Int. J. Advance Soft Compu. Appl, Vol. 8, No. 1, March 2016
ISSN 2074-8523

Novel Multi-Objective Artificial Bee Colony Optimization for Wrapper Based Feature Selection in Intrusion Detection

Waheed Ali H. M. Ghanem[1], and Aman Jantan[2]

[1]School of Computer Science, Universiti Sains Malaysia
Malaysia (USM), Gelugor, 11700 Pulau Pinang, Malaysia
e-mail: wahm13_com031@student.usm.my

[2]School of Computer Science, Universiti Sains Malaysia
Malaysia (USM), Gelugor, 11700 Pulau Pinang, Malaysia
e-mail: aman@cs.usm.my

Abstract

This study proposes a novel approach based on multi-objective artificial bee colony (ABC) for feature selection, particularly for intrusion-detection systems. The approach is divided into two stages: generating the feature subsets of the Pareto front of non-dominated solutions in the first stage and using the hybrid ABC and particle swarm optimization (PSO) with a feed-forward neural network (FFNN) as a classifier to evaluate feature subsets in the second stage. Thus, the proposed approach consists of two stages: (1) using a new feature selection technique called multi-objective ABC feature selection to reduce the number of features of network traffic data and (2) using a new classification technique called hybrid ABC–PSO optimized FFNN to classify the output data from the previous stage, determine an intruder packet, and detect known and unknown intruders.

Keywords: *Multi-Objective Optimization; Swarm Intelligent; Feature Selection; Wrapper Approach; Intrusion Detection System.*

1 Introduction

The evolution of network technology and the Internet proceeds at an accelerated rate. On the one hand, such development provides great convenience in information exchange among people worldwide; on the other hand, it is the reason for the increasing diverse threats [1]. The most infamous threats are hacker attacks, viruses, malware, and other unscrupulous activities; distinguishing between these threats and normal network activities is difficult. In 1980, Anderson proposed the concept of intrusion detection (ID) [2]. ID is a security measure that is based on the premise that malicious behavior deviates from normal behavior. The most important goal of ID is to identify malicious behavior that threatens the integrity, confidentiality, and availability of information resources [3].

An ID system (IDS) is the second wall of defense after a firewall. It is an essential and necessary key to secure computer networks. IDSs operate by recognizing and notifying against abnormal insider network traffic or attacks to network channels. IDSs work by investigating network traffic for possible attacks and raising the alarm when a suspicious activity is detected [3, 4]. The detection methods employed by IDSs can be classified into two: anomaly detection and misuse detection. An anomaly detection method is based on first identifying the normal behavior of all network components and then classifying a behavior as abnormal in case it deviates from normal. Consequently, this method is characterized by its capability to detect new types of attacks [5]. A misuse detection method depends on a database of intrusion patterns that represent well-known vulnerabilities (signatures); the current behavior in a network is matched against the signatures in the database to determine whether it is normal. Therefore, this method is unable to detect new types of attacks. Numerous factors need to be considered when constructing an IDS, including data collection, data processing, and classification accuracy [6]. Classification is the process of forecasting using the class labels of instances that are typically described by a set of features in a data set. Another essential factor in IDS construction is the method of addressing network traffic before sending it to the classifier, which contains a considerable number of features [7]. High-dimensional network traffic causes a decline in the classification accuracy of an IDS. Therefore, an IDS should implement a primary treatment phase for high-dimensional data before solving the classification problem. Reducing the number of features and choosing the best among them influence the accuracy detection rate and the false alarm rate [8].

2 Multi-Objective Optimisation

Multi-objective optimization is considerably useful when an optimal decision needs to be reached, particularly when trade-offs between more than two conflicting objective functions exist. Such optimization involves two conflicting objectives and the maximization or minimization of the multiple conflicting objective functions. Mathematically, multi-objective optimization involves minimizing or maximizing a problem with multiple objective functions. The formula for a minimization problem with n objectives can be written as follows:

$$minimise\ F(x) = [f1\ (x), f2\ (x), f3\ (x), \ldots\ , fn\ (x)] \quad (1)$$

Subject to:

$$g_i(x) \leq 0, i = 1,2,3, \ldots\ m \quad (2)$$

$$h_i(x) = 0, i = 1,2,3, \ldots\ l \quad (3)$$

Where x represents an n-dimensional decision vector and n indicates the number of objective functions to be minimized. Thus, when n is equal to 1, the model in Equation (1) is a single objective problem, and the optimal solution is the one that minimizes the objective. Nevertheless, when n > 1 (i.e., a multi objective problem), $f_i(x)$ is an objective function; and $g_i(x)$ and $h_i(x)$ are the constraint functions of the maximization and minimization problems, respectively.

