Modeling an Artificial Bee Colony with Inspector for Clustering Tasks

Cosimo Birtolo, Giovanni Capasso, Davide Ronca, and Gennaro Sorrentino

Poste Italiane – Information Technology,
S - FSTI - R&D Center – Piazza Matteotti 3 – 80133 Naples, Italy
{birtoloc,capass56,roncadav,sorre137}@posteitaliane.it

Abstract. Artificial Bee Colony (ABC) is a recent meta-heuristic approach. In this paper we face the problem of clustering by ABC and we model a further bee role in the colony, performed by inspector bee. This model conforms with real honey bee colony, indeed, in nature some bees among the foraging ones are called inspectors because they preserve the colony's history and historical information related to food sources. We experiment inspector behavior in ABC and compare the solution to traditional clustering algorithm. Finally, the effect of colony size is investigated and experimental results are discussed.

Keywords: Artificial Bee Colony, Soft Computing, Clustering, Inspector, Data Mining.

1 Introduction

Clustering algorithms play a relevant role in understanding and exploring a dataset. Interest in clustering algorithm is proved by the need of knowledge extraction processes in a huge amount of data in several domains: from bioinformatics to web usage mining, from image segmentation to information retrieval.

Clustering aims at minimizing the dissimilarity between data assigned to the same cluster and is a powerful tool which can arise interesting information in the area it is applied. Moreover, clustering can be considered one of the most difficult and challenging problems in machine learning, particulary due to its unsupervised nature.

Clustering problems have been solved using various techniques, even if K-means, independently discovered in different scientific studies in the 60's [1], is one of the most popular algorithms due to its simplicity and efficiency. However, the clusters resulting from the K-means algorithm are very sensitive to positions of the initial centroids in the problem space and the algorithm can converge to a local optimum. Recently, meta-heuristic approaches have been proposed to solve clustering problems [2] and in particular some Artificial Bee Colony have been adopted for this task [3].

In this paper, we experiment ABC approach to clustering tasks by means of different datasets. The merits of this contribution are: (i) the introduction and the modeling of a new role in the colony, i.e., the inspector bee in an ABC

C. Blum and G. Ochoa (Eds.): EvoCOP 2014, LNCS 8600, pp. 182–193, 2014.

algorithm, and (ii) the experimentation on the way the colony size and its composition can influence the algorithm's results.

The remainder of this paper is organized as follows: Section 2 introduces the clustering; Section 3 describes Artificial Bee Colony (ABC); Section 4 depicts the proposed formulation of ABC; Section 5 describes the algorithm structure, Section 6 provides experimental results; and Section 7 outlines conclusions and future directions.

2 The Clustering Problem

Clustering algorithms aim at grouping data into a number of clusters. Data in the same cluster share a high degree of similarity while they are very dissimilar from data of other clusters. Partitional clustering algorithms aim at partitioning the population into a fixed number k of classes, each of those being represented by an average item named centroid. The traditional partitional clustering algorithm is K-means [1] which has been applied to a wide range of problems in different domains. However, K-means is sensitive to the initial states and can converge to the local optimum solution. Recently, many methods have been proposed in order to overcome this drawback.

The clustering problem can be stated as the minimization of the sum of Euclidean squared distance between each object x_i and the center of the cluster c_j to which it belongs (i.e., centroids). The objective function to be minimized can be expressed by Eq.1:

$$J(w,c) = \sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{d=1}^{D} w_{ij} \left\|x_{i,d} - c_{j,d}\right\|^2 \tag{1}$$

where K is the number of clusters, N is the number of objects, w_{ij} is the association weight of objects x_i in cluster j, i.e., $w_{i,j}$ is 1 if object i is allocated to cluster j, and 0 if it is not. Each data instance x_i and each cluster center c_j is defined by a vector of D values, where D is the number of features. The center of each of the j cluster $c_j = \{c_{j,1}, c_{j,2}, \ldots, c_{j,D}\}$ is the set of the mean of each dimension across all the objects assigned to the jth cluster and it can be calculated by Eq.2 below

$$c_{j,d} = \frac{1}{N_j}\sum_{i=1}^{N} w_{i,j} x_{i,d} \tag{2}$$

where N_j is the number of objects in the jth cluster.

