An Effective Search Economics Based Algorithm for Feature Selection

Cheng-Chia Huang
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 80424, Taiwan, R.O.C.
m073040024@student.nsysu.edu.tw

Ming-Chao Chiang
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 80424, Taiwan, R.O.C.
mcchiang@cse.nsysu.edu.tw

***Abstract*—The basic idea of feature selection, as the name suggests, is to select an appropriate subset of features of a problem in question so as to reduce the data complexity, thus accelerating the analysis process. Being an NP-hard problem, it has become a critical research topic in data mining for years. In this study, we propose a novel, effective metaheuristic algorithm, called search economics for feature selection, for solving this problem. The proposed algorithm is built on the idea of investing in different sectors of the market based on the potential of each sector; in other words, it is built on the idea of searching different regions in the solution space based on the so-called potential of each region. By using such a search strategy, the proposed algorithm is less likely to fall into local optimum at early iterations while at the same time retaining the search diversity during the convergence process. The simulation results show that the proposed algorithm provides a better result than all the other state-of-the-art feature selection algorithms compared in this study in solving the eighteen well-known UCI datasets.**

***Index Terms*—feature selection, metaheuristic algorithm, and search economics.**

I. Introduction

Unlike the other data compression or sampling methods for reducing the data complexity, there are two different ways to reduce the dimensionality of data; namely, "feature extraction" and "feature selection" [1]. The so-called feature extraction, such as principal component analysis (PCA) [2] and linear discriminant analysis (LDA) [3], is usually used to change the original features by projecting the original data from high to low or from low to high dimensional space. In this way, the raw data will be replaced by the new data that preserve enough information of the raw data for the analysis. Unlike the feature extraction, the so-called feature selection will select a subset of features out of all the features in the original dataset. In this way, feature selection will be able to filter out irrelevant features in such a way that only critical features are preserved for the analysis so that the computation time of the analysis process can be reduced [4]. That is why feature selection can be regarded as the preprocessing step of a classification algorithm (e.g., support vector machine or k-nearest neighbor) for prediction or classification. Such a method has been widely used in, say, social media analysis [5] and network intrusion detection [6].

The feature selection generally can be divided into two kinds [7]—filter and wrapper. Independent of the classification algorithm, the filter method works by first measuring all the features based on either statistics of variability or significance between features. Then, it will select a subset of features that are more important than the others based on a predefined threshold. Finally, it will put the selected subset of features into the classification algorithm to verify its accuracy. The wrapper method will first search for possible subsets of features and then use a classification algorithm to evaluate the objective value of the selected features. Different from the filter method, the wrapper method will reuse the results of the selection step, thus repeating the search, classification, and evaluation steps until the termination condition is met.

Since the feature selection problem (FSP) is an NP-hard problem [7], it is impossible to use an exhaustive search algorithm to solve this problem within a reasonable time. Eventually, many search methods have been presented to solve this problem, such as heuristic search and random search [8]. Several recent studies have shown that metaheuristic algorithms, such as genetic algorithm (GA) [9] and ant colony optimization (ACO) [10], provide a much better result than heuristic search algorithms in solving the feature selection problem. An effective metaheuristic algorithm for solving the FSP based on the search economics (SE) [11], [12] will be presented in this paper because several previous studies have shown that search economics has a good search ability for combinatorial optimization problems.

The remainder of this paper is organized as follows: Section II begins with the problem definition of feature selection, followed by a brief introduction to some search algorithms for solving this problem. Section III will provide the basic idea and describe in detail the proposed algorithm for the feature selection problem. Section IV begins with a description of the datasets and parameter settings. Then, the simulation results of the proposed algorithm with other six feature selection algorithms for eighteen datasets are shown. Finally, the conclusions and future work are drawn in Section V.

