

Research Article

Grey Wolf Optimizer Based on Powell Local Optimization Method for Clustering Analysis

Sen Zhang¹ and Yongquan Zhou^{1,2}

¹College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

²Guangxi High School Key Laboratory of Complex System and Computational Intelligence, Nanning 530006, China

Correspondence should be addressed to Yongquan Zhou; yongquanzhou@126.com

Received 25 July 2015; Accepted 11 October 2015

Academic Editor: Lawrence P. Horwitz

Copyright © 2015 S. Zhang and Y. Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One heuristic evolutionary algorithm recently proposed is the grey wolf optimizer (GWO), inspired by the leadership hierarchy and hunting mechanism of grey wolves in nature. This paper presents an extended GWO algorithm based on Powell local optimization method, and we call it PGWO. PGWO algorithm significantly improves the original GWO in solving complex optimization problems. Clustering is a popular data analysis and data mining technique. Hence, the PGWO could be applied in solving clustering problems. In this study, first the PGWO algorithm is tested on seven benchmark functions. Second, the PGWO algorithm is used for data clustering on nine data sets. Compared to other state-of-the-art evolutionary algorithms, the results of benchmark and data clustering demonstrate the superior performance of PGWO algorithm.

1. Introduction

Swarm Intelligence (SI) has received much attention. Many SI algorithms have been proposed, such as genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], Differential Evolution (DE) [3, 4], and evolutionary programming (EP) [5]. Although these algorithms are capable of solving complex optimization problems, it has been proven by the well-known No Free Lunch (NFL) theorem that there is no optimization solving all optimization problems [6]. So the theorem allows researchers to propose some new algorithms. Some of the recent algorithms are grey wolf optimizer (GWO) [7], Multiverse Optimizer (MVO) [8], Ant Lion Optimizer (ALO) [9], artificial bee colony (ABC) algorithm [10], Firefly algorithm (FA) [11], Cuckoo search (CS) algorithm [12], Cuckoo Optimization Algorithm (COA) [13], gravitational search algorithm (GSA) [14], and Adaptive Gbest-guided Gravitational Search Algorithm (GGSA) [15].

In this paper, we concentrate on GWO, developed by Mirjalili et al. [7] in 2014 based on simulating hunting behavior and social leadership of grey wolves in nature. Numerical comparisons showed that the superior performance of GWO is competitive to that of other population-based algorithms.

Because it is simple and easy to implement and has fewer control parameters, GWO has caused much attention and has been used to solve a number of practical optimization problems [16–18].

However, like other stochastic optimization algorithms, such as PSO and GA, as the growth of the search space dimension, GWO algorithm provides a poor convergence behavior at exploitation [19, 20]. Therefore, it is necessary to emphasize that our work falls in increasing the local search ability of GWO algorithm. According to [21], we can know that many direct search methods are fairly fast optimizers and have a strong ability of local search. Powell's method [22] is one of the direct search methods. In order to use good convergence property of Powell's method, we propose a hybrid metaheuristic that is grey wolf optimizer based on Powell local optimization method (PGWO). Compared to other state-of-the-art evolutionary algorithms, PGWO performs significantly better.

Cluster analysis or clustering is the task of grouping similar objects or multidimensional data vectors into a number of clusters or groups. Clustering analysis is an unsupervised learning process, without a priori knowledge of clustering, and the clustering algorithm is automatic classification based

