3 to Table. 6. We solve k-means clustering with 4 best algorithms EICA, ICA, GA, and PSO and compare the results with charts and represent them in Fig. 7 - Fig. 10 and show that EICA is better.

A. Data sets

To evaluate the proposed method, we need data sets specialized for clustering. In this paper we will use data sets from U.C.I. repository [34]. The numbers of 5 completely different data sets are considered. They are suitable for mentioned method, EICA, and can consider the efficiency of the method with high accuracy. Table. 1 shows the statistics of these 5 data sets.

Table 1. Statistics of data.

Data Set	Classes	Attribute	Instances
Iris	3	4	150
Wine	3	13	178
Glass	6	9	214
CMC	3	9	1473
Cancer	2	9	699

We have used five criteria to evaluate the performance of algorithms: (I) the sum of the intra-cluster distances from center have been summarized in Table. 3. (II) the sum of the clusters centers distance have been summarized in Table. 4. (III) the sum of the intra-cluster distances from each point have been summarized in Table. 5. the sum of the inter-cluster distances (IV) have been summarized in Table. 6. (V) the number of fitness function evaluations.

For criterion (I), (II), (III), note that the smaller the sum of the distances, the higher quality of clustering, but in criterion (IV) the higher value is better for quality of the clustering, and in criterion (V), the smaller value for the number of function evaluations indicates the high convergence speed of the algorithm. Since all of these algorithms are stochastic, to counteract randomized nature of them and to indicate the consistency and robustness of algorithms, 20 independent executions were conducted for each experiment. The results are the best, worst, and average achieved from 20 simulations. The last criterion in all Tables is the number of function evaluations (coded as NFE), which indicates the convergence speed of the respective algorithms. NFE is the number of times that the clustering algorithm has calculated the fitness function Eq. 1 to reach the (near) optimal solution. It is dependent on the number of iterations to reach the optimal solution.

VI. Conclusion

In this paper, combined a few meta-heuristic optimization algorithms which are more fortunate than others (for better solutions that have shown) with the k-means clustering and compare the results by five important parameters. Also we realized that if we add some elitism to ICA, we can reach better results.

The results showed that in most cases, EICA is better in both cost reduction and NFE, while the implementation of these algorithm are also having less computational complexity. Also, when the number of examples and the number of features used in the data set is too high (for example CMC and Cancer data set), EICA show more reassuring answers.

It should be noted that the use of meta-heuristic optimization algorithms in the field of data clustering based on density which is time consuming and computationally heavy, is very successful. To determine and set parameters of algorithms in this category such as DBSCAN, DENCLU, SOM, SOFM, we can help strengthen the capacity of optimization algorithms.

Table 2. Parameter settings of applied algorithms.

Algorithm	Control Parameters
ICA and EICA	$X = Dataset. k = No\ Cluster. VarSize = (k.Size(X.2)). nVar = product(VarSize). nCountry = 50. nEmp = 15. \alpha = 1. \beta = 2. \zeta = 0.1. pRevolution = 0.05. \mu = 0.02. MaxIt = 200. VarMin = min(X). VarMax = \max(X)$
GA	$X = Dataset. k = No\ Cluster. VarSize = (k.Size(X.2)). nVar = product(VarSize). nPop = 50. pc = 0.8.$ $nc = 2 * round\left(pc * \frac{nPop}{2}\right). pm = 0.3. nm = round(pm * nPop). \gamma = 0.05. \mu = 0.02. \beta = 8. MaxIt = 200.$ $VarMin = min(X). VarMax = \max(X)$
ACO	$X = Dataset. k = No\ Cluster. VarSize = (k.Size(X.2)). nVar = product(VarSize). nAnt = 50.$ $q = 1. \tau_0 = 1. \tau = (1-\rho) * \tau. \alpha = 1. \rho = 0.05. \zeta = 1. MaxIt = 200. VarMin = min(X). VarMax = \max(X)$
BA	$X = Dataset. k = No\ Cluster. VarSize = (k.Size(X.2)). nVar = product(VarSize). nBee = 50. nBee0 = round$ $(0.3 * nBee). r = 0.1 * (VarMax - VarMin). rDamp = 0.99. MaxIt = 200. VarMin = min(X). VarMax\max(X)$
CA	$X = Dataset. k = No\ Cluster. VarSize = (k.Size(X.2)). nVar = product(VarSize). nInd = 50. pAccept = 0.35.$ $nAccept = round(pAccept * nInd). \alpha = 0.25. \beta = 0.5. MaxIt = 200. VarMin = min(X). VarMax = \max(X)$
PSO	$X = Dataset. k = No\ Cluster. VarSize = (k.Size(X.2)). nVar = product(VarSize).$ $\text{nParticle} = 50. \phi_1 = 2.05. \phi_1 = 2.05. \phi = \phi_1 + \phi_2. \chi = \frac{2}{\phi - 2 + \sqrt{\phi_2 - 4 * \phi}}. w = \chi. wDamp = 1.$ $c_1 = \chi * \phi_1.\ c_2 = \chi * \phi_2. \alpha = 0.1. VelMax = \alpha * (VarMax - VarMin). VelMin = -VelMax$ $MaxIt = 200. VarMin = min(X). VarMax = \max(X)$