II. Related Works

A. The Problem Definition of Feature Selection

The goal of feature selection is to reduce the data complexity, by using only features that most represent the data for the analysis. One of the reasons is that the classification results

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

may be degraded by features that are irrelevant or are noisy. Generally speaking, a feature selection algorithm has two major objectives: (1) maximize the accuracy of a classification algorithm and (2) minimize the number of features to speed up the computation time of a classification algorithm. The objective value of a feature selection problem [13] can be defined as follows:

$$\arg\min_{s \subseteq S} f(s) = \alpha \cdot E_R(s) + \beta \cdot \frac{|s|}{|S|}, \tag{1}$$

where S denotes the set of all features; $s \subseteq S$ a subset of S; $E_R(s)$ the error rate of the classification algorithm; $|s|$ the number of features in s; and $|S|$ the number of features in S. Besides, $\alpha \in [0, 1]$ denotes the weight of the error rate while β the weight of the ratio of the selected features. In this study, α is set equal to 0.99 while β is set equal to $1 - \alpha$; that is, to 0.01 [14]. This definition states that a smaller objective value implies a lower error rate and a fewer number of features; that is, a smaller objective value implies that a better subset is selected. To verify the solution s found by the feature selection algorithm, the k-nearest neighbor (KNN)[1] is used as the classification algorithm in this study, where k is set equal to 5 [14]. The objective value $f(s)$ is a weighted sum of the error rate of the classification algorithm and the ratio of the selected features to evaluate the performance of all the feature selection algorithms.

B. Metaheuristic-based Feature Selection Methods

Since most heuristic algorithms rely on a specific set of rules for finding the subset of features, it is very difficult to search most regions in the solution space. That is why most heuristic algorithms may easily fall into local optimum at early stages during the convergence process. Since most metaheuristic algorithms are capable of searching several regions in the solution space at a time, several early studies [9], [15], [10] attempted to use them for solving the FSP. A good example is the genetic algorithm (GA) [9]—a well-known metaheuristic algorithm that has been applied to the FSP. In a recent study [15], Chen et al. presented a chaos genetic feature selection optimization (CGFSO) for text categorization. The simulation results show that the CGFSO is capable of selecting important features effectively to obtain a higher classification accuracy. Moreover, Ahmed [10] presented a feature subset search algorithm based on ant colony optimization (ACO), which is inspired by the behavior of ants searching for the shortest paths to improve the accuracy of texture classification. The simulation results show that this method can find a better result than GA.

Also, several new metaheuristic algorithms presented in recent years have shown their possibilities; therefore, several studies [16], [13], [6], [17], [18] used them for solving the FSP. For example, Emary [16] presented a gray wolf optimization (GWO) algorithm for the FSP, which works by converting the gray wolf's position to a discrete value using a constant threshold, and the experimental results showed that the proposed method outperforms GA and particle swarm optimization (PSO) in terms of both the quality and the convergence speed. In a later study [13], Emary presented a binary version of GWO (BGWO), by using different initialization methods, which include small, normal, and large initialization. Moreover, a binary whale optimization algorithm (BWOA) was presented in [17] to solve the FSP, followed by an improved version for the FSP of a network intrusion detection system [6]. Mafarja [18] attempted to combine WOA with simulated annealing (SA) in such a way that the SA is used to search for the most promising regions located.

[1]The KNN has been widely used in verifying the end results of an FSP in several recent studies.

III. The Proposed Algorithm

Proposed by Tsai [11] in 2015, the search economics (SE) is a novel metaheuristic algorithm inspired by the return on investment (ROI). In other words, the underlying idea of SE is to treat the solution space as a market that can be divided into a certain number of sectors (i.e., regions) each of which has its own specific products. Then, a set of investors (i.e., searchers) will invest the limited resources they have in regions based on the so-called potential of each region. This implies that all the investors want to maximize the return on investment. As such, all the investors have to take into account many factors, such as the "objective value" of searched solutions, the "number of searchers" in a region, and the "possibility" to find a better solution at later iterations. In other words, in order not to waste the limited resources, all the investors will invest their resources in regions based on remuneration and profitability of each region and constantly adjust the resources invested in each region.