on the distance or similarity between samples. As a result of this characteristic, clustering techniques have been applied to a wide range of problems, such as data mining [23], data analysis, pattern recognition [24], and image segmentation [25]. The traditional clustering methods can be simply classified as partitioning methods, hierarchical methods, density-based methods, and grid-based methods [26]. In this paper, we concentrate on partitioning methods. Partitioning clustering method divides data vectors into a predefined number of clusters through optimizing some certain criterion. K -means is the most popular partitioning clustering method [27, 28]. However, K -means highly depends on initial states and always falls into local optimum. In order to overcome this problem, many methods have been proposed. For example, K -harmonic mean algorithm has been proposed for clustering instead of K -means in [29]. A simulated annealing- (SA-) based approach has been developed in [30]. A tabu search- (TS-) based method was introduced in [31, 32]. Genetic algorithm- (GA-) based methods were presented in [33–36]. Fathian et al. [37] have developed a clustering algorithm based on honey-bee mating optimization (HBMO). The particle swarm optimization (PSO) is applied for clustering in [38]. Hatamlou et al. employed a big bang-big crunch algorithm for data clustering in [39]. Karaboga and Ozturk presented a novel clustering approach based on artificial bee colony (ABC) algorithm in [40]. Data clustering based on gravitational search algorithm was presented in [41, 42]. In 1991, Colorni et al. have presented ant colony optimization (ACO) algorithm based on the behavior of ants seeking a path between their colony and a source of food. Then, Shelokar and Kao solved the clustering problem using the ACO algorithm [43, 44]. Kao et al. have presented a hybrid approach according to combination of the K -means algorithm, Nelder-Mead simplex search, and PSO for clustering analysis [45]. Niknam et al. have presented a hybrid evolutionary algorithm based on PSO and SA to solve the clustering problem [46]. But every algorithm has some drawbacks; for example, K -means algorithm sucks in local optima, and convergence is highly dependent on initial positions in case of genetic algorithm; in ACO, the solution vector has been affected as the number of iterations increased; Kao et al. [45] and Niknam et al. [46] stated that PSO gives better clustering results, when it is applied in one-dimensional data set and for small data set, but when it is applied to large data set, it does not give the good results, and so forth. In this paper, a PGWO algorithm is used to solve the cluster problem, tested on nine data sets. This algorithm takes the advantage of GWO and Powell method. Initial process is started by GWO, which allows searching all the space for a global solution. When global solution is found, the clustering is switched to Powell for faster convergence to finish the clustering process. As can be seen from the simulation results, this proposed algorithm not only has higher convergence speed but also can find out the optimal solution compared to the other algorithms across the majority of data sets whether in small data set or in large data set.

The rest of the paper is organized as follows: Section 2 presents a brief introduction to GWO. Section 3 discusses the basic principles of GWO based on Powell local optimization

method. The experimental results of test functions and data clustering are shown in Sections 4 and 5, respectively. Finally, Section 6 concludes the work and suggests some directions for future studies.

2. Grey Wolf Optimizer (GWO)

The GWO algorithm, proposed by Mirjalili et al. (2014) [7], is inspired by the hunting behavior and social leadership of grey wolves in nature. It is similar to other metaheuristics, and in GWO algorithm the search begins by a population of randomly generated wolves (candidate solutions). In order to formulate the social hierarchy of wolves when designing GWO, in this algorithm the population is split into four groups: alpha (α), beta (β), delta (δ), and omega (ω). Over the course of iterations, the first three best solutions are called α , β , and δ , respectively. The rest of the candidate solutions are named as ω . In this algorithm, the hunting (optimization) is guided by α , β , and δ . The ω wolves are required to encircle α , β , and δ so as to find better solutions. The encircle process could be molded as follows [7]:

$$\begin{aligned}\vec{D} &= |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \\ \vec{X}(t+1) &= \vec{X}_p(t) - \vec{A} \cdot \vec{D},\end{aligned}\quad (1)$$

where t indicates the current iteration, $\vec{C} = 2 \cdot \vec{r}_2$, $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$, \vec{X}_p is the position vector of the prey, \vec{X} indicates the position vector of a grey wolf, \vec{a} is gradually decreased from 2 to 0, and r_1 and r_2 are random numbers over range $[0, 1]$.

In order to mathematically simulate the hunting behavior of grey wolves, in the GWO algorithm we always assume that α , β , and δ have better knowledge about the position of the prey (optimum). Therefore, the positions of the first three best solutions (α , β , δ) obtained so far are saved and other wolves (ω) are obliged to reposition with respect to α , β , and δ . The mathematical model of readjusting the positions of ω wolves is presented as follows [7]:

$$\begin{aligned}\vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|,\end{aligned}\quad (2)$$

$$\begin{aligned}\vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|, \\ \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta),\end{aligned}\quad (3)$$

$$\begin{aligned}\vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta), \\ \vec{X}(t+1) &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3},\end{aligned}\quad (4)$$

where \vec{X}_α is the position of the alpha, \vec{X}_β is the position of the beta, \vec{X}_δ is the position of the delta, \vec{C}_1 , \vec{C}_2 , and \vec{C}_3 and \vec{A}_1 , \vec{A}_2 , and \vec{A}_3 are all random vectors, \vec{X} is the position of the current solution, and t indicates the number of iterations.

```

Initialize the grey wolf population  $X_i = (i = 1, 2, \dots, n)$  and parameters
Calculate the fitness of population
Find the first three agents  $X_\alpha, X_\beta, X_\delta$ 
While ( $t < \text{Max number of iterations}$ )
    Update the position of the current search agent by (4)
    Calculate the fitness of population
    Update  $X_\alpha, X_\beta, X_\delta$ 
     $t = t + 1$ 
End while
Return  $X_\alpha$ 