The quality of a solution generated by multi-objective optimization is measured by the trade-off between n conflicting objectives. Let and represent the two solutions for the previous n-objective minimization problem. If the following conditions have been satisfied, then dominates (or is non-dominated or is better than). All non-dominated solutions are the optimal solutions to the multi-objective problem because these solutions are not dominated by any other solution. The set of these solutions is named Pareto set or Pareto front [9].

$$\forall i : f_i(x) \leq f_i(y) \ \ and \ \ \exists j : f_i(x) < f_i(y) \quad (4)$$

When is not dominated in a multi-objective problem by any other solution, is represented as a Pareto optimal solution. A Pareto optimal solution represents the set of trade-off surfaces in the search space and is called Pareto front. Hence, a multi-objective optimization algorithm is employed to generate a set of non-dominated solutions in the search space.

To apply the concept of feature selection to improve IDS performance, a trade-off among three conflicting objective functions is required to minimize the number of features, the classification error rate, and the false alarm rate simultaneously. The use of multi-objective artificial bee colony (ABC) optimization for feature selection in an IDS has not been investigated until now.

3 Artificial Bee Colony Algorithm (ABC)

To optimize numerical problems, Karaboga proposed the ABC algorithm in 2005 [10]. Several developments were made by the author of this method alongside some researchers who exhibited interest in the approach [11].

The inspiration and starting point of ABC are the behavior of bee swarms, which has been found to be intelligent. Using the population of bees as basis permitted the optimization algorithm to have a simple and stochastic, but nonetheless robust form. In [11], the performance of this algorithm was compared with those of other well-known and previously reported algorithms. The authors compared ABC to particle swarm optimization (PSO), genetic algorithms (GAs), and differential evolution. Bees were categorized into three types: employed, onlookers, and scouts.

Many tasks are performed by a bee in a colony, and the most important of which is finding the locations of food sources. Each location is then evaluated based on food quality [10]. The ABC algorithm depends on two criteria: the solution and the quality of the solution. A possible solution represents the location of a food source. The quality of a solution is comparable with the amount of nectar in the source location. Fitness is determined by the quality and amount of nectar in the food source.

We can classify bee swarms into working bees (employed bees) and non-working bees (unemployed bees). Non-employed bees can be divided into onlooker and scout bees. Employed bees find food sources and then deliver the information related to the location of these food sources to the beehive. The role of non-employed bees is to stay in the hive and evaluate the information on the food sources found by the employed bees to determine which among these is more valuable. An interesting fact is that the number of employed and onlooker bees is equal to the number of available food sources. Scout bees are always looking for new food sources [12].

The ABC algorithm consists of iterative steps. Two particular steps are vital to the evolution of the ABC population. The first step is to find new food location within a certain area, and the second step is to select the best food locations by assessing them based on previous experiences. To accomplish these two steps, the bees have to follow a four-step workflow: the first step is instauration, the second relies on employed bees, the third relies on onlooker bees, and the fourth relies on scout bees. A detailed discussion of the four steps follows [13].