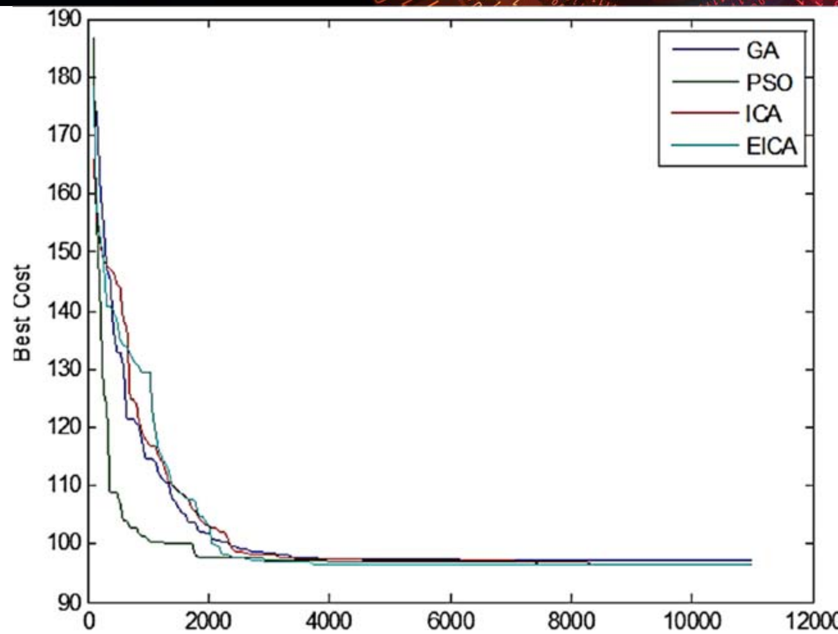

Fig. 7 k-means clustering over Iris data set:

EICA (optimum state: 96.6555 and NFE:4280), ICA (optimum state: 96.668 and NFE:10001),

GA (optimum state: 96.6657 and NFE:11050), PSO (optimum state: 96.6601 and NFE:5800).