```

ALGORITHM 1: GWO algorithm.

```

Initialize the starting point  $X_1$ , independent vectors  $d_i = e_i (i = 1, \dots, D)$ , the tolerance for stop criteria  $\epsilon$ , set
 $f(1) = f(X_1), X_c = X_1, K = 1$ 
While (stopping criterion is not met, namely  $|\Delta f| > \epsilon$ ) do
    For  $i = 1$  to  $D$  do
        If ( $K \geq 2$ ) then
             $d_i = d_{i+1}$ 
        End If
         $X_{i+1} = X_i + \lambda_i d_i$ ,  $\lambda_i$  is determined by Minimizing  $f(X_{i+1})$ 
    End For
     $d_{i+1} = \sum_1^D \lambda_i * d_i = X_{D+1} - X_D$ ,     $X_c(k+1) = X_{D+1} + \lambda_k d_{i+1}$ ,     $f(k+1) = f(X_c(k+1))$ 
     $k = k + 1$ ,     $X_1 = X_c(k)$ ,     $\Delta f = f(k) - f(k-1)$ 
End While

```

ALGORITHM 2: Powell's method.

In these formulas, it may also be observed that there are two vectors \vec{A} and \vec{C} obliging the GWO algorithm to explore and exploit the search space. With decreasing A , half of the iterations are devoted to exploration ($|A| \geq 1$) and the other half are dedicated to exploitation ($|A| < 1$). The range of C is $2 \leq C \leq 0$, and the vector C also improves exploration when $C > 1$ and the exploitation is emphasized when $C < 1$. Note here that A is decreased linearly over the course of the iterations. In contrast, C is generated randomly whose aim is to emphasize exploration/exploitation at any stage avoiding local optima. The main steps of grey wolf optimizer are given in Algorithm 1.

3. Grey Wolf Optimizer Based on Powell Local Optimization Method (PGWO)

3.1. Local Search Based on Powell's Method. Powell's method, strictly Powell's conjugate direction method, is an algorithm proposed by Powell [22] for finding a local minimum of a function. The function need not be differentiable, and no derivatives are taken. The method successively utilized a bidirectional search approach along each search vector to pursue the minimum of a function. The new position is represented as a linear combination of the search vectors. The new displacement vector as a new search vector is added to the search vector list. Correspondingly, the most successful

vector which contributed most to the new direction is deleted from the search vector list. The algorithm is iterated by some run times until no significant improvement is made. The detailed steps of Powell's method procedure are presented in Algorithm 2 [47].

3.2. The Proposed Algorithm. Over the last years, many research results have been published in the field of evolutionary algorithms [7]. And the results show that the hybridizations metaheuristics and local search techniques are exceptionally successful. Successful hybridizations have been proposed, especially in combinatorial and discrete solution spaces [48, 49]. Interestingly, for real-valued solution spaces, few results have been introduced yet—an astonishing fact as many direct search methods are fairly fast optimizers [21]. Therefore, in order to overcome the shortcomings of GWO algorithm including slow convergence speed, easily falling into local optimum value, low computation accuracy, and low success rate of convergence [19, 20], GWO algorithm based on Powell local optimization method is proposed. It adopts the powerful local optimization ability of Powell's method and embeds it into GWO as a local search operator. In this case, the proposed method has potential to provide superior results compared to other state-of-the-art evolutionary algorithms. The general steps of PGWO are presented as follows, among which p indicates the performing probability

Initialize the grey wolf population $X_i = (i = 1, 2, \dots, n)$ and parameters
 Calculate the fitness of population
 Find the first three agents $X_\alpha, X_\beta, X_\delta$
While ($t < \text{Max number of iterations}$)
 Update the position of the current search agent by (4)
 If $\text{rand} > p$, choose current best solution X_α as a starting point and generate a new solution X'_α by Powell's method as illustrated Algorithm 2. If $X'_\alpha < X_\alpha$, replace X_α with X'_α , otherwise, go to Step 6
 Calculate the fitness of population
 Update $X_\alpha, X_\beta, X_\delta$
 $t = t + 1$
End While
 Return X_α

ALGORITHM 3: PGWO algorithm.