3.1 Initialization Phase and the Optimization of Problem Parameters

At first, the ABC Algorithm gets us thinking about a proportionally distributed population that has SN solutions (Solution Number). In the SN solutions every solution (i=1, 2…, SN) represents a D-dimensional vector, where D is the number of variations in the optimization of the coordinates of the problem and refers to the i^th food location for the bee population. The food location can be determined by applying the following equation:

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \qquad (5)$$

In Equation (2), x_{max}^j and x_{min}^j are the bounds of x_i in the *j*th dimension. Moreover, the ABC algorithm relies on three coordinates of control. First, the number of food sources depends on the population of bees. Second, the maximum cycle number can determine the maximum number of generations. Finally, a limit is used to determine the number of accepted generations after which the food sources that have not been improved will be removed from the locations used by that particular bee population. Once a food source is found and given to employed bees, the correct option that is specific for optimizing the coordinates of the problem is executed, and the bees reach the final result, in which Equation (3) is used to obtain every fitness value offered by each food location.

$$fit_i(t) = \begin{cases} \frac{1}{1+f_i(t)} & if\ (f_i(t) \geq 0) \\ 1 + abc\left(f_i(t)\right) & otherwise \end{cases} \tag{6}$$

In Equation (3), fit$_i$(t) represents the fitness value of the i^{th} food location. The result is obtained by using food locations. An option is correct when it specifically optimizes the coordinates of the problem.

3.2 Employed Bee Phase

In this phase, the employed bees use the information obtained from personal experiences and the quantity of nectar of the solution to implement changes on the actual solution. In case the quantity of nectar in the newly found location is greater than that in the previous food locations, the bee population starts procuring food from the new location and abandons the last one. The following equation determines how a food source is modified:

$$\mathcal{V}_{ij} = \mathcal{X}_{ij} + \emptyset_{ij}\left(\mathcal{X}_{ij} - \mathcal{X}_{\mathcal{K}j}\right) \tag{7}$$

Where $\emptyset_{ij}(\mathcal{X}_{ij} - \mathcal{X}_{\mathcal{K}j})$ represents the length of the step, and $\mathcal{K}$ ∈ {1, 2 ... SN} and j ∈ {1, 2 ... D} are two indices that are selected at random. $\mathcal{K}$ Is not equal to i because the length of the step has an important contribution, and $\emptyset_{ij}$ is only a number within the range of [0, 1].

3.3 Onlooker Bee Phase

The onlooker bee phase can be initiated only after the employed bee phase. In this phase, the quantity of nectar that has been collected during the employed bee phase is communicated. In communicating this information, the bees have to include the updated solution of their food location and the coordinates of this location. Onlooker bees also examine the information they hold and choose an option with probability$\mathcal{P}_i$.

$$\mathcal{P}_i = \frac{\mathrm{fit}_i}{\sum_{i=1}^{SN} \mathrm{fit}_i} \tag{8}$$

In Equation (8), fit_i is the quantity of nectar that is determined by the ith solution. Similar to employed bees, an onlooker bee updates the location of the new food source in its memory and compares the nectar of this new source to the nectar of another source. In case the quantity of nectar is greater than the one provided by the last food source, the bees remember the new food source and forget about the unproductive source.

3.4 Scout Bee Phase

In this phase, an employed bee transforms into a scout bee in case the employed bee makes a connection with the unproductive food source. Moreover, the food location is replaced with another food location that can be found within the search area. The scout bee phase can be initiated once the place where the food source is located has not been updated for numerous cycles, which causes the bees to think that that particular food source has been left behind. In the ABC algorithm, the number of cycles is an important control indicator and is called the limit for abandonment. In case the abandoned food source is x_i, scout bees replace the food source with another x_i in the following manner:

$$x_i^j = x_{min}^j + rand(0,1)\left(x_{max}^j - x_{min}^j\right) \forall j = 1, 2, \ldots\ldots, D \tag{8}$$

In Equation (6), x_{max}^j and x_{min}^j are the bounds of x_i in the *j*th dimension.

4 Feature Selection and Classification

Feature selection is one of the most proactive steps before the data classification process. It aims to address the high-dimensional space of the data set and eliminate redundant or irrelevant features. Thus, feature selection is based on removing noise features by selecting only useful and relevant features. Feature selection algorithms are based on two main procedures, namely, generation procedure and evaluation function. The first step generates the subset of features, and the second step evaluates the candidate subset of features. Sub-feature evaluation can be conducted using one of the following approaches: the filter approach or the wrapper approach. If the evaluation function depends on the machine learning algorithm, such as a classifier to evaluate the subset feature, then this approach is called “Wrapper.” By contrast, the approach is called “Filter” if it is dependent on the applied distance and the information measures in the evaluation because it examines the intrinsic properties of the data. Furthermore, compared with the filter approach, the wrapper approach needs more computational power but is more accurate [14].