Algorithm 1 Search economics for feature selection (SEFS)

```
Initialization()
RA()
while the termination criterion is not met do
    VS()
    MR()
end while
Output()
```

Algorithm 1 shows that the proposed method is based on the search economics and is referred to as search economics for feature selection (SEFS). Like the SE, SEFS consists of three major operators—namely, the resource arrangement RA(), vision search VS(), and marketing research MR() operators—for finding out important features. The resource arrangement operator is responsible for (1) dividing the solution space (called market in SE-based algorithms) into several subspaces (called regions) according to predefined rules and (2) creating a set of sample points (called samples) for each region to describe its landscape. The vision search operator that is composed of three suboperators—transition, expected value, and determination—plays the role of searching for the solution. The marketing research takes care of adjusting the resources invested and updating the solutions searched in each region so as to provide a more precise view of the solution space. Although the proposed algorithm is based on SE, the RA() and

VS() operators of the SEFS are quite different from those of SE. The discussion that follows will give a detailed description on these operators.

A. Encoding and Initialization

As far as SEFS is concerned, a solution is encoded as $s_i = \{s_i^1, s_i^2, \ldots, s_i^D\}$ where D is the number of features, and each subsolution $s_i^d \in \{0, 1\}$ of which denotes if the d-th feature is in use or not, with $s_i^d = 1$ indicating that the d-th feature is selected; $s_i^d = 0$ indicating that the d-th feature is not selected by the classification algorithm. In this study, the solution s_i represents the i-th searcher (or investor); r_j the j-th region; and r_j^{best} the best sample (i.e., searched solution) in the region r_j. Besides, $m = \{m_{11}, m_{12}, \ldots, m_{jk}\}$ is the set of samples in all the regions. For instance, m_{jk} is the k-th sample in the j-th region. Moreover, v_{ijk} is a new set of solutions (investments) obtained by applying the transition operator to the searcher s_i and the sample m_{jk}; that is, $v_{ijk} = s_i \otimes m_{jk}$ where $\otimes$ denotes the transition operator. This implies that most transition operators of metaheuristic algorithms can be used as the transition operator of SEFS. For example, using the crossover operator of genetic algorithm as the transition operator, $v_{ijk} = \{v_{ijk}^1, v_{ijk}^2\}$ will be generated by applying the crossover operator to s_i and m_{jk}.

B. Resource Arrangement

The resource arrangement operator plays the role of dividing the solution space into a certain number of regions, randomly generating a set of solutions as the searchers, and assigning searchers to regions. Similar to the SE, it will also randomly generate a set of solutions as samples of each region. Each sample m_{jk} in a region can be regarded as a sample solution. However, the SEFS uses a different way to assign samples to regions than SE. That is, for SEFS, the value of each dimension of a sample m_{jk}, denoted m_{jk}^d, is randomly generated, as follows:

$$m_{jk}^d = \begin{cases} 1, & \text{if } P_u < Z_j, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where Z_j is the threshold defined as $Z_j = (2j-1)/(2h)$, with j being the region number and h the number of regions, and P_u a random number uniformly distributed in the range $[0, 1]$. The main difference between SEFS and SE is how the solution space is divided and how samples m are generated. For the proposed algorithm SEFS, the solution space is divided and the samples m are generated in terms of a probability while for SE, the solution space is divided in terms of a fixed part of the subsolutions and the samples m are generated and assigned to a region based on its position in the solution space. For instance, for the proposed algorithm SEFS, suppose there are four regions; the thresholds Z_1, Z_2, Z_3, and Z_4 can then be easily computed as 0.125, 0.375, 0.625, 0.875. This implies that the probability of generating a '1' in r_1 is 0.125, in r_2 0.375, in r_3 0.625, and in r_4 0.875. Moreover, as shown in Fig. 1, searchers will be randomly assigned to regions. In this example, it happens that the first searcher is assigned to the first region while the second searcher is assigned to the fourth region.

