Accepted Manuscript

International Journal of Pattern Recognition and Artificial Intelligence

Article Title:	WHO: A New Evolutionary Algorithm Bio-Inspired of Wildebeests with a Case Study on Bank Customer Segmentation
Author(s):	Mohammad Mahdi Motevali, Ali Mohammadi Shanghooshabad, Reza Zohouri Aram, Hamidreza Keshavarz
DOI:	10.1142/S0218001419590171
Received:	09 January 2017
Accepted:	06 September 2018
To be cited as:	Mohammad Mahdi Motevali *et al.*, WHO: A New Evolutionary Algorithm Bio-Inspired of Wildebeests with a Case Study on Bank Customer Segmentation, *International Journal of Pattern Recognition and Artificial Intelligence*, doi: 10.1142/S0218001419590171
Link to final version:	https://doi.org/10.1142/S0218001419590171

World Scientific
www.worldscientific.com

WHO: A New Evolutionary Algorithm Bio-Inspired of Wildebeests with a Case Study on Bank Customer Segmentation

Key Words: Wildebeests Herd Optimization (WHO), Population based Algorithms, Evolutionary Clustering, Meta-Heuristic Clustering, Customer Relationship Management, and Customer Segmentation

Abstract

Numerous evolutionary algorithms have been proposed which are inspired by the amazing living of creatures, such as animals, insects, and birds. Each inspired algorithm has its own advantages and disadvantages, and has its own way to accomplish exploration and exploitation. In this paper, a new evolutionary algorithm with novel concepts, called Wildebeests Herd Optimization **(**WHO), is proposed. This algorithm is inspired from the splendid life of wildebeests in Africa. Moving and migration are inseparable from wildebeests' lives. When a wildebeest wants to choose its path during migration, it considers the best path known to itself, the location of the more mature wildebeests in the crowd, and the direction of wildebeests with high mobility. The WHO algorithm imitates these traits, and can concurrently explore and exploit the search space. For validating WHO, it is applied to optimization problems and data mining tasks. It is demonstrated that WHO outperforms other evolutionary algorithms, such as genetic algorithm and particle swarm optimization, in the assessed problems. Then, WHO is applied to the customer segmentation problem. Customer segmentation is one of the most important tasks of data mining, especially in the banking sector. In this paper, the customers of a bank with current accounts are segmented using WHO based on four aspects: profitability, cost, loyalty and credit; some of these aspects are calculated in a novel way. The results were welcome by the bank authorities.

1. Introduction

Evolutionary algorithms that are inspired from nature, have been used for searching possible solutions in the search space for decades [1]. They are incorporated to solve optimization problems and in this way, they have succeeded in achieving high quality results in a considerable number of articles such as [2-4].

The evolutionary algorithms can be divided into two categories: (i) The algorithms that are inspired directly from the Darwin’s theory, such as Evolutionary Programming (EP)[5], Genetic Programming (GP)[6], Genetic Algorithm (GA)[7] and Differential Evolution (DE)[8] that use operations such as crossover and mutation; and (ii) the algorithms that are inspired indirectly from the Darwin’s theory. These types of evolutionary algorithms are mostly imitation-based, such as Particle Swarm Optimization (PSO) [9] in which the members of population imitate each other, or are sign-based, such as Ant Colony Optimization (ACO) [10-12] in which there exists a global storage space for the members.

The algorithms that are inspired indirectly from the Darwin's theory are generally inspired from the way of livings of animals, insects, or birds. For example, ACO is inspired of the ants' foraging behavior, and in particular, how ants can find shortest paths between food sources and their nest by using pheromone [13]. In the following, a set of evolutionary algorithms that are inspired from nature are listed in Table 1. All of these algorithms are swarm based and use a population of individuals to solve the problems.

Table 1: Inspired evolutionary algorithms

ID	Alg. Name	Inspired from	ID	Alg. Name	Inspired of
1	ACO [10-12]	Ants	8	CSO [14]	Cats
2	ABC [15]	Bees	9	FOA [16]	Fruit Flies
3	KH [17-21]	Krills	10	CSA [22]	Cuckoos
4	TCO [23]	Termites	11	BFOA [24]	Bacteria
5	PSO [9]	Birds	12	AFSA [25]	Fish
6	FA [26], GSO[27]	Fireflies	13	BA [28]	Bats
7	SFL [29]	Shuffled Frogs	14	LPO [30]	Lions

GAs makes use of Darwin's natural selection theory to discover the best formula for prediction of patterns, or matching them [31]. The genetic algorithm uses a data structure which is similar to a chromosome to represent a possible solution. In each iteration, the algorithm selects some chromosomes, usually based on a fitness function. Then, a crossover operation is performed on the chromosomes, and a new population is generated. After that, mutation is applied on the produced chromosomes. The children that are generated by crossover and mutation may replace chromosomes that are present in the population, usually by comparing their fitness values. The population evolves by time, and converges to near optimal chromosomes. In data mining, GA has been used for clustering as well [32, 33].

Evolutionary Programming (EP) is another strategy to find near optimal solutions. Its emphasis is on behavioral linkage of children and parents. Another method is Genetic Programming (GP) in which the chromosomes are trees and represent computer programs. It also has crossover and mutation operators. Another algorithm based on swarms is Differential Evolution (DE) which also has crossover, mutation, and selection. The main difference between DE and GA is the emphasis of DE on mutation. In Particle Swarm Optimization (PSO), the solutions are called particles [9]. Each particle moves in the space, and has two values: velocity and location, which are usually vectors. The particles move in the search space by their velocity, and their location changes. Each particles knows the best location it has been in, and the best location of all of the particles. The velocity of a particle is computed based on these two experiences. PSO has been used in clustering in some researches [34-36].

Another common method is Ant Colony Optimizatin (ACO) which is used to find solutions for difficult problems. Real ants lay a pheromone trail for other ants to find good resources of food. The ants tend to choose paths with high amounts of pheromone. Another algorithm which is inspired of insects is Artificial Bee Colony (ABC) which makes use of honey bee's intelligence. In this algorithm, employed bees search for food, and share their location with onlooker bees. The onlooker bees usually choose high quality food sources. There are scout bees as well, which search for new food sources. The Krill Herd algorithm (KH) is inspired from krills. The krills move according to their minimum distance from food, and the density of krill herds. It is possible to tune the exploration and exploitation of this algorithm using parameters. However, the high number of parameters is seen as a disadvantage of this algorithm [18].

Termite Colony Optimization (TCO) is inspired from termites [37]. Termites tend to go to places with higher pheromones. The Glowworm Swarm Optimization (GSO) can find multiple solutions which are near optimum [27]. The Shuffled Frog-Leaping method is another swarm-based technique which incorporates natural memetics [29]. Another swarm-based method is Cat Swarm Optimization (CSO) which consists of two sub-models, based on the cats behaviours [14]. Fruitfly Optimization Algorithm (FOA), fruitflies get a direction and a distance to search for optimized solutions. Cuckoo Search Algorithm is inspired from the cuckoos laying eggs in nests [38].

Bacterial Foraging Optimization Algorithm is inspired from bacteria such as M.xanthus and E.coli [39]. Another method is AFSA which is taken from social behaviors of fish, and has numerous advantages such as high accuracy and high convergence speed [40]. The Bat Algorithm (BA) is based on the lives of bats. Lion Pride Optimizer (LPO) is inspired by the social interactions of lions, and can solve single or multi-variable problems. Table 1 shows a summary of some of the aforementioned algorithms.

These algorithms have different implementations of exploration and exploitation. Some of these algorithms have been compared in [38] on optimization problems. These problems are in 8 categories: Unimodal and Separable (US), Unimodal and Inseparable (UI), Multimodal and Separable (MS), Multimodal and Inseparable (MI), Multimodal (M), Unimodal (U), Separable (S), and Inseparable (I). The results can be seen in Table 2, in which the leftmost column is the types of problems, the second one shows the number of problems in that type, and the rest represent the wins of the algorithm.

Table 2: comparison on different types of problems[38]

Category	**Number of problems**	**GA**	**ACO**	**PSO**	**DE**	**ABC**	**GSO**	**CSA**
Being best performing method	30	15	13	19	24	6	0	3
Unimodal Separable (US)	2	0	0	1	1	0	0	0
Unimodal Inseparable (UI)	10	3	3	6	7	2	0	1
Multimodal Separable (MS)	6	4	3	4	5	3	0	2
Multimodal Inseparable (MI)	12	8	7	8	11	1	0	0
Unimodal (U)	12	3	3	7	8	2	0	1
Multimodal (M)	18	12	10	12	16	4	0	2
Separable (S)	8	4	3	5	6	3	0	2
Inseparable (I)	22	11	10	14	18	3	0	1

In this paper, a new evolutionary algorithm is proposed, called WHO (Wildebeest Herd Optimization) and inspired from splendid life of wildebeests. These huge mammals live in a herd with a population of over 1 million. They live socially alongside each other, and finding food, water and peace in this huge herd is not easy for them. But how can they successfully survive? Most of the researchers argue that the key to the success of these creatures is in their migration. One of the differences between WHO and other algorithms is that in the algorithm, the more mature wildebeests are attracted to the more mobile wildebeests, and vice versa. The attraction of the more mature wildebeests to the more mobile wildebeests in the first iterations of the algorithm leads to higher exploitation, and when the algorithm is likely to be trapped in the local optimum, the exploration occurs, since the members of the population that are in the local optimum have low mobility (but high fitness) and they are attracted to the members that have high mobility values. In fact the WHO algorithm imitates the important features of their migration. WHO will be discussed in detail in Section 3.1. After validating of WHO, this algorithm is examined on a real world problem.

One of the advantages of evolutionary algorithms is their ability to be used in data mining. In fact, knowledge extraction using evolutionary algorithms is the one of the hot spots in research in data mining. These algorithms can perform well in numerous aspects of data mining, such as classifications, clustering, correlation, and anomaly detection [18-21, 41-45]. The papers such as [46-52] can help experts with granular optimization techniques, granular clustering techniques and granular classification techniques. The WHO is also examined in a data mining task in Customer Relationship Management (CRM) context.