Classification is one of the most important algorithms in the field of machine learning and data mining. Its main task is to predict the class label of each instance based on the information described in the features. However, the most common problem in the classification process is the inclusion of a large number of features, which are likely to include irrelevant and redundant features that may reduce classification performance because of the unnecessarily large search space. Identifying the relevant features for classification simplifies the learned classifier, reduces operating time, and improves classification performance [15].

5 Multi-Objective Approach

This section introduces a new algorithm for feature selection using multi-objective ABC. This algorithm addresses three objectives: to minimize the number of features, the false alarm rate, and the classification error rate. It is based on a new fitness function to explore the Pareto front of feature subsets. Numerous studies have applied the ABC (standard, binary) algorithm to solve the feature selection problem. Nevertheless, the majority of the current approaches suggest maximizing classification performance. Studies that solve the feature selection task as a multi-objective problem are minimal. Indeed, this work is considered the first investigation on a new approach to feature selection using multi-objective ABC.

This study does not only provide a new approach for feature selection but also proposes a new fitness function for feature selection to reduce the number of features and achieve the minimum rate of classification errors and false alarms. Fig. 1, represents the block diagram of wrapper-based feature selection using multi-objective ABC. The classifier used in this work, as shown in Fig. 1, is the one we proposed in our previous work [16].

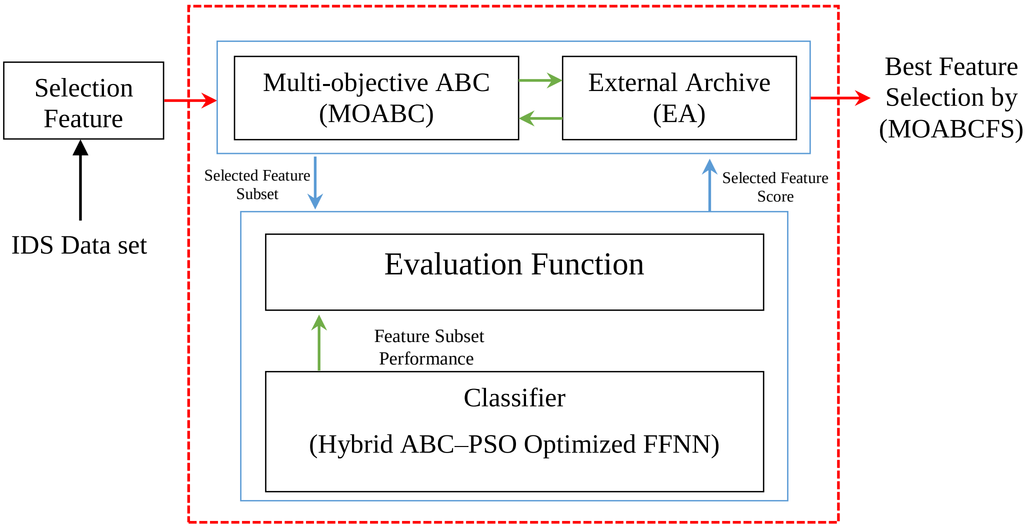

Fig. 1: Diagram for Multi-Objective ABC Based Wrapper Feature Selection.

As a single-objective problem, the ABC algorithm can be used in wrapper-based feature selection to minimize the classification error rate only. The single-objective ABC algorithm is used in this study to assess if it can identify a subset of features to achieve the highest classification accuracy instead of using all features, and the results of such assessment can be used as a benchmark against which the performance of the newly developed approach can be compared. Equation (10) shows the fitness function that is used to minimize the classification error rate in the selection of a feature subset, which has been selected through an evolutionary training process.

$$Fitness = (fp + fn)/(tp + tn + fP + fN) \tag{9}$$

However, Equation (10) may still produce a feature subset that exhibits redundancy because the equation does not intend to reduce the number of features. In addition, it is not concerned with reducing the false error rate, which is important in improving the classification accuracy of IDSs. Hence, we suppose that reducing the number of features while achieving a favorable trade-off between the classification error and false alarm rates is possible. In line with this, we obtain the best trade-off among the three objectives (features, classification error rate, and false alarm rate). To address this problem, a new fitness function is suggested with the goal of minimizing the number of features, the false alarm rate, and the classification error rate. Equation (11) shows the formula of the new function.