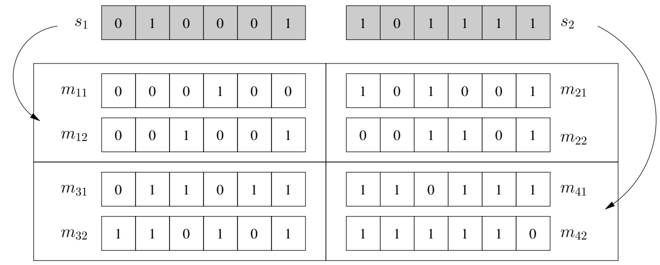

Fig. 1: The resource arrangement scheme.

C. Vision Search

As shown in Fig. 2, the vision search is composed of three major operators; namely, transition, expected value, and determination. The Transition() operator plays the role of generating new candidate solutions; the Expected_Value() operator is responsible for evaluating the potential of each region. The Determination() operator plays the role of deciding the search directions and regions at later iterations.

1) Transition: This operator will first exchange the information of each searcher with samples in all regions to generate new candidate solutions as the investments. It will then evaluate the investments of each region to identify which one has higher potential (or better chance) to find better results at later iterations. In this study, we will use the crossover and mutation mechanism of genetic algorithm as the transition operator for creating the investments. In order to maintain the characteristics of a region, searchers and samples exchange information they own using a discrete crossover defined as follows:

$$Z_c = Z_{\text{base}} + Z_{\text{range}} \times \frac{t}{t_{\max}}, \tag{3}$$

$$v_{ijk}^d = \begin{cases} s_i^d, & \text{if } P_u < Z_c, \\ m_{jk}^d, & \text{otherwise}, \end{cases} \tag{4}$$

where Z_c is the threshold for exchanging information with searchers which will increase linearly from Z_{base} to $Z_{\text{base}} + Z_{\text{range}}$ as the number of iterations increases, Z_{base} is the initial threshold, and Z_{range} is the increasing range of the threshold. Besides, t denotes the current iteration and $t_{\max}$ the maximum number of iterations. Also, v_{ijk} represents the new investments produced by s_i and m_{jk}, as shown in Fig. 3.

In this study, Z_{base} is set equal to 0.25 while Z_{range} is set equal to 0.5; therefore, Z_c will increase linearly from 0.25 to 0.75. This means that most of the information of the investments comes from the sample m_{jk} at the early stage of the convergence process. The investments v_{ijk} will preserve the characteristics of the i-th searcher and samples in the j-th region; thus, SEFS can use this information to evaluate the potential of each region. At the later stage of the convergence process, the impact of region will be less significant because

```
VS() {
    Transition()
    Expected_Value()
    Determination()
}
```

Fig. 2: Outline of the version search.

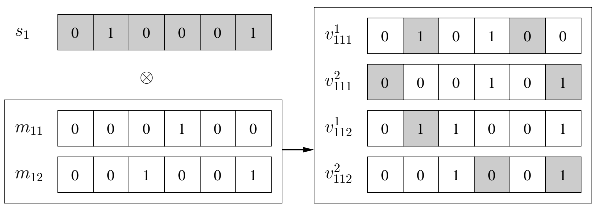

Fig. 3: The crossover scheme of SEFS.

most of the information of the investments comes from the searcher s_i. In this way, the proposed algorithm will move to the region that has the greatest potential to find the best solution.