CRM is a strategy to have customized relationship with customers, and it can help to create value for companies [53]. [54-56] are some of the papers that published in this area. One of the tasks in CRM is segmentation of customers, in which the customers are divided into homogeneous groups with similar traits [57]. [58-61] are some of the papers that are published in this context. Customer segmentation based on Customer Lifetime Value (CLV) and RFM (Recency, Frequency, Monetary) are the important issues in CRM that have been studied in numerous papers, such as [62-65]. Some researchers have also worked on bank customers, such as [66, 67]. In these papers, CLV and RFM are considered in segmenting banks customers.

Nowadays, using data mining techniques has become widespread in customer relationship management. Data mining answers numerous business questions in a proper time; the questions that finding their answers were very time-consuming before [68]. For instance, now clustering algorithms such as k-means [69] can be used in a fast manner for customer segmentation.

Customer segmentation is important because all of the customers are not of the same value for the company. Hence, the companies should be able to identify the more important and valuable customers in order to try to increase their profitability and loyalty, and define some policies for the less important customers.

The rest of the paper is organized as follows: In the next section, the basic concepts are explained which includes K-Means and the usage of evolutionary algorithms in the clustering. Section 3 consists of two sub-sections. In the first, the WHO algorithm is explained. The WHO chooses the path for wildebeests based on the following: (i) the best path that each wildebeest knows and follows; (ii) the location of the mature wildebeests in the crowd; and (iii) the direction of wildebeests with high mobility. Outlier detection is needed before customer segmentation, and hence, an approach is provided for detecting outliers in the second sub-section. In section 4, the WHO is studied extensively and it is examined on optimization problems to assess its performance in different search spaces. Then it will be assessed in segmentation in UCI datasets so its performance can be evaluated in clustering. Finally, in the most important application, it will be used for segmentation of bank customers, which is a real world project. The results of WHO are compared to those of standard GA and PSO algorithms. Finally, conclusions are made and future works are provided in section 5.

2. Basic Concepts

In this section, the K-Means algorithm is explained and the method of clustering by using evolutionary algorithms is described.

1.1. K-means

The K-means method [69] is one of the fastest clustering methods in the data mining. This method is simple, yet is considered as a base method for numerous other clustering methods, such as fuzzy k-means. In K-means, first *k* members are selected randomly from the *n* members as the centroids (k is the number of clusters). Then, the $n - k$ remaining members are allocated to the nearest centroid. After this allocation, all of the centroids are recalculated and the members are reallocated to the centroids, and this process continues until the centroids do not change.

1.2. How to use evolutionary algorithms in clustering of datasets

In order to use evolutionary algorithms in clustering problems, first we should consider how to represent the individuals in population. In many researches in the literature, a structure has been used that has n . m dimensions. n represents the number of features in the dataset, and m is the number of clusters. Figure 1 shows an individual of population of evolutionary algorithms. Each cluster is shown with its centroid and each centroid consists of features values.

Centroid 1	Centroid 2	Centroid i	….	Centroid m
$[f_1, f_2, \dots, f_n]$	$[f_1, f_2, \dots, f_n]$	$[f_1, f_2, \dots, f_n]$	$[f_1, f_2, \dots, f_n]$	$[f_1, f_2, \dots, f_n]$

Figure1 : An Individual of Population

The second important consideration is the fitness function that should be specific to the clustering problems. Numerous fitness functions for clustering have been proposed, and they generally try to maximize the inter-cluster distance and minimize the intra-cluster distance. In the next sections, the fitness function used in our research will be explained.

3. The Proposed Method

This section consists of two sub-sections. In the first, the WHO algorithm is explained. Since outlier detection is needed before customer segmentation, an approach is provided for detecting outliers in the second sub-section.

3.1. Wildebeests Herd Optimization

The Serengeti ecosystem is one of the most fascinating and mysterious locations on this planet[1]. This ecosystem involves a region in Africa, that ranges from north of Tanzania to southwest of Kenya. Wildebeests in this region take long journeys to find better grassland and meadows. Every year, about 1.5 million wildebeests travel thousands of kilometers to search for better habitats and escape from the heat and the drought of Kenya and Tanzania, alongside gazelles and zebras. The group migration of wildebeests and other animals alongside them makes the journey safer. The wildebeests use their swarm intelligence to analyze the situation and take different paths for migration each year. This makes them safer, because the hunter animals such as lions, hyenas and alligators cannot predict their path. More information about the lives of wildebeest is presented in [70-72].

It is very interesting that these huge mammals with a population of over 1 million can live socially and successfully alongside each other. Finding food, water and peace, and surviving is not easy for them in this huge herd. Migration is the key to the survival of these creatures, according to most researchers. Migration is an inseparable aspect of their lives; without it, the

[1] http://www.serengeti.org/

food resources would diminish soon, and the hunters would find their locations and hunt them. The mortality rate in a stationary herd is twice the moving herd.

In this ecosystem, the mature wildebeests (with high fitness) have the right to mate, and for them, their mate should be migrating as well. In these situations, moving herds with high populations appear. In the high population and high moving herds, the children are born in the path. This has several advantages: (i) the children learn the paths of migration and remember their advantages and risks; (ii) being in move prevents them from being sighted by the hunters; (iii) in a high population and moving herd, the number of mature wildebeests is high and this leads to better protection and education for children; and (iv) the less fit children may be hunted by alligators or drowned when crossing the rivers. This helps the whole herd survive because the hunters can have their feast from the less fit children and even parents, and the fitter creatures can survive.

Wildebeests focus on three points in choosing their path:

(i) The best path that each wildebeest knows and follows;
(ii) The location of the mature wildebeests in the crowd;
(iii) The direction of wildebeests with high mobility.

They choose their next location based on these three criteria.

In this paper, considering the behavior of wildebeests and inspiring from their wonderful lives, we propose a new optimization algorithm. Modeling of behavior of wildebeests is similar to the both types of evolutionary algorithms that are inspired directly and indirectly from the Darwin's theory. The algorithms inspired directly from the Darwin's theory, such as GA, perform crossover and mutation, and in the algorithms that are inspired indirectly from the Darwin's theory, such as PSO, individuals imitate each other by using local and global best locations.

The wildebeests tend to move towards the best local and global locations and this is similar to PSO. The birth of children in the way, and having high mobility and maturity in the herd can be achieved by using crossover and mutation which is similar to GA. The crossover is accomplished such that the more mature wildebeests are attracted to the more mobile wildebeests, and vice versa. For example, one of the two parents for crossing over can be chosen from mature wildebeests and those that have high mobility. The mobility can be measured by the amount of improvements in consecutive generations that will be explained later. The attraction of the more mature wildebeests to the more mobile wildebeests has two different effects; this means that the attraction in the first iterations of the algorithm leads to higher exploitation, and when the algorithm is likely to be trapped in the local optimum, the exploration occurs, since the members of the population that are in the local optimum have low mobility (but high fitness) and they are attracted to the members that have high mobility values. Considering the mobility may decrease or increase the fitness values of children, but overall, makes them escape the local optimum and increases the chance of finding the global optimum. Also, crossover of mature wildebeests with the wildebeests that are mobile causes the location of the newborn children not to be always around the mature wildebeests; because of late convergence, this may increase the runtime of the algorithm but makes the algorithm better explore the search space. The children can be more mature and mobile and can have the ability to increase their fitness. This is the same as wildebeests' life style that children are born on the migration path. On the other hand, the children with low maturity and mobility die as in the real-world scenario, and this is done by the replacement function in the algorithm. The mobility and maturity are defined as follows:

Mobility: In order to calculate the mobility of each member of the wildebeests' population, *M* (an input parameter) consecutive fitness values of a member are taken and the square of their correlation coefficient is calculated. Correlation coefficient yields values between -1 and 1, and when it is squared, the result will be between 0 and 1. It shows the mobility of the members. If the fitness increases or decreases very rapidly, the square of correlation coefficient (mobility) will be close to 1.

Maturity: It is the fitness of each wildebeest. Since WHO will be used in clustering task as well, the fitness of an individual is defined based on the type of the problem. In this way, if the problem is optimization, the output of the functions are considered as the fitness value, and if it is a data mining task (such as clustering, classification, association rule mining, …), a function should be defined for the whole input dataset and defined as the fitness function. One of the differences between optimization problems and data mining tasks using evolutionary algorithms is in the fitness calculation. In the optimization tasks, just a function value is calculated, but in the data mining problems, there is at least a dataset that should be explored in each iteration per each individual in the population and exploring the dataset generally increases the time needed to solve the problem.

Here, the case study of the paper is the segmentation of customers that needs to apply clustering methods. Hence, we should find a proper measure for clustering the dataset. For segmentation, numerous fitness functions have been proposed. All of them try to maximize the inter-cluster distance and minimize the intra-cluster distance. Between them, one of the best functions is Sum of Squared Errors (SSE). Here, we use SSE as the fitness function. It is calculated by Equation 1:

$$SSE = \sum_{i=1}^{n_c} (x - c_i)^2 \qquad (1)$$

n_c is the number of clusters and c_i is the centroid of the ith cluster. To calculate the distance, the Euclidean distance is used which is as follows:

$$\sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (2)$$

in which, p_i and q_i are the ith values from different vectors. But why we have used only SSE in the fitness function, not considering the inter-cluster distance? The reason is that minimizing SSE leads to maximizing Sum of Squares Between (SSB) [73].

In this case that SSB is not considered in fitness calculation, the time of execution of clustering on bank data will be lower.