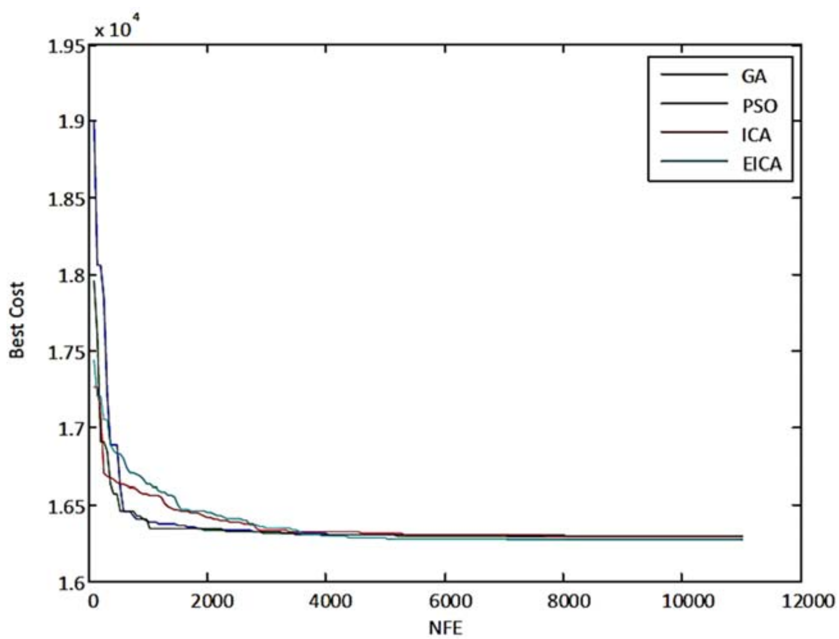

Fig. 8 k-means clustering over Wine data set:

EICA (optimum state: 16292.1852 and NFE =10711), ICA (optimum state: 16292.8018 and NFE = 10711),

GA (optimum state: 16294.3779 and NFE:10830), PSO (optimum state: 16294.7715 and NFE = 10050)

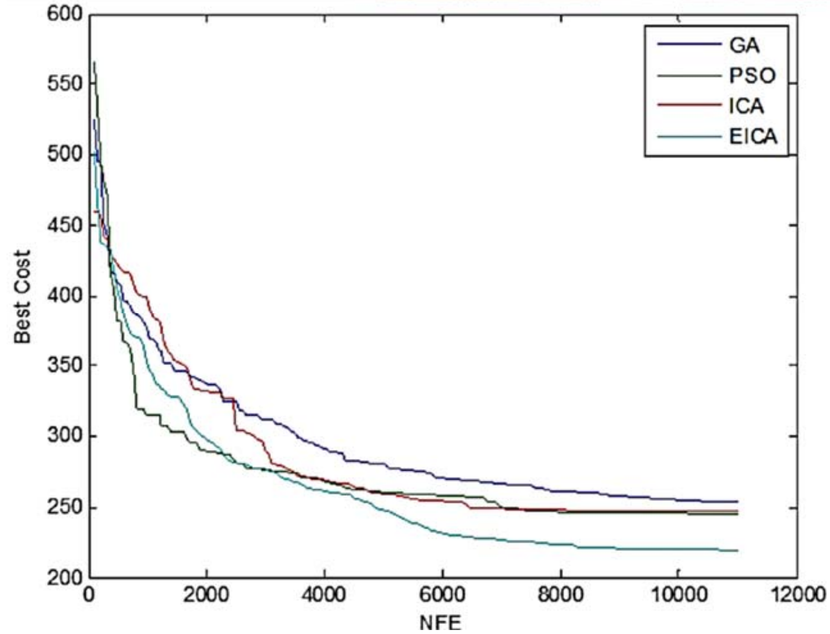

Fig. 9 k-means clustering over Glass data set:

EICA (optimum state: 215.7581 and NFE = 11022), ICA (optimum state: 219.8018 and NFE = 11071),

GA (optimum state: 225.3520 and NFE:10830), PSO (optimum state: 220.4815 and NFE = 10050)