TABLE 1: Benchmark functions.

Functions	Name	Dim	Range	f_{\min}
$F_1 = \sum_{i=1}^n x_i^2$	Sphere	30	[-100, 100]	0
$F_2 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Rosenbrock	30	[-30, 30]	0
$F_3 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	Ackley	30	[-32, 32]	0
$F_4 = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	Penalized	30	[-50, 50]	0
$F_5 = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	Penalized 2	30	[-50, 50]	0
$F_6 = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	Kowalik	4	[-5, 5]	0.0003075
$F_7 = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	Goldstein-Price	2	[-2, 2]	3

of Powell's method and is set to 0.5. We also have tried to vary the number of grey wolves (population size n) and the performing probability of Powell's method. From our simulations, we found that $n = 30$ and $p = 0.5$ are sufficient for most optimization problems. So we will use fixed $n = 30$ and $p = 0.5$ in the rest of the simulations. In the following section, various benchmark functions are employed to investigate the efficiency of PGWO algorithm (see Algorithm 3).

4. Experimental Results and Discussion

4.1. Simulation Platform. All algorithms are tested in Matlab R2012a (7.14) and experiments are executed on AMD Athlon

(tm) II X4 640 Processor 3.00 GHz PC with 3 G RAM. Operating system is Windows 7.

4.2. Benchmark Functions. In order to evaluate the property of PGWO algorithm, seven standard benchmark functions are employed [50, 51]. These benchmark functions can be divided into two different groups: unimodal and multimodal functions. Table 1 lists the functions, where Dim indicates dimension of the function, Range is the boundary of the function's search space, and f_{\min} is the minimum return value of the function, among which the former two functions (F_1 - F_2) are unimodal, (F_3 - F_5) are multimodal functions, and the last two functions (F_6 - F_7) are fixed-dimension multimodal

TABLE 2: Initial parameters of algorithms.

Algorithm	Parameter	Value
PGWO	Population size	30
	$\vec{\alpha}$	Linearly decreased from 2 to 0
	Performing probability of Powell (P)	0.5
	Max iteration	500 (benchmark functions test), 200 (data sets test)
	Stopping criteria	Max iteration
GWO	Population size	30
	$\vec{\alpha}$	Linearly decreased from 2 to 0
	Max iteration	500 (functions test), 200 (data sets test)
	Stopping criteria	Max iteration
PSO	Population size	30
	C_1, C_2	1.4962, 1.4962
	W	0.7298
	Max iteration	500 (benchmark functions test), 200 (data sets test)
	Stopping criteria	Max iteration
ABC	Population size	30
	Limit	10
	Max iteration	500 (benchmark functions test), 200 (data sets test)
	Stopping criteria	Max iteration
CS	Population size	30
	P_a	0.25
	Max iteration	500 (benchmark functions test), 200 (data sets test)
	Stopping criteria	Max iteration
GGSA	Population size	30
	c_1'	$(-2t^3/T^3) + 2$
	c_2'	$(2t^3/T^3)$
	G_0	1
	α	20
	Max iteration	500 (benchmark functions test), 200 (data sets test)
	Stopping criteria	Max iteration

T indicates the maximum number of interactions; t is the current iteration.

benchmark functions. There are some parameters that should be initialized before running. Table 2 is the initial value of these algorithms.

The experimental results are presented in Table 3. The results are averaged over 20 independent runs, and bold results mean that PGWO is better, while underlined results mean that the other algorithm is better. The Best, Worst, Mean, and Std. represent the optimal fitness value, worst fitness value, mean fitness value, and standard deviation, respectively. Note that the Matlab code of the GGSA algorithm is given in <http://www.alimirjalili.com/Projects.html>.

To improve the performance evaluation of evolutionary algorithms, statistical tests should be conducted [52]. In order to determine whether the results of PGWO differ from the best results of the other algorithms in a statistical method, a nonparametric test which is known as Wilcoxon's rank-sum test [53, 54] is performed at 5% significance level. The p values calculated in Wilcoxon's rank-sum test comparing PGWO and the other algorithms over all the benchmark functions are given in Table 4. According to [52], p values < 0.05 can be considered as sufficient evidence against the null hypothesis.