Different evolutionary approaches are adopted to address clustering tasks (i.e., fixed or variable number of clusters, centroid-based, medoid-based, label-based, tree-based or graph-based representation) as described by Hruschka et al.[2], but recently some swarm intelligence techniques are proposed [3,4].

3 Artificial Bee Colony

Artificial Bee Colony (ABC) Algorithm is a recent swarm intelligence algorithm based on the intelligent behavior of honey bee foraging. It was proposed by Karaboga [5] in 2005 and performances are analyzed in 2008 [6]. ABC is based on modeling the behaviors of real bees on finding nectar amounts and sharing the information of food sources to the other bees in the hive.

Honey bees are social insects and live in large organized communities. Each bee has specific skills and carries out determined works with the aim of facilitating the survival of the colony. The provision of the food is one of the major activities within a colony. This activity involves specific worker bees which collaborate among each other: the "employed bees", which research and communicate where the food sources are; the "onlooker bees" which extract and carry the food. The main task of an employed bee is to look for food. When the food source has been found, the bee memorizes the spatial coordinates and communicates the position and the quality of the source through a dance around the hive. The dance and the research activity alternate each other. The main task of an onlooker bee is observing the employed bees dance outside the hive. On the basis of the message expressed by the dance, the onlooker bee chooses the food source that best fits its needs. After the choice, the onlooker bee reaches the source to extract the food and carry it to the hive. On the way to the food source, the onlooker bee may discover a better food source than the chosen one. In this case, when the onlooker bee goes back to the hive communicates the position of the new food source to the employed bee. When a food source is finished by onlooker bees, the employed bee, that was communicating that source's position, forgets those coordinates and looks for a new source.

Taking inspiration from the nature, Karaboga models three bee behaviors in the colony: (i) The Employed Bee, (ii) The Onlooker Bee, and (iii) The Scout Bee. The employed bees are associated with the specific food sources, onlooker bees watch the dance of employed bees within the hive to choose a food source, and scout bees look for food sources randomly [5].

In nature, the employed bee whose food source has been exhausted becomes a scout bee to look for the further food sources. In ABC, the solutions represent the food sources and the nectar quantity of the food sources corresponds to the fitness of the associated solution. Employed bees whose solutions was not improved after a fixed number of trials, defined *limit*, become scouts and their solutions are abandoned.

In other words, the general formulation of the ABC algorithm can be described by the following phases: (i) Bee Initialization, (ii) Employed Bee Phase (iii) Onlooker Bee Phase (iv) Scout Bee Phase (v) Memorization of the best solution found. These last four phases are iterated until the stop criteria is met. Commonly the algorithm stops when a fixed maximum number of cycles is reached.

Nowadays, different real world applications of ABC algorithm [7] have been investigated. In 2011, Karaboga and Ozturk [3] firstly introduced ABC for clustering tasks, showing how ABC formulation outperformed Particle Swarm Optimization (PSO) algorithm. Moreover the authors experimented ABC in classification tasks, comparing it with traditional classification algorithms such as Neural Networks (Multi Layer Perceptron), Bayesian Network, Radial Basis Function (RBF) proving the benefits of bee colony. A first hybrid approach is proposed by Yan et al. [4] who present a Hybrid Artificial Bee Colony algorithm. The authors consider a social learning between bees by means of cross-over operators of Genetic Algorithm and apply the proposed algorithm to some classification tasks proving some benefits despite of traditional k-means, ABC and PSO algorithm.

4 Inspector Bee in the Colony

Our proposed algorithm is inspired by the Simple ABC given by Karaboga [6], but it extends the colony modeling a forth bee behavior, i.e., Inspector Bee.

In a real bee colony, inspection role was modeled by Biesmeijer and de Vries [8], who introduced additional behavioral states for forager bees. The authors defined 7 different bee behaviors and the transitions between them: (i) novice forager, (ii) scout, (iii) recruit, (iv) employed forager, (v) unemployed experienced forager, (vi) inspector, and (vii) reactivated forager. In their work they define the inspectors as foragers that retire from an unprofitable food source but continue to make occasional trips to it, while reactivated foragers are bees that stop inspecting after a certain period of time and return to wait for dances to follow at the nest.