The pseudo code for the WHO is presented in Pseudo code 1:

```
Function WHO (Problem) returns best result as a optimum state
Input: Pop_size, Problem_size, LU_vector, P_Cross, P_Mut, $\alpha$, $\beta$,M, Inertia, Tourn_size, N_Replacement, RR_rate
Output: W_Gbest

Population← InitializePopulation(Pop_size, Problem_size, LU_vector)
EvaluatePopulation(Problem, Population)
W_Gbest← GetBest(by fitness, Population)
While (not satisfied conditions){
    W_Tbest← GetBestWithTournament(by fitness, Population, Tourn_size)
    CalculateMotions(Population, W_Tbest, LU_vector)
    MoveWildebeest(Population)
    EvaluatePopulation(Problem, Population)
    Parents ← SelectParents(Population, Tourn_size ) // according to fitness and R
    Children ← DoCrossover(Parents, P_Cross, LU_vector)
    DoMutation(Children, P_Mut, LU_vector) // on X and M vectors
    EvaluatePopulation(Problem, Children)
    For ( i=0 to (1 - RR_rate) * N_Replacement) do
    {
        For( j=0 to Tourn_size) do
        {
            Fitness_Temp1← SelectByRandom(Population)
            Fitness_Temp2← SelectByRandom(Children)
        }
        Replace Worst(Fitness_Temp1) with Best(Fitness_Temp2)
    }
    For ( i=0 to RR_rate * N_Replacement) do
    {
        For( j=0 to Tourn_size) do
        {
            Mobility_Temp1← SelectByRandom(Population)
            Mobility_Temp2← SelectByRandom(Children)
        }
        Replace Worst(Mobility_Temp1) with Best(Mobility_Temp2)
    }
    W_Tbest← GetBest(by fitness, Population)
    If W_Tbest is better than W_Gbest then W_Gbest← W_Tbest}
Return W_Gbest
```

Pseudo code 1: Workmanship of the WHO

There are some parameters which must be tuned. Pop_size determines the number of individuals in population, P_Cross, P_Mut determine the probability of crossover and mutation operations respectively, α is the importance coefficient of gBest (global best), β is the importance of lBest (local best), M determines the number of consecutive fitness values for calculating the mobility of an individual, Inertia is the importance coefficient of the previous motion, Tourn_size determines the number of random individuals that should be selected in each tournament selection, N_Replacement determines the number of replacements in each generation, and RR_rate determines the importance coefficient of individuals with high fitness and individuals with high mobility in the replacement step. Some other parameters are set with the problem characteristics, Problem_size determines the number of dimensions that the problem has and LU_vector determines lower and upper limitations for each dimension.

Each individual of the population (each wildebeest) will have 4 vectors: X for keeping the current location, M for storing the Motion Step which is calculated based on the best location the individual knows and the best location of all of the individuals in the population, D is for keeping the fitness values of previous generations to compute the square of correlation coefficient (mobility) and L for keeping the best location that the member has known since the beginning of the algorithm. In the first step of the pseudo code, all of the vectors are initialized with random values according to LU_vector. In the second step, the fitness and mobility of all individuals are calculated. In the third step, the best individual of the population in terms of fitness is chosen and then the algorithm enters the evolution loop which runs until the termination criteria are met. In this loop, new generations are evolved. Here, the termination criteria are twofold: either we have reached (i) the maximum number of generations in which no improvement has been occurred; or (ii) the maximum number of generations. Each of which occurs, the algorithm stops. In the fifth step, the best individual is chosen by using tournament selection in term of fitness, and in the sixth step, the motion of members is calculated based on Equation 4:

$$M(t+1) = inertia * M(t) + \alpha\emptyset_1(gBest - X) + \beta\emptyset_2(lBest - X) \quad (4)$$

$\emptyset_1$and $\emptyset_2$ are random numbers between 0 and 1 and X is the current position. This equation in WHO for finding gBest differs from the equation in PSO. Here, the global best is chosen from the population using the tournament selection considering the highest fitness value (W_Tbest). This prevents rapid convergence. The next position is calculated by Equation 5 where *t* is time.

$$X(t+1) = X(t) + M(t+1) \quad (5)$$

After the imitation-based phase, it is time for evolutionary operations which is accomplished using single-point crossover and mutation in all of the D, M, X, and L vectors. In the eighth step, fitness and mobility of all individuals are calculated. Selecting parents is done incorporating the tournament selection. Selecting the first parent is done by using tournament selection considering fitness values, and selecting the second parent is done by using tournament selection considering mobility values. The single-point crossover is accomplished using these two parents across all of the vectors with the probability of P_Cross and two children are created. These two children inherit mobility and maturity from the parents with a probability. Then, the mutation is done on new children's vectors (on X and M vectors) with the probability of P_Mut. The described crossover process is illustrated in figure 2.

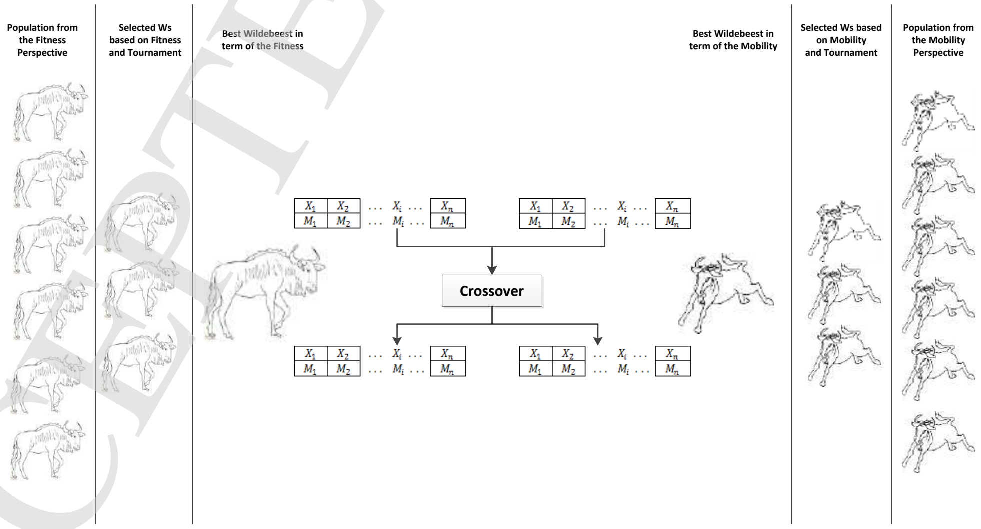

Population from the Fitness Perspective
Selected Ws based on Fitness and Tournament
Best Wildebeest in term of the Fitness
Best Wildebeest in term of the Mobility
Selected Ws based on Mobility and Tournament
Population from the Mobility Perspective
X_1 X_2 … X_i … X_n
M_1 M_2 … M_i … M_n
Crossover

Figure 2: how to choose parents in terms of the fitness and the mobility

After assessing the children in the step 12, it is time for replacement. In this step, the new generation is produced from the previous generation and the new children, as follows.

We should replace N_Replacement individuals (an input parameter) of the population. The tournament selection paradigm is applied in this section. To this end, RR_rate * N_Replacement of individuals are replaced based on the fitness function, and (1 - RR_rate) * N_Replacement of individuals are replaced based on the mobility of the population. In the selecting of the cases in terms of fitness function, we choose Tourn_size (an input parameter) individuals from the current population, and the individual with the lowest fitness value in the selected individuals is replaced by the member from the children that has been selected with a similar scheme and has the highest fitness in the selected children. In the selecting of the cases in term of the mobility function, individuals are chosen based on the mobility of the individuals, in which a tournament selection scheme is incorporated and a number of current population are selected, and the individual with the lowest mobility in the whole population is replaced with the member from the children that has been chosen similarly and has the highest mobility. Using a hybrid of imitation-based methods (motion) and evolutionary methods (crossover and mutation) causes the algorithm to explore and exploit simultaneously. However, GA generally first explores the search space and then exploits the results, and in PSO, exploitation is more emphasized than exploration. The concurrency of the exploration and the exploitation in the WHO leads to better results that can be seen in the results section.

3.2 Outlier detection approach

In this section, we consider a special case where the problem is clustering. Because of the high impact of outliers on clustering results, it is important to find them. The approach used in this paper for customer segmentation is a simple and effective approach which is based on statistical functions such as standard deviation and averaging. For each feature in the dataset, two thresholds are defined such that one of them shows the lower limit and other shows the upper limit for the outlier detection. A record that more than half of its features are out of the range is detected as outlier. Thresholds are calculated as follows:

$$Thresholds = Mean \pm 2 * ST_{Dev} \qquad (6)$$

For example, assume there is a record in dataset with 5 features. Equation 6 is calculated for each feature and so if the value of the feature is not in the range of the thresholds, it is known as an outlier feature, and if the number of outlier features exceeds the half number of features (suppose 3), then the record is known as an outlier record.

4. The Results

The case study of the paper is the segmentation of bank customers. To apply WHO on the sensitive and valuable data of the bank, we have to scrutinize the algorithm to assess its performance. First, we apply it on the standard optimization problems to assess its exploration and exploitation, and to see the strength of the algorithm in different search spaces that are explained in subsection a. Then, since this algorithm will be applied on a segmentation problem, it will be tested on standard UCI datasets with the SSE measure in subsection b, and finally it will be applied on a real bank dataset in subsection c. The algorithms are implemented in Java, and run on a personal computer with 8 gigabytes of RAM, an Intel Core i3 3.3 GHz CPU and the 64-bit version of Windows 7.

a. The results on optimization problems with different search spaces

In Table 3, problems are illustrated. The first column shows the problem number, column 2 is the name, the 3rd column shows the type (consists of US, UN, MS and MN where U means unimodal, M multimodal, S separable and N Non-Separable; in choosing problems, it is tried to choose diverse problems with different search spaces), column 4 is the number of dimensions, and columns 5 and 6 show lower bound and upper bound of function.

Table 3: The standard optimization problems

Problem	Name	Type	Dimension	L_Bound	U_Bound
1	Sphere	US	30	-100	100
2	Rosenbrock	UN	30	-30	30
3	Rastrigin	MS	30	-5.12	5.12
4	Griewank	MN	30	-600	600
5	Ackley	MN	30	-32	32

GA Standard, PSO Standard and WHO are implemented on the standard problems and results of other algorithms are taken from [74]. The results of WHO, PSO standard, GA standard and algorithms that are examined in [74] with 30 dimensions are listed in Table 4. It can be seen that when the dimensions number is low, the results are almost the same (though WHO outperforms other algorithms in some cases).