2) Expected Value: This operator is responsible for calculating the expected value of each searcher based on the frequency, value, and experience of investment in a region. The determination operator will then use this information to decide the region that has a higher expected value for investment. The expected value is defined as follows:

$$e_{ij} = T_j V_{ij} M_j, \tag{5}$$

where e_{ij} is the expected value of the i-th searcher in the j-th region; T_j the number of times the j-th region is invested by all the searchers; V_{ij} the investment potential; i.e., the value of the newly generated investments based on the i-th searcher and the samples in the j-th region; and M_j the experience of investment in the j-th region. T_j is defined as

$$T_j = \frac{t^b_j}{t^a_j}, \tag{6}$$

where t^a_j denotes the number of times the j-th region has been invested, and t^b_j the number of times the j-th region has not been invested. They are both set equal to 1 initially and updated in the marketing research operator. This mechanism is used to prevent searchers from searching the same region for too many times, thus reducing the investments that would only waste the resources. Now, for a minimization problem, V_{ij} and M_j are defined as

$$V_{ij} = 1 - V'_{ij}, \tag{7}$$

$$M_j = 1 - M'_j, \tag{8}$$

where V'_{ij} denotes the ratio of "the sum of the objective values of the investments by the i-th searcher in the j-th region" to "the sum of the objective values of the investments by the i-th searcher in all the regions;" that is, it is defined as

$$V'_{ij} = \frac{\sum_{k=1}^{w}\left(f(v^1_{ijk}) + f(v^2_{ijk})\right)}{\sum_{l=1}^{h}\sum_{k=1}^{w}\left(f(v^1_{ilk}) + f(v^2_{ilk})\right)}, \tag{9}$$

where w is the number of samples in a region; h the number of regions while M'_j denotes the ratio of "the objective value of the so far best sample in the j-th region" to "the sum of the objective values of samples in all the regions;" that is, it is defined as

$$M'_j = \frac{f(r^{\text{best}}_j)}{\sum_{l=1}^{h}\sum_{k=1}^{w} f(m_{lk})}. \tag{10}$$

3) Determination: This operator will first select the best w investments from the newly generated investments for a region as the samples of that region. Then, each searcher will use tournament selection to select a region that has the highest expected value. Finally, the best of the selected samples will be used to update the searcher of the selected region. For instance, as shown in Fig. 4, let us suppose that the region selected by the searcher s_1 is r_3, and the samples selected for the region r_3 are m_{31} and m_{32}. Let us further suppose that the sample m_{32} is better than the sample m_{31} in terms of the objective value. Then, the sample m_{32} will be used to update the searcher s_1.

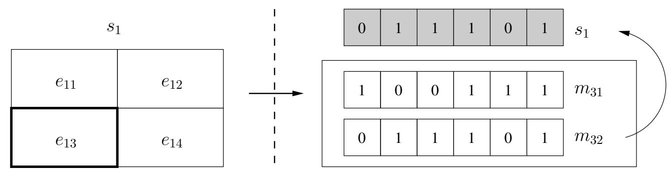

Fig. 4: The determination scheme.

D. Marketing Research

This operator is responsible for updating the current status of the market, represented by t^a_j, t^b_j, and r^{best}_j. First, all the t^b_j are increased by 1, indicating that all the regions are preset to "not selected." Once a region is chosen, t^a_j will be increased by 1 and t^b_j will be reset to 1 to indicate that the region has been invested. Finally, once the region has been invested by all the searchers, the SEFS will check all the t^b_j again. By $t^b_j > 1$, it means that the region is not selected by any searcher. In this case, t^a_j will be reset to 1. Moreover, r^{best}_j will also be updated by the sample m_{jk}, if $f(m_{jk}) < f(r^{\text{best}}_j)$.

IV. The Experimental Results

A. Environment and Parameter Settings

The experiments are carried out on a PC with Intel Core i7-9700 3.0 GHz CPUs (with 8 cores and 12 MB of cache) and 16 GB of memory running Ubuntu 18.04. All the programs are written in C++ and compiled using g++. To evaluate the performance of the proposed algorithm (SEFS), it is compared with other six state-of-the-art algorithms for solving the FSP. These algorithms are the common GA [9], binary particle swarm algorithm (BPSO) [19], binary gravitational search algorithm (BGSA) [20], binary bat algorithm (BBA) [21], BGWO [13], and binary grasshopper optimisation algorithm (BGOA) [14]. As shown in Table I, the parameter settings are based on the study described in [14]. As for the proposed