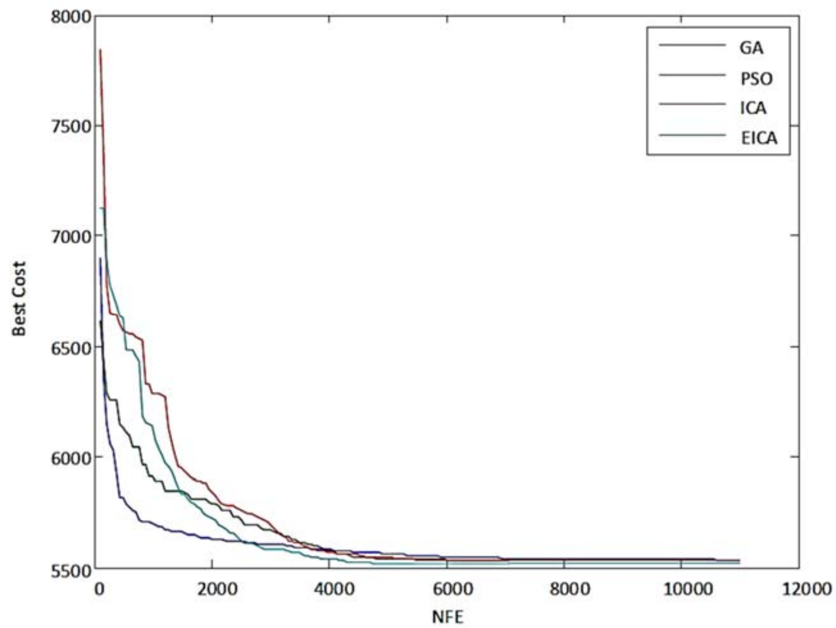

Fig. 10 k-means clustering over CMC data set:

EICA (optimum state: 5532.184 and NFE = 10821), ICA (optimum state: 5532.192 and NFE = 10821),

GA (optimum state: 5532.330 and NFE:10010), PSO (optimum state: 5532.190 and NFE = 10050)

Table 3. Parameter Evaluate results for clustering algorithms on criterion (I). for each data set and depending on the cost (total distance within the cluster from centers) and number of NFE, algorithms provide different answers. As it shown, for all data sets EICA algorithms generate best results.

Data set	Criteria	Iris	Wine	Glass	CMC	Cancer
k-means	Best	97.1233	16313.7	213.4205	5541.001	3050.244
	Average	97.2651	17172.16	222.9871	5543.784	3056.382
	Worst	97.3462	18996.49	240.0353	5545.778	3056.943
	NFE	8	7	11	6	5
ACO	Best	96.6656	16295.12	246.3942	5537.364	3036.721
	Average	96.6686	16358.71	253.706	5577.27	3038.877
	Worst	96.6699	16706.37	263.5694	5600.661	3040.484
	NFE	16700	13850	19500	9250	9800
BA	Best	96.6662	16311.96	244.0362	5583.011	3038.516
	Average	96.6672	16332.3	258.2563	5585.685	3039.695
	Worst	96.6677	16348.74	270.1917	5590.833	3044.103
	NFE	262401	293388	281512	122651	143994
CA	Best	96.6677	16302.62	280.5867	5608.722	3108.493
	Average	97.08	16323.42	294.8646	5729.693	3194.267
	Worst	99.7877	16387.14	313.3266	5839.678	3347.071
	NFE	12800	16700	19500	9850	10000
GA	Best	96.6657	16292.4	213.8378	5532.32	3035.482
	Average	97.0238	16294.3	232.352	5538.773	3036.067
	Worst	97.4609	16296.46	253.7388	5545.99	3038.254
	NFE	12050	13500	16600	9600	13500
PSO	Best	96.6601	16292.19	210.472	5532.185	3035.423
	Average	96.6603	16292.28	237.4815	5532.187	3035.424
	Worst	96.6665	16292.67	253.4747	5532.189	3035.427
	NFE	6050	10050	10050	10850	10050
ICA	Best	96.668	16292.19	210.8626	5532.185	3035.423
	Average	96.67	16293.09	239.0157	5532.188	3035.453
	Worst	96.675	16294.25	257.9678	5532.192	3035.554
	NFE	6607	10711	10616	10821	9669
EICA	**Best**	**96.6555**	**16292.19**	**210.4625**	**5532.184**	**3035.423**
	Average	96.6558	16292.87	238.0137	5532.187	3035.443
	Worst	96.6562	16293.25	255.9608	5532.19	3035.453
	NFE	6607	10711	10616	10821	9669

Table 4. Evaluate results for clustering algorithms on criterion (II). Considering both criteria of cost and NFE observed that EICA is more efficient algorithm for all data sets.