Table 4: Results and Comparison with low dimensions

	WHO			GA Standard			PSO Standard			SA			PS			PSO2011			ABC			VS		
	W_Mean	W_SD	W_Best	G_Mean	G_SD	G_Best	P_Mean	P_SD	P_Best	SA_Mean	SA_SD	SA_Best	PS_Mean	PS_SD	PS_Best	PSO2011_Mean	PSO2011_SD	PSO2011_Best	ABC_Mean	ABC_SD	ABC_Best	VS_Mean	VS_SD	VS_Best
1	2.49E-10	9.78E-10	0	2.68E-05	9.00E-06	1.04E-05	4.186338	16.98961	0	0	0	0	0	0	0	0	0	0	2.79E-16	0	2.23E-16	0	0	0
2	29.66217791	27.09248939	0.0000805	31.30275694	25.81361148	0.233216932	21303.63784	37925.98305	0.079291013	0.224618742	0.097171414	0.082077849	9.84185348	0	9.84185348	0.930212233	1.714978077	0	0.003535257	0.003314818	7.09E-05	0.367860114	1.130879848	9.43E-05
3	0.000247	0.00132763	0	1.94E-06	1.39E-06	4.58E-07	0.012629737	0.014154434	0	0	0	0	0	0	0	0.004761038	0.008047673	0	0	0	0	0.032798017	0.018570459	0.00739604
4	2.16E-12	3.15E-12	3.6E-14	0.003541125	0.001087215	0.001749061	1.41E-09	7.58E-09	4.00E-15	8.88E-16	0	8.88E-16	8.88E-16	0	8.88E-16	0.660186991	0.711496752	7.99E-15	2.45E-14	3.02E-15	2.22E-14	1.15E-14	3.61E-15	7.99E-15
5	0.000000004	2.12E-08	0	0.006452435	0.002172396	0.001914978	43.21825372	27.26928696	10.94454459	0	0	0	0	0	0	26.11016129	5.686650032	16.91429893	0	0	0	57.60799224	13.94980276	33.82857771

The parameters values for WHO, PSO Standard and GA Standard related to Table 4 are listed in Table 5. For the parameters of the rest of algorithms, please refer to [74].

Table 5: Parameters values for WHO, PSO Standard and GA related on 30-dimension problem

	WHO				PSO Standard			GA Standard		
	inertia	P_Cross	P_Mut	Tourn_size	c1	c2	inertia	Tourn_size	P_Cross	P_Mut
1	0.2	0.7	0.15	0.7	2	1.75	0.3	0.9	0.9	0.05
2	0.1	0.7	0.1	0.7	2	1.75	0.2	0.9	0.85	0.05
3	0.2	0.7	0.15	0.7	2.25	1.75	0.2	0.9	0.85	0.05
4	0.1	0.8	0.15	0.7	2	2	0.3	0.9	0.95	0.05
5	0.1	0.7	0.05	0.6	2	2	0.08	0.8	0.9	0.05

For more accurate and fair comparison, the Wilcoxon signed-rank test is used. IBM SPSS Statistics has been used to get these comparisons. Wilcoxon signed-rank test does not assume normality in the data, and it can be used when this assumption is violated and the use of the dependent t-test is inappropriate. The results of Wilcoxon Signed Ranks Test is shown in Table 6. By examining the final Test Statistics table (Table 7), we can discover whether changes led overall to a statistically significant difference in results. All of tests have been done on the best results of the algorithms.

Table 6: Ranks

Pairs	Title	N	Mean Rank	Sum of Ranks
GA $_{standard}$ - WHO	Negative Ranks	0^{a}	.00	.00
	Positive Ranks	5^{b}	3.00	15.00
	Ties	0^{c}	-	-
	Total	5	-	-
PSO $_{standard}$- WHO	Negative Ranks	1^{d}	1.00	1.00
	Positive Ranks	2^{e}	2.50	5.00
	Ties	2^{f}	-	-
	Total	5	-	-
SA - WHO	Negative Ranks	1^{g}	1.00	1.00
	Positive Ranks	1^{h}	2.00	2.00
	Ties	3^{i}	-	-
	Total	5	-	-
PS - WHO	Negative Ranks	1^{j}	1.00	1.00
	Positive Ranks	1^{k}	2.00	2.00
	Ties	3^{l}	-	-
	Total	5	-	-
PSO2011 - WHO	Negative Ranks	2^{m}	1.50	3.00
	Positive Ranks	1^{n}	3.00	3.00
	Ties	2^{o}	-	-
	Total	5	-	-
ABC - WHO	Negative Ranks	2^{p}	2.50	5.00
	Positive Ranks	1^{q}	1.00	1.00
	Ties	2^{r}	-	-
	Total	5	-	-
VS - WHO	Negative Ranks	1^{s}	1.00	1.00
	Positive Ranks	3^{t}	3.00	9.00
	Ties	1^{u}	-	-
	Total	5	-	-

a. GA < WHO
b. GA > WHO
c. GA = WHO
d. PSO < WHO
e. PSO > WHO
f. PSO = WHO
g. SA < WHO
h. SA > WHO
i. SA = WHO
j. PS < WHO
k. PS > WHO
l. PS = WHO
m. PSO2011 < WHO
n. PSO2011 > WHO
o. PSO2011 = WHO
p. ABC < WHO
q. ABC > WHO
r. ABC = WHO
s. VS < WHO
t. VS > WHO
u. VS = WHO

Table 7: Test *Statistics*[a]

	GA $_{standard}$ - WHO	PSO $_{standard}$- WHO	SA - WHO	PS - WHO	PSO2011 - WHO	ABC - WHO	VS - WHO
Z	-2.023[b]	-1.069[b]	-.447[b]	-.447[b]	.000[c]	-1.069[d]	-1.461[b]
Asymp. Sig. (2-tailed)	.043	.285	.655	.655	1.000	.285	.144
a. Wilcoxon Signed Ranks Test - b. Based on negative ranks. - c. The sum of negative ranks equals the sum of positive ranks. - d. Based on positive ranks.							

Since the clustering problem usually has a high dimension, the algorithms are run on aforementioned problems with their dimensions increased to 150. It is shown that WHO outperforms the other algorithms when number of dimensions is high.

In all of the runs, the termination criterion is if the algorithm runs for 2000 generations without improvement in the fitness. The maximum number of generations is also set to 5000. All the algorithms were run on all of the problems for 30 times, and in all of them, the population size was set to 100. In all of the runs, the parameter tuning is done using trial and error and the best parameter values have been selected.

Table 8: The results of PSO Standard on the optimization problems:

Problem	Parameters			Results - Limited No. of Generations			
	Inertia	C1	C2	Best	Average	Standard Deviation	Tot. Time
1	0.3	2	2	6.39E-08	34.122852	37.880072	**3**
2	0.3	2.2	1.5	206.77136	3986.5959	8213.116	**3**
3	0.2	2	2	4.1444128	440.15413	275.94902	**3**
4	0.25	2	2	2.1589362	11.056721	4.9056919	**9**
5	0.12	2	2	568.34582	936.53569	391.19692	**9**

The results of running PSO are shown in Table 8. The first column shows the problem. The next third columns show the best parameters (inertia is the importance of the previous velocity, C1 is the importance of gBest and C2 is the importance of lBest). The four remaining columns show the results: the best results in 30 runs, the average of results in 30 runs, the standard deviation in 30 runs, and the runtime in 30 runs (in minutes). Of the main advantages of PSO, we can mention its low number of parameters, the robustness of parameters, and the fast runtime.

The results of the genetic algorithm can be seen in Table 9. In all of the runs, the *N_Replacement* parameter is set to 0.8 of the whole population. The parameters of crossover rate, mutation rate and the number of random selections in the tournament selection can be seen in columns 2 to 4. The four remaining columns are the same as the Table 8.

Table 9: results of GA Standard on optimization problems

Problem	Parameters			Results - Limited No. of Generations			
	P_Cross	P_Mut	Tourn_size	Best	Average	Standard Deviation	Tot. Time
1	0.9	0.05	0.8*pop_size	5.0864205	5.9663757	0.567816	17
2	0.9	0.05	0.8*pop_size	330.91922	419.99084	**48.19671**	18
3	0.8	0.05	0.6*pop_size	0.0375018	0.0448809	0.003672	24
4	0.9	0.07	0.8*pop_size	1.3502583	1.4630611	0.03	29
5	0.9	0.05	0.7*pop_size	375.97954	434.25635	8.036455	25

GA Standard is also robust in its parameters. It has better results than PSO but its runtime is also higher and the algorithm is slower.

Table 10: The results of running WHO on optimization problems

Problem	Parameters				Results - Limited No. of Generations			
	P_Cross	P_Mut	Tourn_size	inertia	Best	Average	Standard Deviation	Tot. Time
1	0.5	0.05	0.7*pop_size	0.1	**1.33E-14**	**1.33E-07**	**4.72E-07**	22
2	0.5	0.03	0.7*pop_size	0.1	**148.4860254**	**272.5720654**	61.85402462	23
3	0.4	0.05	0.6*pop_size	0.2	**4.60E-14**	**0.001149487**	**0.003582231**	34
4	0.4	0.05	0.7*pop_size	0.2	**5.40E-08**	**8.91E-05**	**2.33E-04**	35
5	0.5	0.05	0.7*pop_size	0.1	**8.972856**	**15.5978696**	**3.427406697**	34

The results of WHO can be seen in Table 10. The number of replacements is similar to GA. The number of kept fitness values from previous steps for calculating the mobility of the members (M) is set to 5. α and β are set to 2. In the columns 2 to 5, crossover rate, mutation rate, number of random selections in tournament selection and the importance of previous motion are shown. The remaining columns are similar to Tables 8 and 9.

The results of WHO show that though this algorithm is slower than the previous assessed algorithms, and though its parameters are a combination of the parameters of previous algorithms, it significantly has the best results, the best average result, and the best standard deviation.

In the following two tables, WHO is compared to standard GA and standard PSO. It seems that in most types of problems (US, UN, MS and MN) with high dimensions, WHO outperforms the other two algorithms.