TABLE I: PARAMETERS FOR THE ALGORITHMS

Algorithm	Parameter	Value
BPSO	c_1	2
	c_2	2
	W	0.1
BGSA	G_0	1
	α	20
BBA	$F_{\min}$	0
	$F_{\max}$	2
	A	0.5
	r	0.5
BGWO	a	Decrease linearly from 2 to 0
BGOA	c	$[0, 2.079]$
	l	1.5
	f	0.5
GA	crossover rate	0.8
	mutation rate	0.02
SEFS	# of searchers	2
	# of regions	8
	# of samples	16
	# of players	6
	mutation rate	0.01
	Z_c	Increase linearly from 0.25 to 0.75

TABLE II: LIST OF THE UCI DATASETS.

No.	Dataset	# of features	# of instances
1	Breastcancer	9	699
2	BreastEW	30	569
3	Exactly	13	1,000
4	Exactly2	13	1,000
5	HeartEW	13	270
6	Lymphography	18	148
7	M-of-n	13	1,000
8	PenglungEW	325	73
9	SonarEW	60	208
10	SpectEW	22	267
11	CongressEW	16	435
12	IonosphereEW	34	351
13	KrvskpEW	36	3,196
14	Tic-tac-toe	9	958
15	Vote	16	300
16	WaveformEW	40	5,000
17	WineEW	13	178
18	Zoo	16	101

algorithm, the number of searchers is set equal to 2; the number of regions to 8; the number of samples to 16; the number of players for tournament selection to 6; and the mutation rate to 0.01. In addition, Z_c will increase linearly from 0.25 to 0.75.

To make the experiments as fair as possible, all the algorithms use the same initial setting of the solution and perform 10,000 evaluations so that all the algorithms check the same number of candidate solutions during the convergence process. Besides, the population size is set equal to 8 for all the algorithms each of which are carried out for 30 runs for each dataset. The 18 datasets that are commonly used to verify the performance of a feature selection algorithm, such as [13], [14], are as shown in Table II. In this study, each dataset is divided into two parts so that 80% of which are for training while 20% of which are for testing, to avoid overfitting during classification.

TABLE III: THE AVERAGE OBJECTIVE VALUE FOR EACH DATASET.

No.	BPSO	BGSA	BBA	BGWO	BGOA	GA	SEFS
1	**0.0336**	**0.0336**	**0.0336**	0.0365	**0.0336**	0.0356	0.0337
2	0.0260	0.0267	0.0260	0.0262	0.0261	0.0276	**0.0214**
3	0.0049	0.0226	0.0064	0.1722	**0.0046**	0.1433	**0.0046**
4	0.2386	0.2396	0.2390	0.2403	**0.2381**	0.2403	0.2403
5	0.1483	0.1514	0.1504	0.1610	**0.1476**	0.1587	**0.1476**
6	0.1454	0.1538	0.1516	0.1678	0.1426	0.1607	**0.1422**
7	0.0048	0.0065	0.0055	0.0686	**0.0046**	**0.0046**	**0.0046**
8	0.1103	0.1151	0.1081	0.0760	0.1114	0.1155	**0.0626**
9	0.1073	0.1073	0.1039	0.0937	0.1043	0.0949	**0.0539**
10	0.1680	0.1736	0.1702	0.1880	0.1628	0.1953	**0.1540**
11	0.0361	0.0371	0.0345	0.0418	0.0355	0.0386	**0.0327**
12	0.0995	0.0977	0.0881	0.0822	0.0985	0.0834	**0.0646**
13	0.0414	0.0489	0.0549	0.0728	0.0337	0.0279	**0.0215**
14	**0.2081**	0.2082	**0.2081**	0.2396	**0.2081**	0.2268	0.2082
15	0.0462	0.0478	0.0433	0.0460	0.0459	0.0447	**0.0421**
16	0.1840	0.1874	0.1901	0.1867	0.1798	0.1676	**0.1581**
17	0.0166	0.0192	0.0189	0.0252	0.0166	0.0234	**0.0165**
18	0.0278	0.0338	0.0326	0.0492	0.0260	0.0437	**0.0255**

TABLE IV: THE AVERAGE CLASSIFICATION ACCURACY FOR EACH DATASET.