$$Fitness = \alpha * \left(\frac{Selected\ Feature}{All\ Feature}\right) + \beta * \left(\frac{False\ Rate}{All\ False}\right) + \beta * \left(\frac{Error\ Rate}{All\ Error}\right) \tag{10}$$

Where $\alpha \in [0,1]$. The relative importance of the number of selected features, and the false alarm and classification error rates are represented by α and β, respectively; $\beta = \frac{(1-\alpha)}{2}$ is set to be larger than α because the false alarm and classification error rates are assumed to be more important than the number of selected features (SF). However, SF is frequently considerably larger than the false alarm rate or "$False\ Rate$" (FR) and the classification error rate or "$Error\ Rate$" (ER). To achieve a balance among these three components, SF is divided by the total number of features to assign a value within the range of (0, 1). Furthermore, FR and ER are normalized by dividing them by the false alarm and classification error rates of all available features ($All\ False, All$ Error).

In general, the ABC algorithm was originally proposed as a single-objective technique. However, it is used in many studies to solve multi-objective problems, and the multi-objective approach has demonstrated relatively higher effectiveness compared with the other algorithms, such as PSO and GAs [9, 17]. The multi-objective ABC algorithm is used to address the wrapper-based feature selection problem, and consequently, to make an informed decision from many available feature subsets in the Pareto front by using Equation (11) as the fitness function. The Pareto front concept is used in the proposed approach to select the non-dominated solution as the leader solution and maintain it in an external archive (EA). A good leader in Pareto front is determined by the best solution of the multi-objective ABC feature selection (MOABCFS) algorithm.

In this work, all the solutions are stored in the EA, which represents the food source position and all the bees. The EA uses crowding distance to estimate the density of the solutions as well as to sort and rank these solutions according to each objective function value (fitness function).

The **MOABCFS** pseudo-code proposed in this study is as follows:

```
Input: A Training set and a Test set;
Output: A set of non-dominated solutions, training and test accuracies.
Begin
Initialize the food source positions (solutions) $X_i$, I = 1 … SN;
Evaluate the nectar amount by three objective values (Use new fitness Function) of
food sources; /*number of features and the classification error rate and false
alarm rate on the Training set */
Identify the non-dominated solutions Bees (NonDomBee) in colony;
Calculate crowding distance of each Bee in (NonDomBee);
Store bees in (NonDomBee) solutions sorted based on the crowding distance in the
external archive EA
While Cycle ≤ Max_Cycle do
Employed Bees' Phase
For each Employed bee
   Randomly chooses a solution from the highest ranked (Least crowded) solutions in
   NonDomBee;
   Produce new solution $v_i$ by using expression (Eq.7);
   Calculate the value fitness $fit_i$, and evaluate it using by three objective
   values;
   Apply greedy selection mechanism to decide which solution enters EA
End
Onlooker Bees' Phase
Calculate probabilities for each food source
For each Onlooker bee
   Randomly chooses a solution from the highest ranked (Least crowded) solutions in
   NonDomBee;
   Produce new solution $v_i$ by using expression (Eq.8);
   Calculate the value fitness $fit_i$, and evaluate it using three objective values;
   Apply greedy selection mechanism between old solution $x_i$, and new solution $v_i$ to
decide which solution enters EA
End
Scout Bees' Phase
For each Scout bee
   Determine the abandoned solution for the scout
   If exists, and replace it with a new randomly produced solution for the scout
   using ($X_{ij}$)
End
Update the Archive
Identify different levels of Pareto fronts PF = (PF1, PF2, …$PF_i$) in external
archive;
Empty the current food sources for the next iteration;
```

```
 i = 1;
While | food sources | < Population Size do
  If (|food sources | + |Fi | ≤ Population Size) then
    Add Fi to food sources;
    i = i + 1;
  End
  If (|food sources | + |Fi| > Population Size) then
    Calculate crowding distance in Fi and sort bees in Fi;
    Add the (Population Size - | food sources |) least crowded bees to food
    source;
  End
End
Calculate the classification error rate of the solutions (feature subsets) in the
F1 on the test set; /* F1 is the achieved Pareto front */
Return the positions of Bees in F1, the training and test classification error
rates of the solutions in F1;
End
```