Table 11: Ranks

Pairs	Title	N	Mean Rank	Sum of Ranks
GA - WHO	Negative Ranks	3[a]	2.67	8.00
	Positive Ranks	2[b]	3.50	7.00
	Ties	0[c]	-	-
	Total	5	-	-
PSO - WHO	Negative Ranks	5[d]	3.00	15.00
	Positive Ranks	0[e]	.00	.00
	Ties	0[f]	-	-
	Total	5	-	-
a. GA < WHO b. GA > WHO c. GA = WHO		d. PSO < WHO e. PSO > WHO f. PSO = WHO		

Table 12: Test *Statistics*[a]

	GA - WHO	PSO - WHO
Z	-.135[b]	-2.023[b]
Asymp. Sig. (2-tailed)	.893	.043
a. Wilcoxon Signed Ranks Test b. Based on positive ranks.		

The fitness improvement charts in three algorithms on five optimization problems are demonstrated in Figure 3. It should be noted that the vertical axis is the fitness, and the horizontal axis is the number of improvement has been occurred up to 50 first improvements. For the final results, please refer to the aforementioned tables.

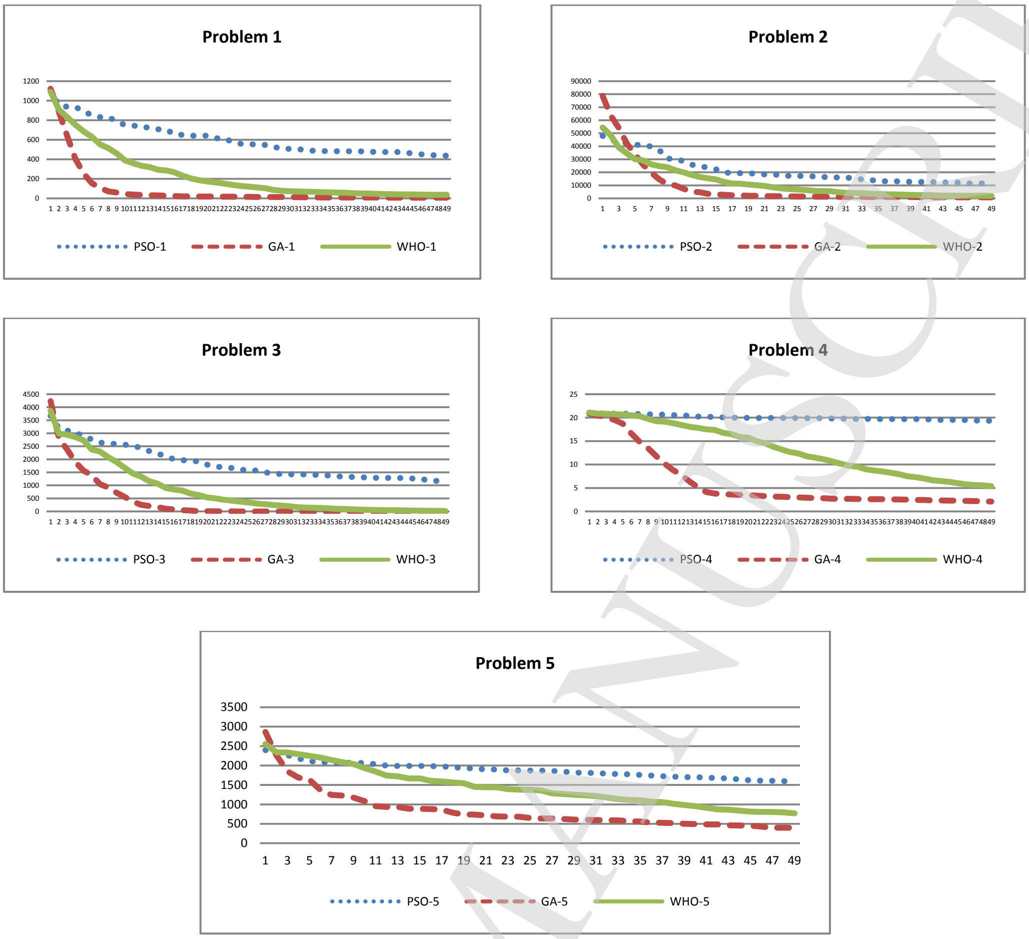


Figure3 : The fitness improvement charts in three algorithms on 5 optimization problems

There are numerous factors that play roles in exploration and exploitation of algorithms. For PSO, inertia, $\emptyset_1, \emptyset_2$, C1 and C2 could be mentioned. For GA, mutation rate, crossover rate, the number of random choices in the tournament selection and the selection method are decisive. In WHO, the aforementioned parameters play a role in exploration and exploitation, in addition to the number of stored fitness values for calculation of mobility, the method of calculating mobility, and the order of doing imitation-based and evolutionary methods. Since the number of parameters that influence exploration and exploitations is high, we analyze the parameters in whole with the above charts.

As can be seen, all of the charts follow the same pattern: the WHO is between GA and PSO. PSO begins to exploit from the beginning and does not explore highly which is necessary to escape the local optima, and GA first explores the search space and then the exploitation gets momentum and yields better results. But WHO concurrently explores and exploits. This concurrency causes the algorithm not to converge quickly, and to better explore the search space.

b. The results on the clustering problems with standard UCI datasets

In this section, the aforementioned evolutionary algorithms are applied for clustering of 7 standard UCI datasets. Some of datasets are labeled, so their label is omitted during clustering. Each of the datasets is examined with 4, 8, 12 and 24 clusters. All of datasets are numeric and before clustering, they are normalized. In the meantime, missing values are solved with averaging. The list of datasets with their characteristics is illustrated in Table 13. Please note that the number of dimensions in each individual of population is calculated as the number of features * the number of clusters. For example, if the number of clusters is set to 24, the number of dimension for Wine dataset equals to 312.

Table 13: List of Datasets

Name	Number of Features	Number of Records
Iris	4	150
Pima	8	769
Wine	13	179
Glass	9	215
Data_User_Modeling_Dataset_Hamdi Tolga KAHRAMAN	5	257
StoneFlakes	8	79
Wholesale customers data	8	440

Table 14: Parameter Tuning

DS		C4	C8	C12	C24
Iris	PSO	C1=2 , C2=2, In.=0.2	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.15
	GA	Cr=0.9, Mu=0.2, Rep.= 0.7, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.7, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.7, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.7, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.2, Cr=0.9, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.15, Cr=0.8, Mu=0.05, Rep.= 1.0, Tourna=0.9 ,	C1=2, C2=2, Ms=0.15, Cr=0.6, Mu=0.07, Rep.= 1.0, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.8, Mu=0.05, Rep.= 1.0, Tourna=0.9 ,
Pima	PSO	C1=2 , C2=2, In.=0.4	C1=2 , C2=2, In.=0.2	C1=2 , C2=2, In.=0.2	C1=2 , C2=2, In.=0.3
	GA	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.8, Tourna=0.8	Cr=0.9, Mu=0.1, Rep.= 0.7, Tourna=0.8	Cr=0.9, Mu=0.2, Rep.= 0.7, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.3, Cr=0.5, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.9, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.15, Cr=0.9, Mu=0.1, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.8, Mu=0.1, Rep.= 0.7, Tourna=0.9 ,
wine	PSO	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.2	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1
	GA	Cr=0.9, Mu=0.08, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=01, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.2, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.07, Rep.= 0.9, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.3, Cr=0.5, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.8, Mu=0.1, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.8, Mu=0.1, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.9, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,
glass	PSO	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.15	C1=2 , C2=2, In.=0.15	C1=2 , C2=2, In.=0.1
	GA	Cr=0.9, Mu=0.08, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.07, Rep.= 0.9, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.1, Cr=0.9, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.3, Cr=0.5, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.8, Mu=0.03, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.8, Mu=0.05, Rep.= 0.7, Tourna=0.9 ,
data	PSO	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.2	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1
	GA	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.05, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,
stone	PSO	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1
	GA	Cr=0.9, Mu=0.05, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,
Wholesale	PSO	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.1	C1=2 , C2=2, In.=0.15	C1=2 , C2=2, In.=0.1
	GA	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.05, Rep.= 0.9, Tourna=0.9	Cr=0.9, Mu=0.1, Rep.= 0.9, Tourna=0.9
	WHO	C1=2, C2=2, Ms=0.2, Cr=0.6, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.6, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.2, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,	C1=2, C2=2, Ms=0.1, Cr=0.5, Mu=0.08, Rep.= 0.7, Tourna=0.9 ,

The maximum number of iterations without improvement in these runs is set to 1000, and the maximum number of generations is set to 3000. The number of population is set to 50, and the number of stored fitness values for calculating mobility (M) is set to 5. The other parameter values are shown in Table 14 for 4, 8, 12 and 24 clusters. In parameter tuning, because of robustness, the results did not change significantly, hence the parameter tuning is done manually.

In Tables 15 to 18, the results on PSO, GA, K-Means and WHO are shown. It can be seen from results that the same pattern can be seen here and WHO yields relatively better results.