Granovskiy et al. [9] studied the role of inspector bees. Their experiments show that a bee colony is able to successfully reallocate its foraging resources in dynamic environments even when dance language information is limited. According to the authors, it remains unclear in what foraging situations reactivation and inspection are important and in what cases the dance language is the primary mechanism for communicating memory. The ability of the colony to react to rapid changes in their environment can be justified by the inspector bees that act as the colony's short-term memory [8]. So that, these bees allow the colony to quickly begin utilizing previously abandoned food sources once they become profitable again.

Inspection can be considered an important mechanism for reallocating foragers when food sources are hard to find: for these reasons we introduce inspector in the proposed Artificial Bee Colony. In our model, the Inspector Bee memorizes the best solution across the different cycles, so that if a solution is abandoned by bees and is not considered as the best solution for the next cycle, the inspector preserves this information.

5 Algorithm Structure and Fitness Function

Pseudo-code of our Artificial Bee Colony with Inspector behavior (ABCi) is outlined by Algorithm 1. The parameters of the proposed ABC algorithm as well as Karaboga's formulation are: the number of food sources (i.e., SN), the

Algorithm 1. ABCi: algorithm's pseudo-code

```
Load training samples
Set the number of employed bees and onlooker bees
Generate the initial population z_s, s = 1..SN with trial counter t_s = 0
Evaluate the nectar amount (fitness function) of the food sources (∀s)
Inspector bee moves to the best food source
Set cycle to 1
repeat
  for all employed bee assigned to solution s do
    Produce new solution v_s with t_s = 0
    Evaluate the fitness of the new solution v_s
    Apply greedy selection process for the identification of new population z_s
  end for
  Calculate the probability values p_s for the solutions z_s, s = 1..SN
  for all onlooker bee do
    Select a solution z_s depending on p_s
    Produce new solution v_s with t_s = 0
    Evaluate the fitness of the new solution v_s
    Apply greedy selection process for the identification of new population z_s
    if greedy selection process preserves old solution then
      Increment the trial counter t_s associated to the solution z_s
    end if
  end for
  Inspector bee moves to the best food source and memorize it
  if there is a solution with t > limit (scout bee) then
    Generate a new solution according a randomized process
    Memorize the new solution, replacing the abandoned one
  end if
  cycle = cycle + 1
until cycle = MCN
```

number of employed and onlooker bees, the value of the *limit*, and the maximum cycle number (MCN).

In clustering problem the food sources are the cluster centroids, while the solution is the position of food source which maximizes the nectar amount (the position of centroids which minimizes the fitness function).

In the initialization phase, the algorithm generates randomly a group of food sources corresponding to the solutions in the search space. According to Eq.3, the fitness of food sources is evaluated and for each food source a counter which stores the number of trials of each bee is set to 0 in this phase.

$$fitness(s) = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{i,j} \left\| x_i - c_j \right\|^2 \tag{3}$$

where K is the number of clusters, N is the number of objects, x_i is a generic input to be clustered, c_j is the jth centroid, and s is the solution (the position of K centroids).

In the employed bees' phase (see lines 8-13 in algorithm's pseudo-code), each employed bee is sent to the food source and finds a neighboring food source. The neighboring food source is produced according to Eq.4 as follows:

$$v_{i,j} = z_{i,j} + \phi\left(z_{i,j} - z_{k,j}\right) \tag{4}$$

where k is a randomly selected food source different from i, j is a randomly chosen centroid. ϕ is a random number between [-1,1]. The new food source v is determined by changing randomly one dimension on jth centroid. If the produced value exceeds its predetermined boundary, it will set to be equal to the boundary. Then the new food source is evaluated. Therefore, a greedy selection is applied. In other words, the employed bee produces a modification in the position (i.e. solution) and checks the nectar amount (fitness value) of that source (solution). The employed bee evaluates this nectar information (fitness value) and then assigns to the food source a probability related to its fitness value according to the Eq.5.

$$p(s) = f(s) \bigg/ \sum_{j=1}^{K} f_j \tag{5}$$

where K is the number of food sources and $f(s) = \frac{1}{1+fitness(s)}$

In the onlooker bees' phase (see lines 14-23 in algorithm's pseudo-code), the onlooker bee selects a food source based on a probability of a source explored by employed bees. Once the food sources have been selected, each onlooker bee finds a new food source similarly to the employed bee (see Eq.4) and the greedy selection process select the new source. If this process preserves old solution, the value of counter, which is associated to the employed bee, increases.