4.3. Comparison of Experiment Results. As shown in Table 3, PGWO has the best result for F_1 . For F_2 , PGWO provided better results than the other algorithms across Best, Worst, and Mean, while the Std. of GWO is better than PGWO. The unimodal benchmark functions have only one global solution without any local optima so they are very suitable to examine exploitation. Hence, these results indicate that the proposed method provides greatly improved exploitation compared to the other algorithms.

However, it should be noticed that multimodal benchmark functions have many local minima. The final results are more important because these functions can reflect the ability of the algorithm escaping from poor local optima and obtaining the global optimum. We have tested the experiments on F_3 - F_5 . Seen from Table 3, the PGWO provides the best results.

For F_6 - F_7 , these are fixed-dimension multimodal benchmark functions with only a few local minima; the dimensions of the functions are also small. In this case, it is hard to judge the property of the algorithms. The major difference compared with functions F_3 - F_5 is that functions F_6 - F_7 appear

TABLE 3: Results comparison of different optimal algorithms for 20 independent runs.

Function	Criteria	GGSA	CS	ABC	PSO	GWO	PGWO
F_1	Best	1.0133E - 17	3.7450E + 00	2.1820E + 01	1.7964E - 06	4.7447E - 29	2.5422E - 54
	Worst	1.0181E - 16	2.4776E + 01	1.8094E + 02	2.2907E - 04	2.2227E - 26	7.9888E - 53
	Mean	5.1858E - 17	8.8292E + 00	8.1908E + 01	6.5181E - 05	1.8269E - 27	1.6276E - 53
	Std.	2.6953E - 17	4.7633E + 00	4.3834E + 01	6.9891E - 05	4.8884E - 27	1.7834E - 53
F_2	Best	2.3974E + 01	3.0920E + 02	4.4296E + 03	2.4432E + 01	2.6182E + 01	4.9374E + 00
	Worst	3.4411E + 02	2.2219E + 03	4.5866E + 04	4.1217E + 02	2.8769E + 01	1.5466E + 01
	Mean	7.7481E + 01	5.5445E + 02	1.5193E + 04	1.2465E + 02	2.7507E + 01	1.1692E + 01
	Std.	7.4751E + 01	4.1835E + 02	9.7239E + 03	1.4004E + 02	<u>6.7488E - 01</u>	2.0844E + 00
F_3	Best	2.3591E - 09	4.0632E + 00	9.2131E + 00	6.3889E - 03	7.9048E - 14	4.4409E - 15
	Worst	9.8704E - 09	1.0603E + 01	1.6954E + 01	4.9535E + 00	1.5010E - 13	4.4409E - 15
	Mean	5.0646E - 09	6.0677E + 00	1.3090E + 01	2.0423E + 00	1.0498E - 13	4.4409E - 15
	Std.	1.9095E - 09	1.5421E + 00	2.2882E + 00	1.1464E + 00	1.7485E - 14	0
F_4	Best	5.4733E - 01	2.0863E + 00	7.5357E - 01	2.4473E - 03	1.8626E - 02	1.6432E - 32
	Worst	4.6556E + 00	4.1099E + 00	5.5219E + 00	3.6426E + 00	1.2149E - 01	1.6432E - 32
	Mean	1.9728E + 00	3.2711E + 00	2.0544E + 00	8.5739E - 01	5.4300E - 02	1.6432E - 32
	Std.	1.0488E + 00	5.9739E - 01	1.3358E + 00	1.0326E + 00	2.9934E - 02	2.8080E - 48
F_5	Best	2.1024E - 02	2.7015E + 00	2.2507E + 00	6.3080E - 06	3.9444E - 01	1.3498E - 32
	Worst	2.4387E + 01	1.2516E + 01	1.1090E + 02	3.6461E + 00	8.5465E - 01	1.9625E - 01
	Mean	7.7881E + 00	6.2644E + 00	1.1124E + 01	3.7285E - 01	6.7323E - 01	4.5918E - 02
	Std.	6.6733E + 00	2.5624E + 00	2.3792E + 01	9.3480E - 01	1.4330E - 01	6.0835E - 02
F_6	Best	1.2358E - 03	3.1550E - 04	8.3768E - 04	<u>3.0749E - 04</u>	3.0750E - 04	3.0752E - 04
	Worst	1.2219E - 02	<u>6.0087E - 04</u>	3.4810E - 03	1.9361E - 03	2.0363E - 02	1.5943E - 03
	Mean	4.3558E - 03	<u>4.3266E - 04</u>	1.6421E - 03	9.3309E - 04	6.4067E - 03	4.5499E - 04
	Std.	2.6046E - 03	<u>8.5397E - 05</u>	7.2141E - 04	4.1733E - 04	9.3762E - 03	3.4265E - 04
F_7	Best	3.0000E + 00	3.0000E + 00	3.0000E + 00	3.0000E + 00	3.0000E + 00	3.0000E + 00
	Worst	3.0000E + 00	3.0000E + 00	3.0620E + 00	3.0000E + 00	8.4000E + 01	3.0000E + 00
	Mean	3.0000E + 00	3.0000E + 00	3.0136E + 00	3.0000E + 00	1.1100E + 01	3.0000E + 00
	Std.	2.5794E - 15	1.5717E - 15	1.9505E - 02	<u>1.2310E - 15</u>	2.4931E + 01	9.2458E - 07