Table 15: The results of the PSO

DS	PSO – Limited Generations							
	C4		C8		C12		C24	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
Iris	6.6197 +- 0.6724	5.5328	4.2698+- 0.5398	3.3139	3.7929 +- 0.5448	2.9688	2.6380+- 0.3027	2.0714
Pima	140.7713 +- 16.1003	116.5422	122.5898+- 13.1039	105.8399	113.3775 +- 8.8827	93.6997	110.0458+- 8.8758	94.3241
wine	73.0503 +- 11.6387	52.0850	68.7057+- 11.6225	50.3671	58.3018 +- 7.1438	47.7383	53.1040+- 6.7688	44.0780
glass	42.7668+- 9.6696	29.0196	32.8652+- 6.2379	27.3824	31.0413 +- 5.9385	22.8022	24.8229+- 2.5967	19.9050
data	47.1915 +- 1.3821	45.4829	34.1461+- 1.1165	32.0609	28.7328+- 1.1724	26.6700	20.759 +- 1.0029	18.765
stone	16.965+- 1.4614	15.1608	14.3473+- 1.1578	11.9622	13.5615+- 0.6769	12.9477	10.4817+- 0.46109	9.7291
Wholesale	100.6506+- 39.5486	65.3174	81.5790+- 36.2591	31.8486	55.9279+- 28.3542	29.8480	38.2613+- 18.4305	24.2322

Table 16: The results of the GA

DS	GA – Limited Generations							
	C4		C8		C12		C24	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
Iris	5.6506 +- 0.2129	**5.5329**	**3.2307 +- 0.1327**	**3.1181**	**2.2571 +- 0.0995**	2.1327	1.6869 +- 0.0558	1.6127
Pima	**96.5793 +- 1.3774**	95.2378	75.8469 +- 2.2142	74.0660	71.8039 +- 1.6823	69.6250	73.1049 +- 0.9726	71.6539
wine	**45.3972 +- 0.7634**	44.8056	40.3860 +- 0.5549	39.3708	40.0553 +- 0.7735	38.9104	40.2007+- 0.6958	39.2727
glass	26.0931+- 0.7163	25.7620	17.1023+- 0.9449	15.7268	16.2057+- 0.5994	14.9784	10.6304+- 0.2653	10.2271
data	**45.9256+- 0.5126**	45.4839	32.3773+- 0.5024	**31.7836**	**25.5798+- 0.2815**	25.2334	18.3198+- 0.2092	17.9202
stone	**14.0901+- 0.1845**	**13.9093**	10.70430+- 0.2741	10.0279	9.7442+- 0.3672	9.0135	9.475+- 0.3383	9.0529
Wholesale	29.2026+- 5.0956	26.6199	19.0095+- 1.4819	16.1872	19.0766+- 1.2847	16.16101	17.2712+- 1.5926	15.7101

Table 17: The results of the K-Means

DS	Kmeans							
	C4		C8		C12		C24	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
Iris	6.7012+- 1.5779	**5.5328**	4.4819 +- 0.5681	3.2955	3.8498 +- 0.6104	2.8438	2.8373+- 0.4294	2.3124
Pima	100.0796 +- 3.6089	95.2365	83.4069+- 5.6366	73.2828	72.3495 +- 4.8974	65.4907	61.4372 +- 3.4584	56.4359
wine	46.3150 +- 1.2569	**44.7693**	**39.2250 +- 1.6564**	**37.1380**	**34.5028+- 1.6016**	**31.2841**	**29.7712 +- 2.0372**	**26.6754**
glass	26.7362 +- 2.4480	**23.7101**	22.2397+- 3.5468	17.2456	18.9189+- 2.6739	13.5867	13.5739+- 1.9921	11.0040
data	46.4706+- 0.6158	45.5940	33.3162+- 1.0578	32.0377	26.8483+- 1.0067	25.27419	18.140 +- 0.4874	17.1712
stone	14.8872+- 0.6167	13.9707	12.6482 +- 1.8888	10.8412	10.0969 +- 0.8867	**8.1160**	7.8454+- 1.2343	6.1947
Wholesale	39.9974+- 11.1537	**26.5692**	24.5845+- 4.9459	20.1022	23.0971+- 14.3077	13.7364	14.2151+- 3.4639	9.14831

Table 18: The results of the WHO

DS	WHO – Limited Generations							
	C4		C8		C12		C24	
	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
Iris	**5.6516 +- 0.2124**	**5.5328**	3.4015 +- 0.3139	3.1196	2.3032 +- 0.13042	**2.1139**	**1.3986+- 0.0767**	**1.3062**
Pima	96.9518 +- 2.2994	**95.2352**	**74.2754+- 1.0359**	**73.3164**	**66.0381+- 1.5202**	**64.2589**	**61.0406 +- 1.6866**	**55.9711**
wine	46.0921 +- 1.0317	**44.7693**	41.5196 +- 1.4179	39.4253	41.3520+- 2.0407	38.5202	38.1090 +- 1.1081	36.5249
glass	**25.6427+- 1.0546**	**23.7101**	**16.7796+- 1.4886**	**14.6731**	**14.6310 +- 1.6442**	**12.0034**	**10.3354+- 0.7697**	**9.2720**
data	46.0860 +- 0.6277	**45.4827**	**32.3715+- 0.5318**	31.9019	25.9067+- 0.5287	**25.1942**	**17.4322 +- 0.4354**	**16.8024**
stone	14.3513+- 0.5021	13.9526	**10.3687+- 0.4024**	**9.74959**	**8.8447+- 0.3946**	8.2276	**6.7468+- 0.4981**	**6.0408**
Wholesale	**27.8444+- 3.8247**	**26.5692**	**16.1032+- 2.1251**	**13.4600**	**13.9276+- 2.2780**	**10.4891**	**10.6581+- 2.1062**	**8.2141**

In the following two tables, WHO is compared to GA Standard, PSO Standard and Kmeans. The test shows that WHO is considerably outperforms others.

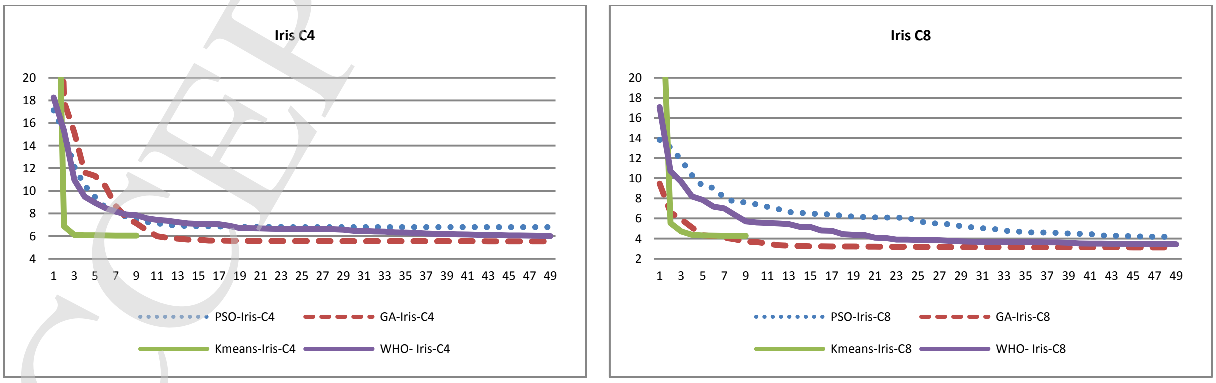
Table 19: Ranks

Pairs	Title	N	Mean Rank	Sum of Ranks
PSO - WHO	Negative Ranks	0[a]	.00	.00
	Positive Ranks	27[b]	14.00	378.00
	Ties	1[c]	-	-
	Total	28	-	-
GA - WHO	Negative Ranks	4[d]	8.00	32.00
	Positive Ranks	24[e]	15.58	374.00
	Ties	0[f]	-	-
	Total	28	-	-
Kmeans - WHO	Negative Ranks	5[g]	15.00	75.00
	Positive Ranks	19[h]	11.84	225.00
	Ties	4[i]	-	-
	Total	28	-	-
a. PSO < WHO b. PSO > WHO c. PSO = WHO d. GA < WHO e. GA > WHO	f. GA = WHO g. Kmeans < WHO h. Kmeans > WHO i. Kmeans = WHO			

Table 20: Test *Statistics*[a]

	PSO - WHO	GA - WHO	Kmeans - WHO
Z	-4.541[b]	-3.894[b]	-2.143[b]
Asymp. Sig. (2-tailed)	.000	.000	.032
a. Wilcoxon Signed Ranks Test b. Based on negative ranks.			

In the following charts (Figure 4), the SSE improvement trend on the Iris dataset is shown with 4, 8, 12 and 24 clusters. In other datasets the same patterns are observed. These charts, similar to the optimization charts, and show the improvements in the fitness function. In all of the charts, PSO starts with small improvements and exploits from the beginning and gives less importance to exploration. But the genetic algorithm first explores the search space and then gives more importance to exploitation. It yields better results than PSO, but it is more likely for it to be trapped in local optima after the beginning exploration. The K-means has huge steps and exploits very rapidly. Its exploration is done randomly. The first points chosen as centroids have a huge influence on the results. WHO tries to concurrently explore and exploit, which is because of its combination of imitation-based and evolutionary methods. WHO gives importance to exploration during the whole iterations and because of that, gains better results.



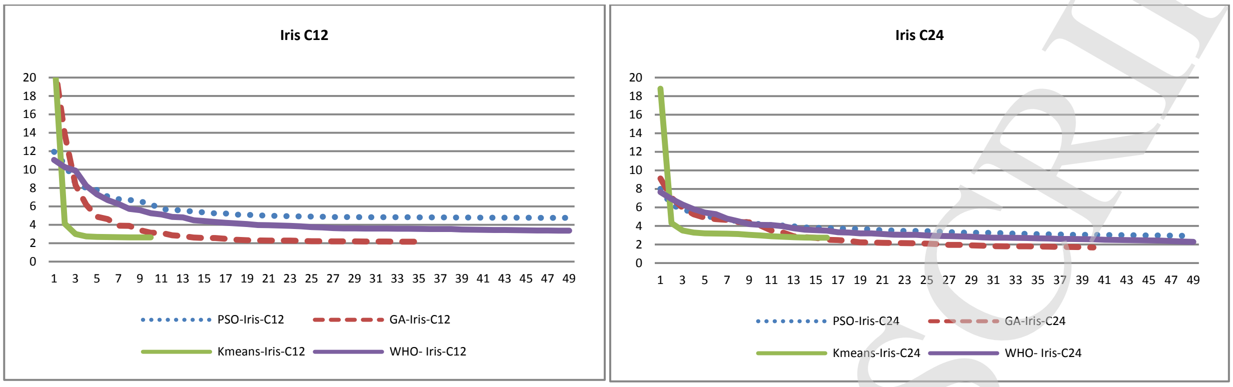


Figure4 : the SSE improvement trend on the Iris dataset

c. The results on the bank customer dataset

In this sub-section, the features used in the customer segmentation based on current accounts is described which are proposed in this paper. The dataset is taken from a bank.

I. Dataset

For creating the main dataset, the data of the current accounts in one of the banks was gathered. The data is about one year of customer transactions (in 2015), and consists only of the current accounts of the customers. In creating the dataset, four aspects of customers have been considered: profitability, cost, loyalty and credit. The mentioned aspects are calculated as follows:

The profitability is obtained based on the durability of money in customer account. For this purpose, the average of daily balance is calculated for each customer.