In scout bees' phase (see lines 24-27 in algorithm's pseudo-code), when the value of the counter t of a food source is greater than $limit$, the food source is abandoned, the inspector bee memorizes the source and the employed bee becomes a scout. The scout bee generates a new solution according to Eq.6 and sets the value of counter equal to 0, so that the bee memorizes the new solution replacing the abandoned one.

$$z_{j,d} = \min_{i=1}^{N}\left(x_{i,d}\right) + rand(0,1) \cdot \left(\max_{i=1}^{N}\left(x_{i,d}\right) - \min_{i=1}^{N}\left(x_{i,d}\right)\right) \tag{6}$$

where $j = 1, 2, ...K$ and $d = 1, 2, ..., D$. N is the number of objects, K is the number of clusters, and D is the number of features. $x_{i,d}$ represents the d-th feature of the input data x_i.

6 Experimental Results

In this section we experiment the proposed ABC algorithm for some clustering problems and for an application in Transportation System.

In order to evaluate the performance of the proposed ABC approach, we compare the results of the K-means, ABC, and the proposed ABC for a clustering task by comparing four different datasets. These datasets are selected from the UCI machine learning repository (Breast Cancer Wisconsis, Credit Approval, Dermatology and Iris datasets) [10]. An additional dataset, which have been extracted from a real-world clustering problem in Poste Italiane domain, is considered as an example of application.

Iris data was collected by Anderson in 1935 and consists of 150 random samples of flowers from the iris species setosa, versicolor, and virginica. From each species there are 50 observations for sepal length, sepal width, petal length, and petal width in cm.

Wisconsin Breast Cancer consists of 683 objects characterized by 9 features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects).

Credit Approval dataset contains 690 samples, which are different credit card applications, with 15 attributes. This dataset has a good mix of attributes (continuous, nominal with small numbers of values, and nominal with larger numbers of values) and data can be grouped either in approved or not approved.

Dermatology consists of 366 samples characterized by 34 features which are 12 patient clinical attributes and 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope. The diseases in this group can be one of the following six: psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris.

6.1 Convergence Analysis

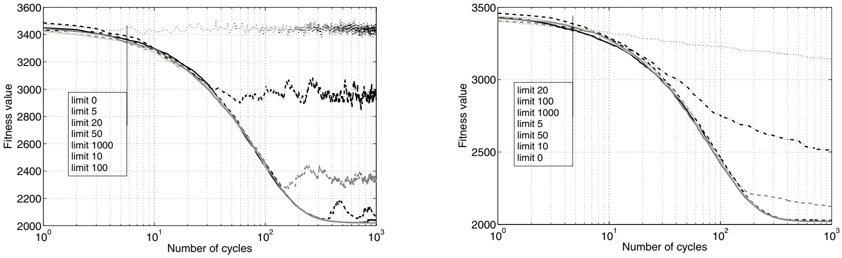
We run the algorithm several times with different value of *limit* in order to study quantitatively the convergence of the two different ABC formulations. We consider the four datasets from UCI database as benchmark data.

The parameters in an ABC approach are the *limit* and the colony size [11], so that we study algorithm's performance as long as the parameters change.

We repeated 20 runs for different problem configurations. First of all, cycle after cycle, we report the average of best fitness with different abandonment behavior of a nectar source (*limit* is equals to 0, 5, 10, 20, 50, 100, 1000) when a colony of 20 bees is considered.

To find a better solution, one may search the largely unknown region (*exploration*) or search around the current solution (*exploitation*). The tradeoff between exploration and exploitation is represented by the *limit*. Indeed, higher values of limit emphasize exploitation behavior of the algorithm, while lower values of limit foster exploration phase.

Best solutions occurs when *limit* increases, as the exploitation behavior becomes more relevant. On the other hand, very high value (i.e., 1000) of limit holds algorithm back for exploration of new solutions. However, we can notice



(a) Inspector is not considered (b) Inspector is considered

Fig. 1. Dermatology dataset: Average fitness behavior by varying the *limit*. Colony size equals 110.