Data set	Criteria	Iris	Wine	Glass	CMC	Cancer
k-means	Best	9.7968	1347.6057	76.4193	39.2746	14.6703
	Average	10.1018	1480.7672	81.3271	39.8061	13.8346
	Worst	10.167	1578.7291	86.4655	40.0293	13.85
	NFE	8	7	11	6	5
ACO	Best	9.8732	1345.9652	84.6615	39.5574	14.0651
	Average	9.876	1366.4868	95.7412	38.7219	14.1961
	Worst	9.8777	1470.0117	110.1353	39.8646	14.3086
	NFE	16700	13850	19500	9250	9800
BA	Best	9.8679	1343.9664	77.8679	39.5982	14.0663
	Average	9.8771	1357.7392	88.8494	38.9844	14.1551
	Worst	9.8834	1371.0548	105.0083	39.6917	14.2361
	NFE	262401	293388	281512	122651	143994
CA	Best	9.5631	1342.7952	78.6714	39.3229	14.8295
	Average	9.6632	1362.8522	94.1693	39.2327	13.6205
	Worst	9.968	1377.6095	108.4679	39.7486	14.9257
	NFE	12800	16700	19500	9850	10000
GA	Best	9.7928	1340.4635	70.9498	39.0207	14.1059
	Average	9.6183	1362.974	80.4227	38.4794	14.1685
	Worst	9.8753	1379.8775	99.3044	38.9739	14.2291
	NFE	12050	13500	16600	9600	13500
PSO	Best	9.8761	1347.6374	75.003	39.8971	14.2143
	Average	9.8761	1350.9732	88.5796	38.9016	14.2164
	Worst	9.8761	1364.2792	102.5854	38.9112	14.2213
	NFE	6050	10050	10050	10850	10050
ICA	Best	9.8495	1345.7184	72.1494	39.8995	14.1911
	Average	9.8734	1361.1982	89.7198	38.9022	14.2092
	Worst	9.8761	1375.9346	110.2368	38.9051	14.2196
	NFE	6607	10711	10616	10821	9669

EICA	**Best**	**9.4595**	**1340.6182**	**68.1392**	**38.7965**	**14.1321**
	Average	9.8034	1344.1906	89.132	38.8212	14.2001
	Worst	9.8261	1350.9126	106.3685	38.9001	14.212
	NFE	6607	10711	10616	10821	9669

Table 5. Evaluate results for clustering algorithms on criterion (III). examines the cost (total distance of all points in one cluster to another) and NFE. EICA algorithms has better results.

Data set	Criteria	Iris	Wine	Glass	CMC	Cancer
k-means	Best	3419.197	639765.8	9292.381	1916670	702989.7
	Average	3507.261	746496.7	11798.72	1937769	704601.1
	Worst	3529.349	1077924	17407.24	1956012	704685.9
	NFE	8	7	11	6	5
ACO	Best	3499.192	639765.8	13270.87	1914519	702989.7
	Average	3499.192	640871.5	18241.1	1921463	704007.4
	Worst	3499.192	641440.5	21350.67	1934295	704685.9
	NFE	16700	13850	19500	9250	9800
BA	Best	3499.192	639765.8	12058.64	1918313	702989.7
	Average	3499.192	640865.3	18647.48	1922861	703272.4
	Worst	3499.193	641440.5	21348.43	1928413	704685.9
	NFE	262401	293338	281512	122651	143994
CA	Best	3417.51	639765.8	19089.57	1916733	698924.3
	Average	3490.229	640701.6	20963.7	1929695	706671
	Worst	3756.844	641440.5	22829.39	1953212	731784.4
	NFE	12800	16700	19500	9850	10000
GA	Best	3415.51	639765.8	9319.35	1914012	702989.7
	Average	3458.81	640300	13413.02	1929017	704346.7
	Worst	3499.192	641440.5	19170.52	1949590	704685.9
	NFE	12050	13500	16600	9600	13500
PSO	Best	3499.192	640824.4	9319.889	1919441	704685.9
	Average	3499.192	641317.2	16785.5	1919441	704685.9
	Worst	3499.192	641440.5	21146.25	1919441	704685.9
	NFE	6050	10050	10050	10850	10050
ICA	Best	3413.368	639765.8	10730.98	1919441	702989.7
	Average	3425.51	640685.4	16415.48	1919441	704177.1
	Worst	3439.192	641440.5	21274.96	1919441	704685.9
	NFE	6607	10711	10616	10821	9669
EICA	**Best**	**3412.307**	**639755.8**	**9217.892**	**1914011**	**698323.6**
	Average	3422.464	640685.1	16415.22	1919441	704075
	Worst	3436.495	641440.1	21274.75	1919442	704485
	NFE	6607	10711	10616	10821	9669