The cost of each customer is the sum of transactions costs done using different channels. The different channels consist of the bank branches, the mobile bank, ATM, the phone bank, Internet and Check. We consider costs coefficient of each transaction in each channel as follows as the managers of the bank suggested:

Table 21: Costs Coefficients of the transactions

	Branches	ATM	Phone Bank	Mobile Bank	Internet	Check
Cost	0.45	0.05	0.15	0.03	0.02	0.30

For each customer, the cost is calculated according to the Table 21 for a year.

The loyalty of a customer is calculated as the sum of correlation coefficients of the number of transactions, and the average yearly balances, as follows (Equation 7). It is calculated on a yearly basis:

$$L = \frac{L_{1+} L_2}{2} \qquad (7)$$

$$L_1 = \frac{\frac{\sum_{i=1}^{12} R_{fi}}{12} + R_{fy}}{2}$$

$$L_2 = \frac{\frac{\sum_{i=1}^{12} R_{mi}}{12} + R_{my}}{2}$$

where L is the customer loyalty value, L_1 is trend of the number of customer transactions, L_2 is trend of the customer money durability, R_{fi} is the correlation coefficient of the number of customer transactions in different months, R_{fy} is the correlation coefficient of 12 correlation coefficients calculated from the R_{fi}, R_{mi} is the correlation coefficient of average daily balance in different months, and $R_{my}$is the correlation coefficient of 12 resulting numbers from R_{mi}. The mathematical formula for computing ***r*** is presented in Equation 8.

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n\sum x^2 - (\sum x)^2} \cdot \sqrt{n\sum y^2 - (\sum y)^2}} \qquad (8)$$

Here, *n* is the number of pairs of data. *x* is the data and *y* is time.

One of the important points in calculating the loyalty is the combination of number of transactions trend and durability of money trend, which clearly represents the loyalty. Also, we have considered these trends as monthly and yearly to better show the loyalty.

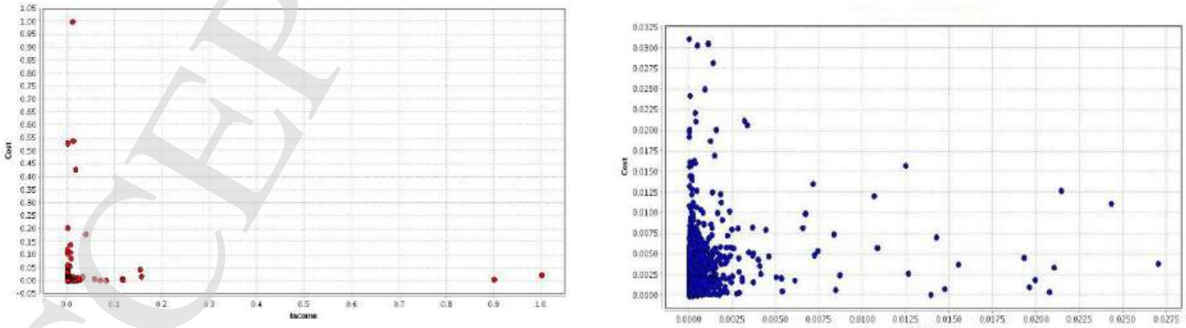
The credit of a customer is calculated based on the total amount of their dud checks in the period.

From our point of view, these four aspects of customer can properly show a customer's value for a bank. With these measures of customer aspects, the bank can design customized products for different groups.

II. Results

Before applying the algorithm, data pre-processing is needed to be done. Because of different scales of features, the Euclidean distance cannot work fairly then the features are normalized between 0 and 1.

The outliers of bank data were grouped together by using the approach presented in section 3.2, and the clustering is done on the records that are not outliers. Outliers have a huge impact on clustering; outliers cause building clusters with few members. In Figure 5, sampled bank data with outliers and without outliers (after outlier detection with our approach) is illustrated. As it can be seen in the left scatter chart, which is denser than the right scatter chart, with outliers, it is possible to have clusters with few members. In Figure 4, the vertical axis shows customers profitability and the horizontal axis shows customers cost.



Sample data with outliers

Sample data without outliers

Figure 5: Sample Data with and without Outliers

On one hand, all four aspects of customer (profitability, cost, loyalty and credit) are equally important and on the other hand, practically, the loyalty has greater impact on clustering. For this reason, in calculating of the Euclidean distance, weights were given to features. We have set weight of loyalty to 0.2 and others to 1.

The number of clusters is set to 4. Obtained SSEs for four algorithms on real bank data are illustrated in Table 22.

Table 22: SSE Results on Real Bank Data

Algorithm	K-means		PSO		GA		WHO	
Result	Avg.	Best	Avg.	Best	Avg.	Best	Avg.	Best
	151.3573 +- 29.8622	61.7706	161.3114 +- 1.90734	161.3114	23.66931 +- 3.1472	21.2331	**19.2391 +- 1.7894**	**18.0566**

As shown in Figure 6, 0.35 percent of customers are identified as the outlier cluster and were grouped in a distinct cluster for further research. Other customers are distributed in 4 clusters. In the following pie chart, distribution of customers is shown. 64.32 percent of customers (except outliers) are identified as Cluster 1 that had lower profitability, relatively high cost, and neither good nor bad loyalty. 35.00 percent of customers are identified as Cluster 2 that profitability and cost of them are like Cluster 1 but this cluster had better loyalty compared to Cluster 1. 0.46 percent of customers are identified as Cluster 3 with relatively high profitability and low cost. This cluster had higher loyalty in compared to Cluster 1 and 2. 0.22 percent of customers are identified as Cluster 4 with high profitability and low cost. Besides this, the cluster 4 had higher loyalty compared to other clusters.

In term of credit, the Clusters 4, 3, 2 and 1 are sorted from best to worst.

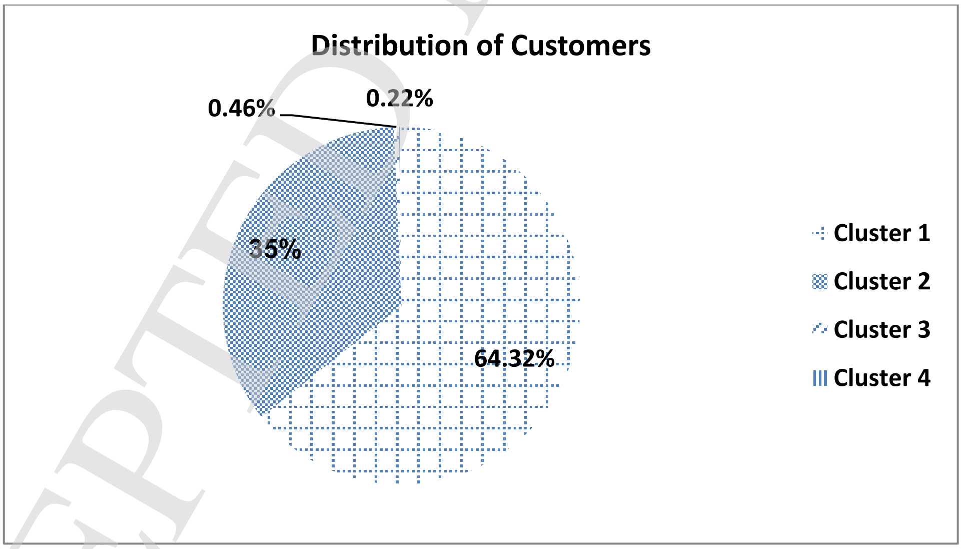


Figure 6: Distribution of Customers in Clusters

5. Conclusion

In this paper, a new evolutionary algorithm was proposed that we call WHO. The WHO algorithm is inspired from the glorious life of wildebeests, and has the ability to explore and exploit concurrently. To prove its superior performance, it was tested in the optimization and clustering problems and it yielded better results than GA and PSO. Then the data was applied to the bank customer's data; the segmentation of customers of a bank has been done based on profitability, cost, loyalty, and credit, and the results were welcome by the bank authorities.

The WHO algorithm is capable in searching and data mining problems. This algorithm can be applied for classification, association rule discovery, outlier detection, and so on.

One of the disadvantages of the proposed algorithm is the large number of the parameters compared to other well-known algorithms. In the future, we will try to decrease the number of parameters and in the meantime, we will apply the proposed algorithm on some data mining tasks in several contexts such as bioinformatics, medicine, and other financial services.

6. References

[1] D. B. Fogel, "Introduction to evolutionary computation," *Evolutionary computation,* vol. 1, pp. 1-3, 2000.

[2] M. Wolfram, A.-K. Marten, and D. Westermann, "A comparative study of evolutionary algorithms for phase shifting transformer setting optimization," in *Energy Conference (ENERGYCON), 2016 IEEE International*, 2016, pp. 1-6.

[3] H. Z. Sabzi, D. Humberson, S. Abudu, and J. P. King, "Optimization of adaptive fuzzy logic controller using novel combined evolutionary algorithms, and its application in Diez Lagos flood controlling system, Southern New Mexico," *Expert Systems with Applications,* vol. 43, pp. 154-164, 2016.

[4] A. H. Gandomi, A. R. Kashani, D. A. Roke, and M. Mousavi, "Optimization of retaining wall design using evolutionary algorithms," *Structural and Multidisciplinary Optimization,* vol. 55, pp. 809-825, 2017.

[5] X. Yao and Y. Liu, "Fast Evolutionary Programming," *Evolutionary Programming,* vol. 3, pp. 451-460, 1996.

[6] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection* vol. 1: MIT press, 1992.

[7] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing,* vol. 4, pp. 65-85, 1994.

[8] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.

[9] G.-c. Chen and J.-s. Yu, "Particle swarm optimization algorithm," *INFORMATION AND CONTROL-SHENYANG-,* vol. 34, p. 318, 2005.

[10] N. M. Al Salami, "Ant colony optimization algorithm," *UbiCC Journal,* vol. 4, pp. 823-826, 2009.

[11] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine,* vol. 1, pp. 28-39, 2006.

[12] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999, pp. 1470-1477.

[13] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life reviews,* vol. 2, pp. 353-373, 2005.