Table 1. Wilcoxon paired test on Iris dataset: Average fitness and p-values

Limit	Average Best Fitness			p-value		
	ABC	ABCi	k-Means	ABC vs. ABCi	ABC vs. k-Means	ABCi vs. k-Means
0	214.278	140.450		7.254e-12	7.562e-09	7.562e-09
5	139.540	101.306		7.254e-12	9.637e-08	3.016e-06
10	111.508	97.461	99.990	1.451e-11	1.006e-06	1.709e-01
20	97.956	96.698		3.685e-09	1.709e-01	1
50	96.675	96.656		1.066e-01	1	1
100	96.659	96.655		1.024e-02	1	1
1000	96.655	96.655		8.318e-01	1	1

how ABCi's convergence is not heavily affected by limit value if they ranges between 20 and 100, thus resulting algorithm to be robust to this situation.

Furthermore, *limit* equals to 50 (black curve) could be a good tradeoff, even if the optimal parameter value depends on the particular problem. Indeed, the Dermatology dataset (see Fig.3) seems not to converge with *limit* equal to 0, 5 and 1000. Moreover this dataset presents high convergence time with other values. The problem is the colony size which must be incremented as proved in Fig.1 when we repeat the 20 different runs with 100 onlookers and 10 employers.

Investigating these results more deeply, we consider Mann-Whitney-Wilcoxon test, reporting results in Table 1, where the average value of best fitness of 20 different trials per technique (i.e., ABC, ABC with inspector, k-means). The null hypothesis is: the investigated techniques provide solutions which belong to the same population entailing a comparable clustering performance, and the alternative hypothesis is: (i) ABC fitness is greater than ABCi one, (ii) ABC fitness is greater than k-means, and (iii) ABCi fitness is greater than k-means.

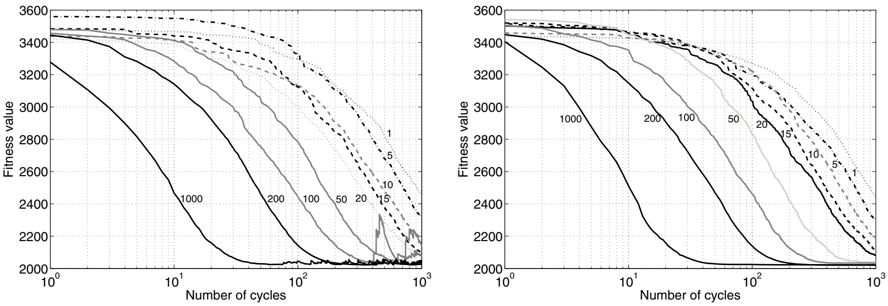
Assuming 0.05 as upper bound to reject the null hypothesis, we can affirm that there is statistical difference between ABC and ABCi. We prove that ABCi outperforms ABC because ABCi provides a lower fitness value in most of the cases. We cannot reject the null hypothesis with higher value of limit (i.e., *limit*

Table 2. Wilcoxon paired test on Breast Cancer dataset: Average fitness and p-values

Limit	Average Best Fitness			p-value		
	ABC	ABCi	k-Means	ABC vs. ABCi	ABC vs. k-Means	ABCi vs. k-Means
0	7254.312	4645.158		7.254e-12	4.003e-09	4.003e-09
5	4860.267	3280.301		3.265e-10	4.003e-09	4.003e-09
10	3462.931	3044.500	3061.098	2.176e-10	1.996e-06	1
20	3165.311	3035.615		1.183e-04	1	1
50	3037.424	3035.571		5.658e-01	1	1
100	3035.571	3035.571		8.367e-01	1	1
1000	3035.571	3035.571		3.088e-01	1	1

Table 3. Wilcoxon paired test on Credit Approval dataset: Average fitness and p-values

Limit	Average Best Fitness			p-value		
	ABC	ABCi	k-Means	ABC vs. ABCi	ABC vs. k-Means	ABCi vs. k-Means
0	4.399e+06	5.958+05		7.254e-12	4.003e-09	1
5	1.617e+06	5.731e+05		8.705e-11	1.267e-01	1
10	6.183e+05	5.624e+05	8.087e+05	7.254e-12	1	1
20	5.614e+05	5.571e+05		6.673e-06	1	1
50	5.570e+05	5.568e+05		9.328e-01	1	1
100	5.570e+05	5.568e+05		7.858e-02	1	1
1000	5.568e+05	5.568e+05		9.214e-01	1	1



(a) Inspector is not considered (b) Inspector is considered

Fig. 2. Dermatology dataset: Average fitness behavior by varying the number of onlookers (10 employers)

greater than 100) and ABC and ABCi performance are comparable. Indeed, considering a higher value of limit, the abandonment behavior of an employed bee decreases and the benefit of an inspector bee is not estimable.