TABLE 4: p values produced by Wilcoxon's rank-sum test comparing PGWO versus GWO, PSO, ABC, CS, and GGSA over benchmark functions ($p \geq 0.05$ have been underlined).

PGWO vs	GWO	PSO	ABC	CS	GGSA
F_1	6.80E - 08	6.80E - 08	6.80E - 08	6.80E - 08	6.80E - 08
F_2	6.80E - 08	6.80E - 08	6.80E - 08	6.80E - 08	6.80E - 08
F_3	7.66E - 09	8.01E - 09	8.01E - 09	8.01E - 09	8.01E - 09
F_4	8.01E - 09	8.01E - 09	8.01E - 09	8.01E - 09	8.01E - 09
F_5	5.97E - 08	<u>0.295606</u>	5.97E - 08	5.97E - 08	4.67E - 07
F_6	0.003057	0.000375	1.38E - 06	0.001782	1.06E - 07
F_7	0.0002	4.93E - 08	6.80E - 08	6.19E - 08	6.49E - 08

to be simpler than F_3 - F_5 due to their low dimension and a smaller number of local minima. For F_6 , CS provided better results than the other algorithms across Mean, Std., and Worst. For F_7 , the majority of the algorithms can find the optimal solution, but the PSO is more stable than the other algorithms in terms of Std.

The p values of Wilcoxon's rank-sum in Table 4 show that the results of PGWO in F_5 are not significantly better than PSO. However, the PGWO achieves significant improvement in all remaining benchmark functions compared to the other algorithms. Therefore, this is evidence that the proposed

algorithm has high performance in dealing with unimodal, multimodal, and fixed-dimension multimodal benchmark functions.

The convergence curves of six algorithms are illustrated in Figure 1. As can be seen in these figures, PGWO has a much better convergence rate than the other algorithms on all benchmark functions.

According to this comprehensive comparative study and discussions, these results show that the proposed algorithm is able to significantly improve the performance of GWO and overcome its major shortcomings. For that reason, in the