No.	BPSO	BGSA	BBA	BGWO	BGOA	GA	SEFS
1	**0.9728**	**0.9728**	**0.9728**	0.9680	**0.9728**	0.9703	0.9725
2	0.9790	0.9782	0.9779	0.9766	0.9797	0.9768	**0.9828**
3	0.9999	0.9826	0.9986	0.8295	**1.0000**	0.8592	**1.0000**
4	0.7620	0.7599	0.7610	0.7580	**0.7631**	0.7580	0.7580
5	0.8547	0.8516	0.8523	0.8410	**0.8556**	0.8442	**0.8556**
6	0.8597	0.8502	0.8511	0.8340	**0.8626**	0.8432	0.8621
7	**1.0000**	0.9987	0.9996	0.9351	**1.0000**	**1.0000**	**1.0000**
8	0.8941	0.8886	0.8945	0.9237	0.8936	0.8872	**0.9397**
9	0.8970	0.8965	0.8989	0.9074	0.9003	0.9085	**0.9490**
10	0.8353	0.8298	0.8325	0.8126	0.8412	0.8050	**0.8487**
11	0.9675	0.9664	0.9682	0.9603	0.9683	0.9641	**0.9701**
12	0.9034	0.9049	0.9137	0.9183	0.9048	0.9181	**0.9368**
13	0.9642	0.9561	0.9496	0.9298	0.9723	0.9772	**0.9833**
14	0.7954	0.7956	0.7954	0.7637	0.7955	0.7780	**0.7969**
15	0.9566	0.9548	0.9588	0.9552	0.9569	0.9568	**0.9600**
16	0.8199	0.8163	0.8125	0.8142	0.8248	0.8353	**0.8446**
17	**0.9888**	0.9860	0.9860	0.9792	**0.9888**	0.9816	**0.9888**
18	0.9776	0.9713	0.9719	0.9541	**0.9795**	0.9607	**0.9795**

TABLE V: THE AVERAGE NUMBER OF SELECTED FEATURES FOR EACH DATASET.

No.	BPSO	BGSA	BBA	BGWO	BGOA	GA	SEFS
1	6.00	6.00	6.00	5.97	6.00	**5.63**	5.90
2	15.57	15.43	**12.27**	15.27	17.90	13.87	12.93
3	6.30	7.03	6.53	8.07	6.00	**5.03**	6.00
4	3.83	2.47	3.17	1.87	4.67	**1.00**	**1.00**
5	5.77	5.87	**5.50**	6.97	6.00	5.80	6.00
6	11.60	9.97	**7.60**	9.80	11.77	9.90	10.30
7	6.20	6.70	6.63	7.70	**6.00**	**6.00**	**6.00**
8	174.97	155.37	118.97	155.77	198.53	126.43	**93.63**
9	31.47	29.00	22.63	28.87	33.60	25.83	**20.41**
10	11.00	11.27	9.50	11.63	12.33	**4.93**	9.30
11	6.33	6.00	**4.90**	6.63	6.50	4.93	5.13
12	13.07	12.23	8.83	13.03	14.63	8.10	**6.80**
13	21.73	19.67	18.00	19.87	22.63	18.97	**17.80**
14	**5.00**	5.27	**5.00**	6.47	5.07	6.30	6.40
15	5.07	4.80	3.97	6.03	5.10	**3.13**	4.00
16	22.80	21.80	18.00	22.73	25.33	18.47	**17.01**
17	7.13	6.87	**6.47**	7.03	7.07	6.80	7.00
18	8.97	8.60	7.70	9.10	9.23	**7.67**	8.37