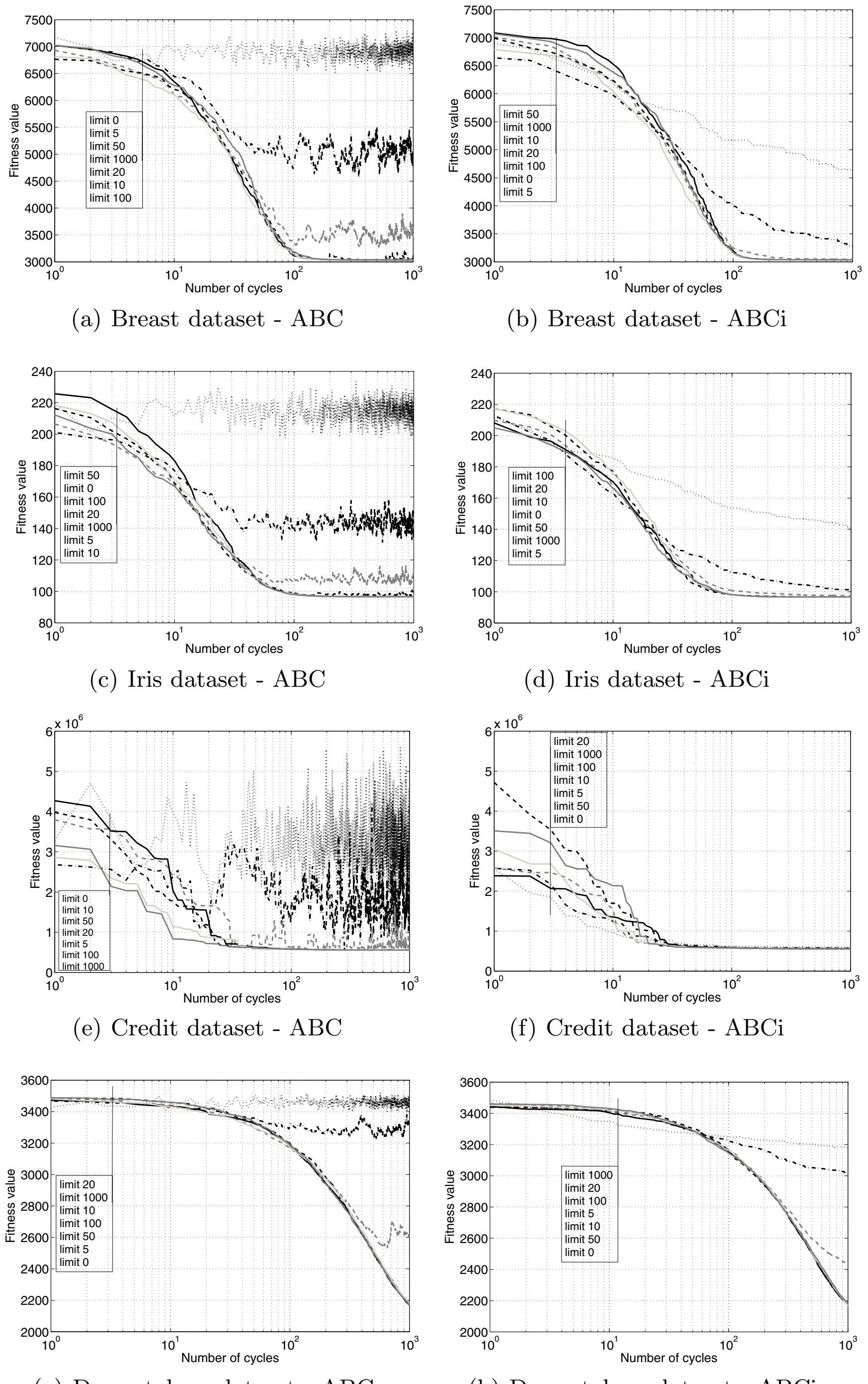


Fig. 3. Average fitness behavior by varying the *limit*. Colony size equals to 20.

Instead, comparing k-means with ABC approach, we prove how a bee colony can outperform with *limit* greater than 20. Low values of limit penalizes exploitation behavior, while ABCi with *limit* equal to 20 or 50 is able to provide promising results in clustering problem, improving k-means results.