6 Conclusion

In this article, we reported the first-ever study on a new feature selection approach that employed multi-objective ABC to select a set of Pareto fronts of non-dominated solutions, which represented the feature subset. In addition, a new fitness function that aimed to minimize the number of features, the classification error rate, and the false alarm rate in IDSs was investigated. The proposed approach allowed the optimization of the number of selected features, the false alarm rate, and classification performance.

ACKNOWLEDGEMENTS
This research experiment has been funded by Universiti Sains Malaysia under the Research University Cluster Grant Scheme (RUC) for “Reconstruction of the Neural Microcircuitry or Reward-Controlled Learning in the Rat Hippocampus” [1001/PKOMP/8630022]. The research also partially supported by the Fundamental Research Grant Scheme (FRGS) for "Content-Based Analysis Framework for Better Email Forensic and Cyber Investigation” [203/PKOMP/67 11426], also support from USM Fellowship (APEX [1002/JHEA/ATSG4001]) from Institute of Postgraduate Studies, Universiti Sains Malaysia.

References

[1] Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L., & Perkasa, C. D. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert systems with Applications, 38(1), 306-313.

[2] Anderson, J. P. (1980). Computer security threat monitoring and surveillance (Vol. 17). Technical report, James P. Anderson Company, Fort Washington, Pennsylvania.

[3] Subbulakshmi, T., Ramamoorthi, A., & Shalinie, S. M. (2010). Feature Selection and Classification of Intrusions Using Genetic Algorithm and Neural Networks. In Recent Trends in Networks and Communications (pp. 223-234). Springer Berlin Heidelberg.

[4] Elngar, A. A., Dowlat, A., & Ghaleb, F. F. (2012). A Fast Accurate Network Intrusion Detection System. International Journal of Computer Science and Information Security, 10(9), 29.

[5] Wu, S. X., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing, 10(1), 1-35.

[6] Luo, B., & Xia, J. (2014). A novel intrusion detection system based on feature generation with visualization strategy. Expert Systems with Applications, 41(9), 4139-4147.

[7] Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. Cybernetics, IEEE Transactions on, 43(6), 1656-1671.

[8] Idowu, R. K., Maroosi, A., Muniyandi, R. C., & Othman, Z. A. (2013). An Application of Membrane Computing to Anomaly-based Intrusion Detection System. Procedia Technology, 11, 585-592.

[9] Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. Swarm and Evolutionary Computation, 2, 39-52.

[10]Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical report-tr06, Erciyes University, engineering faculty, computer engineering department.

[11]Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), 459-471.

[12]Beheshti, Zahra, and Siti Mariyam Hj Shamsuddin. "A review of population-based meta-heuristic algorithms." Int. J. Adv. Soft Comput. Appl 5.1 (2013): 1-35.

[13] BOLAJI, A. L. A., Khader, A. T., Al-Betar, M. A., & Awadallah, M. A. (2013). Artificial bee colony algorithm, its variants and applications: a survey. Journal of Theoretical & Applied Information Technology, 47(2).

[14]Zainal, A., Maarof, M. A., & Shamsuddin, S. M. (2007). Feature selection using rough-DPSO in anomaly intrusion detection. In Computational Science and Its Applications–ICCSA 2007 (pp. 512-524). Springer Berlin Heidelberg.

[15] Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. Knowledge and Data Engineering, IEEE Transactions on, 17(4), 491-502.

[16]Ghanem, W. A. H., & Jantan, A. (2014). Using hybrid artificial bee colony algorithm and particle swarm optimization for training feed-forward neural networks. Journal of Theoretical and Applied Information Technology, 67(3).

[17]Zou, W., Zhu, Y., Chen, H., & Zhang, B. (2011). Solving multiobjective optimization problems using artificial bee colony algorithm. Discrete dynamics in nature and society, 2011.