Table 6. Evaluate results for clustering algorithms on criterion (IV). examines the sum distance of all points within a cluster with others. Higher Cost and smaller NFE shows the efficiency of the algorithm. For all data sets, EICA provides best **results.**

Data set	Criteria	Iris	Wine	Glass	CMC	Cancer
k-means	Best	35323.75	6756751	84655.32	14117767	1739973
	Average	35087.89	6439438	77958.2	14027226	1738375
	Worst	35048.81	5483756	59898.13	13952953	1738290
	NFE	8	7	11	6	5
ACO	Best	35155.91	6759431	71047.32	14121274	1739973
	Average	35155.91	6755807	58468.14	14084790	1738964
	Worst	35155.91	6747827	52299.21	14030278	1738290
	NFE	16700	13850	19500	9250	9800
BA	Best	35155.91	6756751	76004.73	14096295	1739973
	Average	35155.91	6754301	57486.35	14079805	1739693
	Worst	35155.91	6751877	51459.77	14049882	1738290
	NFE	262401	293388	281512	122651	143994
CA	Best	35316.39	6759433	54462.21	14129305	1744600
	Average	35054.89	6755916	51609.16	14056639	1736316
	Worst	33379.8	6751877	47944.75	13961656	1711371
	NFE	12800	16700	19500	9850	10000
GA	Best	35326.39	6759433	85168.6	14127654	1739973
	Average	35184.77	6756039	72053.83	14047304	1738627
	Worst	34931.21	6751877	55503.74	13957414	1738290
	NFE	12050	13500	16600	9600	13500
PSO	Best	35155.91	6756698	85271.87	14092501	1738290
	Average	35155.91	6752841	62942.73	14092501	1738290
	Worst	35155.91	6751877	52108.18	14092501	1738290

	NFE	6050	10050	10050	10850	10050
ICA	Best	35289	6759434	80993.96	14092501	1739973
	Average	35250.22	6758404	63713.26	14092501	1738795
	Worst	35255.91	6754877	51722.94	14092501	1738290
	NFE	6607	10711	10616	10821	9669
EICA	**Best**	**35334.2563**	**6759435.996**	**85275.3625**	**14142600.362**	**1745975**
	Average	35165.13	6758405	85115.36	14122600	1741596
	Worst	35260.24	6757983	72635.7	14092600	1739036
	NFE	6607	10711	10616	10821	9669

REFERENCES

[1] X.-S. Yang, Nature-inspired metaheuristic algorithms, First Edition, Luniver press, 2008.

[2] X.-S. Yang, Engineering optimization: an introduction with metaheuristic applications, John Wiley & Sons, 2010.

[3] E.-G. Talbi, Metaheuristics: from design to implementation, Vol. 74, John Wiley & Sons, 2009.

[4] A. Colorni, M. Dorigo, V. Maniezzo, et al., Distributed optimization by ant colonies, in: Proceedings of the first European conference on artificial life, Vol. 142, Paris, France, 1991, pp. 134–142.

[5] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The bees algorithm. technical note, Manufacturing Engineering Centre, Cardiff University, UK (2005) 1–57.

[6] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005).

[7] D. T. Pham, M. Castellani, the bee's algorithm: modelling foraging behavior to solve continuous optimization problems, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 223 (12) (2009) 2919–2938.

[8] R. G. Reynolds, An introduction to cultural algorithms, in: Proceedings of the third annual conference on evolutionary programming, Singapore, 1994, pp. 131–139.

[9] R. G. Reynolds, B. Peng, Knowledge learning and social swarms in cultural systems, Journal of Mathematical Sociology 29 (2) (2005) 115–132.

[10] M. C. Cowgill, R. J. Harvey, L. T. Watson, A genetic algorithm approach to cluster analysis, Computers & Mathematics with Applications 37 (7) (1999) 99–108.

[11] M. Mitchell, An introduction to genetic algorithms, MIT press, 1998.

[12] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence., U Michigan Press, 1975.

[13] J. Kennedy, Particle swarm optimization, in: Encyclopedia of Machine Learning, Springer, 2010, pp. 760–766.