B. Simulation Results

Table III gives the simulation results of 18 datasets for feature selection in terms of the average objective value. The results show that the proposed algorithm (SEFS) outperforms all the other algorithms for most datasets. This implies that the results of the proposed algorithm are worse than all the other algorithms compared in this study for datasets with a smaller number of features, such as datasets 1, 4, and 14. On the contrary, it can be easily seen that the proposed algorithm outperforms all the other algorithms compared in this study for datasets with a larger number of features, such as datasets 8 and 9. Table IV shows the simulation results in terms of the accuracy. It can be easily seen from datasets 17, and 18 that both SEFS and BGOA give the same accuracy, but SEFS outperforms BGOA in terms of the objective value. It can be easily seen from Table V that SEFS can use a fewer number of features to obtain the same accuracy. This implies that SEFS is able to select features that are more important, thus giving a better search capability.

V. Conclusions

In this study, a search economics based algorithm, named search economics for feature selection (SEFS), was proposed for the feature selection problem. The proposed algorithm takes a different approach for dividing the solution space into regions, thus making the regions more influential. It also modifies the crossover scheme to make it more suitable for the newly defined regions. Moreover, the way to measure the expected value for the feature selection problem has also been changed. The experimental results show that the proposed method beats GA, BPSO, BGSA, BBA, BGWO, and BGOA for most datasets, especially for high-dimensional and large-scale datasets. In the future, one of our goals is to improve SEFS for datasets with a small number of features. The other is to apply SEFS to large-scale feature selection problems.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions on the paper. This work was supported in part by the Ministry of Science and Technology of Taiwan, R.O.C., under Contracts MOST107-2221-E-110-021 and MOST108-2221-E-110-028.

References

[1] H. Liu and H. Motoda, *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media, 1998.

[2] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[3] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Proceedings of Neural Networks for Signal Processing Society Workshop*, 1999, pp. 41–48.

[4] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of National Conference on Artificial Intelligence*, vol. 2, 1992, pp. 129–134.

[5] J. Tang and H. Liu, "Unsupervised feature selection for linked social media data," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 904–912.

[6] H. Xu, Y. Fu, C. Fang, Q. Cao, J. Su, and S. Wei, "An improved binary whale optimization algorithm for feature selection of network intrusion detection," in *Proceedings of International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, 2018, pp. 10–15.

[7] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[8] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[9] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *Intelligent Systems*, vol. 13, no. 2, pp. 44–49, 1998.

[10] A. Al-Ani, "Feature subset selection using ant colony optimization," *International Journal of Computational Intelligence*, vol. 2, no. 1, pp. 53–58, 2005.

[11] C. W. Tsai, "Search economics: A solution space and computing resource aware search method," in *Proceedings of International Conference on Systems, Man, and Cybernetics*, 2015, pp. 2555–2560.

[12] C. W. Tsai and S. J. Liu, "An effective IoT service-to-interface assignment algorithm via search economics," *Internet of Things Journal*, vol. 5, no. 3, pp. 1708–1718, 2018.

[13] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[14] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. Z. Ala'M, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Systems with Applications*, vol. 117, pp. 267–286, 2019.

[15] H. Chen, W. Jiang, C. Li, and R. Li, "A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–6, 2013.

[16] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassenian, "Feature subset selection approach by gray-wolf optimization," in *Proceedings of Afro-European Conference for Industrial Advancement*, vol. 334, 2015, pp. 1–13.

[17] A. G. Hussien, E. H. Houssein, and A. E. Hassanien, "A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection," in *Proceedings of International Conference on Intelligent Computing and Information Systems*, 2017, pp. 166–172.

[18] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[19] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of International Conference on Systems, Man, and Cybernetics.*, vol. 5, 1997, pp. 4104–4108.

[20] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, 2010.

[21] R. Y. Nakamura, L. A. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X. S. Yang, "BBA: A binary bat algorithm for feature selection," in *Proceedings of Conference on Graphics, Patterns and Images*, 2012, pp. 291–297.