The same findings arise from Breast Cancer dataset (e.g., see Tab.2), confirming the importance of *limit* and the benefits of inspector bee within the colony.

Taking into account Credit Approval dataset (see Tab.3), ABCi formulation is confirmed to improve ABC's results when the *limit* is lesser than 50, no statistical evidence when limit value increases. Comparing ABCi with k-means we prove better results of honey bee approaches. Moreover, considering Dermatology dataset as depicted in Fig.3, we can observe how *limit* ranging between 20 and 50 represents the best choice even if ABC approach does not outperform k-means due to the colony size which need to be increased as proved in Fig.1.

Finally, in order to study the effect of the number of onlookers for algorithm's convergence speed, we show in Fig.2 the average fitness behavior of 20 different runs. As we expected, the more number of onlookers increases, the more quickly the algorithm converges.

As an example of application in Transportation domain, we consider the problem of vehicle clustering. The purpose of the analysis is to group together Poste Italiane vehicle with the same features, i.e., the average monthly fuel consumption index and the average monthly vehicle route. In particular, fuel consumption index measures the vehicle's cost and identify at the same time the vehicle's performance. Indeed, it considers the fuel demand related to the followed route.

Starting with a set of 10984 Poste Italiane cars which supply postal items to the national addressees, we adopt ABC clustering in order to group together in a same cluster those cars with the same delivery behavior and the same fuel consumption index. The ABC algorithm is setup with following parameters: MCN = 1000, colony size = 61 (50 onlooker bees, 10 employed bees and 1 inspector), $limit = 50$.

Analyzing the results, we can state that the cars are properly grouped in clusters which are suitable for knowledge extraction process and are useful to understand the reason of the provided cars' performances.

7 Conclusions and Future Work

In this paper we presented a bee colony algorithm for clustering problem. Starting from the experiment conducted by Granovskiy regarding the role of inspector bee within a colony, we modeled and proposed a bee colony with inspector. Our experimentation showed the impact in adopting this bee within the colony, and the benefit is proved.

Comparing bee colony with other evolutionary techniques as genetic algorithms, the role of inspector in the convergence can be compared to the role of elitism in genetic approach.

We adopted bee colony for different clustering tasks from biomedical to industrial domain and experimentation provided very encouraging results, proving the ability of a ABC algorithm in converging towards solutions with high fitness, also in presence of different features (e.g., Dermatology dataset) and different input. Moreover, the algorithm has been proven to provide better results increasing the colony size and exploration and exploitation behavior is investigated as long as the *limit* changes.

However, we aim to investigate two main directions in the future. The first is how to improve performance in algorithm's computational time. Parallel ABC colony seems to be a promising solution. The second direction is to investigate other real-world optimization problems with ABC as the vehicle routing problem. In this case, ABC poses additional interesting questions and can be a valid solution in Intelligent Transport System domain.

References

1. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31(8), 651–666 (2010)
2. Hruschka, E., Campello, R.J.G.B., Freitas, A., De Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(2), 133–155 (2009)
3. Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial bee colony (ABC) algorithm. Applied Soft Computing 11(1), 652–657 (2011)
4. Yan, X., Zhu, Y., Zou, W., Wang, L.: A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomput. 97, 241–250 (2012)
5. Karaboga, D.: An idea based on Honey Bee Swarm for Numerical Optimization. Technical Report TR06, Erciyes University (October 2005)
6. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8(1), 687–697 (2008)
7. Abu-Mouti, F., El-Hawary, M.: Overview of artificial bee colony (ABC) algorithm and its applications. In: 2012 IEEE International Systems Conference (SysCon), pp. 1–6 (2012)
8. Biesmeijer, J.C., de Vries, H.: Exploration and exploitation of food sources by social insect colonies: a revision of the scout-recruit concept. Behavioral Ecology and Sociobiology 49(2-3), 89–99 (2001)
9. Granovskiy, B., Latty, T., Duncan, M., Sumpter, D.J.T., Beekman, M.: How dancing honey bees keep track of changes: the role of inspector bees. Behavioral Ecology 23(3), 588–596 (2012)
10. Bache, K., Lichman, M.: UCI machine learning repository (2013)
11. Akay, B., Karaboga, D.: Parameter tuning for the artificial bee colony algorithm. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 608–619. Springer, Heidelberg (2009)