[14] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Pacific Rim International Conference on Artificial Intelligence*, 2006, pp. 854-858.

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

[16] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems,* vol. 26, pp. 69-74, 2012.

[17] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation,* vol. 17, pp. 4831-4845, 2012.

[18] A. M. Shanghooshabad and M. S. Abadeh, "Sifter: an approach for robust fuzzy rule set discovery," *Soft Computing,* pp. 1-17, 2015.

[19] A. M. Shanghooshabad and M. S. Abadeh, "Robust medical data mining using a clustering and swarm-based framework," *International Journal of Data Mining and Bioinformatics,* vol. 14, pp. 22-39, 2016.

[20] A. M. Shanghooshabad and M. S. Abadeh, "Robust, interpretable and high quality fuzzy rule discovery using krill herd algorithm," *Journal of Intelligent & Fuzzy Systems,* vol. 30, pp. 1601-1612, 2016.

[21] A. Mohammadi, M. S. Abadeh, and H. Keshavarz, "Breast cancer detection using a multi-objective binary Krill Herd algorithm," in *Biomedical Engineering (ICBME), 2014 21th Iranian Conference on*, 2014, pp. 128-133.

[22] R. Rajabioun, "Cuckoo optimization algorithm," *Applied soft computing,* vol. 11, pp. 5508-5518, 2011.

[23] R. Hedayatzadeh, F. A. Salmassi, M. Keshtgari, R. Akbari, and K. Ziarati, "Termite colony optimization: A novel approach for optimizing continuous problems," in *2010 18th Iranian Conference on Electrical Engineering*, 2010, pp. 553-558.

[24] S. Das, A. Biswas, S. Dasgupta, and A. Abraham, "Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications," in *Foundations of Computational Intelligence Volume 3*, ed: Springer, 2009, pp. 23-55.

[25] X.-l. Li, F. Lu, G.-h. Tian, and J.-x. QIAN, "Applications of artificial fish school algorithm in combinatorial optimization problems [J]," *Journal of Shandong University (Engineering Science),* vol. 5, p. 015, 2004.

[26] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International Symposium on Stochastic Algorithms*, 2009, pp. 169-178.

[27] K. Krishnanand and D. Ghose, "Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications," *Multiagent and Grid Systems,* vol. 2, pp. 209-222, 2006.

[28] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, ed: Springer, 2010, pp. 65-74.

[29] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization,* vol. 38, pp. 129-154, 2006.

[30] B. Wang, X. Jin, and B. Cheng, "Lion pride optimizer: An optimization algorithm inspired by lion pride behavior," *Science China Information Sciences,* vol. 55, pp. 2369-2389, 2012.

[31] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion wesley,* vol. 1989, p. 102, 1989.

[32] D. Doval, S. Mancoridis, and B. S. Mitchell, "Automatic clustering of software systems using a genetic algorithm," in *Software Technology and Engineering Practice, 1999. STEP'99. Proceedings*, 1999, pp. 73-81.

[33] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern recognition,* vol. 33, pp. 1455-1465, 2000.

[34] T. Küçükdeniz and Ş. Esnaf, "Data Clustering by Particle Swarm Optimization with the Focal Particles," in *International Workshop on Machine Learning, Optimization and Big Data*, 2015, pp. 280-292.

[35] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, and L. Jiao, "Greedy discrete particle swarm optimization for large-scale social network clustering," *Information Sciences,* vol. 316, pp. 503-516, 2015.

[36] A. A. Esmin, R. A. Coelho, and S. Matwin, "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data," *Artificial Intelligence Review,* vol. 44, pp. 23-45, 2015.

[37] R. Hedayatzadeh, F. A. Salmassi, M. Keshtgari, R. Akbari, and K. Ziarati, "Termite colony optimization: A novel approach for optimizing continuous problems," in *Electrical Engineering (ICEE), 2010 18th Iranian Conference on*, 2010, pp. 553-558.

[38] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PloS one,* vol. 10, p. e0122827, 2015.

[39] J. Brownlee, *Clever algorithms: nature-inspired programming recipes*: Jason Brownlee, 2011.

[40] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review,* pp. 1-33, 2014.

[41] R. Prado, S. García-Galán, A. J. Yuste, and J. M. Expósito, "A fuzzy rule-based meta-scheduler with evolutionary learning for grid computing," *Engineering Applications of Artificial Intelligence,* vol. 23, pp. 1072-1082, 2010.

[42] A. Zibakhsh and M. S. Abadeh, "Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function," *Engineering Applications of Artificial Intelligence,* vol. 26, pp. 1274-1281, 2013.

[43] K. Sarath and V. Ravi, "Association rule mining using binary particle swarm optimization," *Engineering Applications of Artificial Intelligence,* vol. 26, pp. 1832-1840, 2013.

[44] X. Yan, Y. Zhu, W. Zou, and L. Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm," *Neurocomputing,* vol. 97, pp. 241-250, 2012.

[45] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences,* vol. 222, pp. 175-184, 2013.

[46] G. Peters and R. Weber, "DCC: a framework for dynamic granular clustering," *Granular Computing,* vol. 1, pp. 1-11, 2016.

[47] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Multi-objective evolutionary design of granular rule-based classifiers," *Granular Computing,* vol. 1, pp. 37-58, 2016.

[48] P. Lingras, F. Haider, and M. Triff, "Granular meta-clustering based on hierarchical, network, and temporal connections," *Granular Computing,* vol. 1, pp. 71-92, 2016.

[49] H. Liu, A. Gegov, and M. Cocea, "Rule-based systems: a granular computing perspective," *Granular Computing,* vol. 1, pp. 259-274, 2016.

[50] S. S. S. Ahmad and W. Pedrycz, "The development of granular rule-based systems: a study in structural model compression," *Granular Computing,* vol. 2, pp. 1-12, 2017.

[51] G. Wang, J. Yang, and J. Xu, "Granular computing: from granularity optimization to multi-granularity joint problem solving," *Granular Computing,* pp. 1-16, 2017.

[52] H. Liu and M. Cocea, "Granular computing-based approach for classification towards reduction of bias in ensemble learning," *Granular Computing,* vol. 2, pp. 131-139, 2017.

[53] V. Kumar, *Customer relationship management*: Wiley Online Library, 2010.

[54] Z. Ehsani and N. H. Hashim, "Incorporated Definition of Customer Relationship Management towards Successful Performance," *International Journal of Service Science, Management and Engineering,* vol. 1, p. 57, 2015.

[55] K. Srinivasan and S. Saravanan, "Principles and Practices of Customer Relationship Management in Ethiopian Banks," *EXCEL International Journal of Multidisciplinary Management Studies,* vol. 5, pp. 8-20, 2015.

[56] R. Rahimi and M. Kozak, "Impact of Customer Relationship Management on Customer Satisfaction: The Case of a Budget Hotel Chain," *Journal of Travel & Tourism Marketing,* pp. 1-12, 2016.

[57] J.-J. Jonker, N. Piersma, and D. Van den Poel, "Joint optimization of customer segmentation and marketing policy to maximize long-term profitability," *Expert Systems with Applications,* vol. 27, pp. 159-168, 2004.

[58] N. David, H. Schramm-Klein, O. Rank, and G. Wagner, "Customer segmentation in retailing based on retail brand patronage patterns," *The International Review of Retail, Distribution and Consumer Research,* vol. 25, pp. 449-459, 2015.

[59] R. Kuo, C. Mei, F. Zulvia, and C. Tsai, "An application of a metaheuristic algorithm-based clustering ensemble method to APP customer segmentation," *Neurocomputing,* vol. 205, pp. 116-129, 2016.

[60] A. Sharma and D. M. Lambert, "Segmentation of markets based on customer service," *International Journal of Physical Distribution & Logistics Management,* 2013.

[61] F. Hamka, H. Bouwman, M. De Reuver, and M. Kroesen, "Mobile customer segmentation based on smartphone measurement," *Telematics and Informatics,* vol. 31, pp. 220-227, 2014.

[62] S.-Y. Kim, T.-S. Jung, E.-H. Suh, and H.-S. Hwang, "Customer segmentation and strategy development based on customer lifetime value: A case study," *Expert Systems with Applications,* vol. 31, pp. 101-107, 2006.

[63] H. Hwang, T. Jung, and E. Suh, "An LTV model and customer segmentation based on customer value: a case study on the wireless telecommunication industry," *Expert Systems with Applications,* vol. 26, pp. 181-188, 2004.

[64] M. S. Kahreh, M. Tive, A. Babania, and M. Hesan, "Analyzing the applications of customer lifetime value (CLV) based on benefit segmentation for the banking sector," *Procedia-Social and Behavioral Sciences,* vol. 109, pp. 590-594, 2014.

[65] N. Delafrooz and E. Farzanfar, "Determining the customer lifetime value based on the benefit clustering in the insurance industry," *Indian Journal of science and Technology,* vol. 9, 2016.

[66] D. Zakrzewska and J. Murlewski, "Clustering algorithms for bank customer segmentation," in *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005, pp. 197-202.

[67] A. Machauer and S. Morgner, "Segmentation of bank customers by expected benefits and attitudes," *International Journal of Bank Marketing,* vol. 19, pp. 6-18, 2001.

[68] C. Rygielski, J.-C. Wang, and D. C. Yen, "Data mining techniques for customer relationship management," *Technology in society,* vol. 24, pp. 483-502, 2002.

[69] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition,* vol. 36, pp. 451-461, 2003.

[70] R. D. Estes, *The Gnu's World: Serengeti Wildebeest Ecology and Life History*: Univ of California Press, 2014.

[71] R. D. Estes, "Behaviour and life history of the wildebeest (Connochaetes taurinus Burchell)," 1966.

[72] S. McNaughton, "Serengeti migratory wildebeest: facilitation of energy flow by grazing," *Science,* vol. 191, pp. 92-94, 1976.

[73] D. M. Witten and R. Tibshirani, "A framework for feature selection in clustering," *Journal of the American Statistical Association,* 2012.

[74] B. Doğan and T. Ölmez, "A new metaheuristic for numerical function optimization: Vortex Search algorithm," *Information Sciences,* vol. 293, pp. 125-145, 2015.