[14] C.-Y. Chen, F. Ye, Particle swarm optimization algorithm and its application to clustering analysis, in: Networking, Sensing and Control, 2004 IEEE International Conference on, Vol. 2, IEEE, 2004, pp. 789–794.

[15] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: Evolutionary computation, 2007. CEC 2007. IEEE Congress on, IEEE, 2007, pp. 4661–4667.

[16] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, ACM Computing Surveys (CSUR) 35 (3) (2003) 268–308.

[17] T. Niknam, E. T. Fard, N. Pourjafarian, A. Rousta, An efficient hybrid algorithm based on modified imperialist competitive algorithm and kmeans for data clustering, Engineering Applications of Artificial Intelligence 24 (2) (2011) 306–317.

[18] C. Lucas, Z. Nasiri-Gheidari, F. Tootoonchian, Application of an imperialist competitive algorithm to the design of a linear induction motor, Energy conversion and management 51 (7) (2010) 1407–1411.

[19] A. Kaveh, S. Talatahari, Optimum design of skeletal structures using imperialist competitive algorithm, Computers & Structures 88 (21) (2010) 1220–1229.

[20] M. H. Moradi, A. Zeinalzadeh, Y. Mohammadi, M. Abedini, An efficient hybrid method for solving the optimal sitting and sizing problem of dg and shunt capacitor banks simultaneously based on imperialist competitive algorithm and genetic algorithm, International Journal of Electrical Power & Energy Systems 54 (2014) 101–111.

[21] S. Kirkpatrick, M. P. Vecchi, et al., Optimization by simmulated annealing, science 220 (4598) (1983) 671–680.

[22] E. W. Forgy, Cluster analysis of multivariate data: efficiency versus interpretability of classifications, Biometrics 21 (1965) 768–769.

[23] J. A. Hartigan, Clustering algorithms, John Wiley & Sons, Inc., 1975.

[24] A. M. Bagirov, modified global k-means algorithm for minimum sum of squares clustering problems, Pattern Recognition 41 (10) (2008) 3192–3199.

[25] S. J. Redmond, C. Heneghan, A method for initialising the k-means clustering algorithm using kd-trees, Pattern Recognition Letters 28 (8) (2007) 965–973.

[26] A. Hadidi, M. Hadidi, A. Nazari, A new design approach for shell-andtube heat exchangers using imperialist competitive algorithm (ica) from economic point of view, Energy Conversion and Management 67 (2013) 66–74.9

[27] S. H. Mirhoseini, S. M. Hosseini, M. Ghanbari, M. Ahmadi, A new improved adaptive imperialist competitive algorithm to solve the reconfiguration problem of distribution systems for loss reduction and voltage profile improvement, International Journal of Electrical Power & Energy Systems 55 (2014) 128–143.

[28] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian, H. R. Massrur, M. Gharibzadeh, Application of imperialist competitive algorithm with its modified techniques for multi-objective optimal power flow problem: a comparative study, Information Sciences 281 (2014) 225–247.

[29] M. A. Ahmadi, M. Ebadi, A. Shokrollahi, S. M. J. Majidi, Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir, Applied Soft Computing 13 (2) (2013) 1085–1098.

[30] L. D. Afonso, V. C. Mariani, L. dos Santos Coelho, Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization, Expert Systems with Applications 40 (9) (2013) 3794–3802.

[31] J.-L. Lin, H.-C. Chuan, Y.-H. Tsai, C.-W. Cho, Improving imperialist competitive algorithm with local search for global optimization, in: Modelling Symposium (AMS), 2013 7th Asia, IEEE, 2013, pp. 61–64.

[32] S. Xu, Y. Wang, P. Lu, Improved imperialist competitive algorithm with mutation operator for continuous optimization problems, Neural Computing and Applications (2015) 1–16.

[33] Y. Zhang, Y. Wang, C. Peng, Improved imperialist competitive algorithm for constrained optimization, in: Computer Science-Technology and Applications, 2009. IFCSTA'09. International Forum on, Vol. 1, IEEE, 2009, pp. 204–207.

[34] archive.ics.usi.edu.