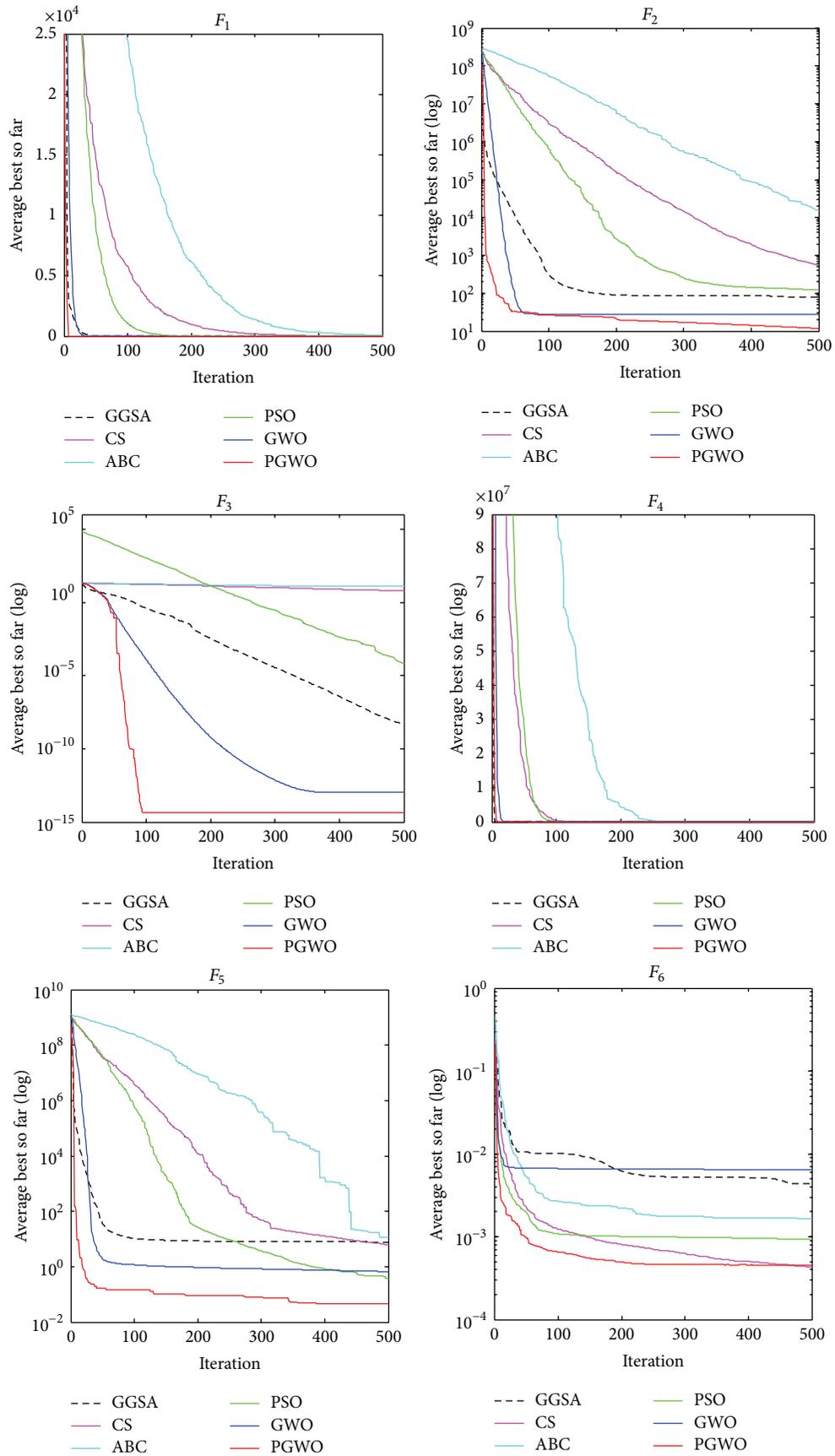


FIGURE 1: Continued.

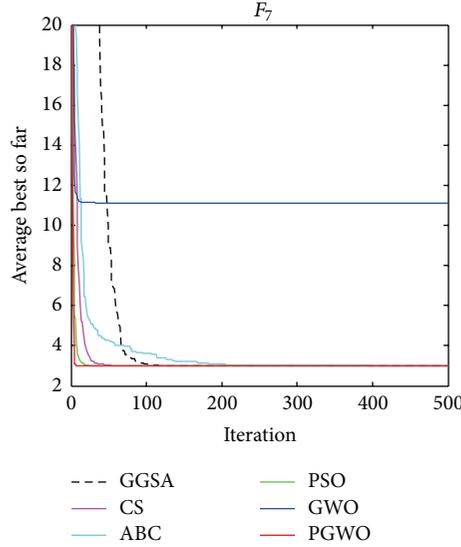


FIGURE 1: The convergence curves of the average fitness value over 20 independent runs.

next section, we apply PGWO algorithm to solve clustering problem.

5. Data Clustering

5.1. The Mathematical Description of Data Clustering. Clustering sample is set to $X = \{X_i, i = 1, 2, \dots, n\}$, where X_i is a p dimensional vector. Clustering problem is to find a partition $C = \{C_1, C_2, \dots, C_m\}$, satisfying the following conditions [26]:

- (1) $X = \bigcup_{i=1}^m C_i$.
- (2) $C_i \neq \phi, i = 1, 2, \dots, m$.
- (3) $C_i \cap C_j = \phi, i, j = 1, 2, \dots, m, i \neq j$.

5.2. The Clustering Criterion. Clustering is the process of grouping a set of data objects into a number of clusters or groups. The aim of clustering is to make the data within a cluster have a high similarity while being very dissimilar to objects in other clusters. Dissimilarities and similarities are evaluated based on the attribute of data sets, containing distance metric. The most popular distance metric is Euclidean distance [55]. Let $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ be two objects described by p numeric attributes; the Euclidean distance between object i and j is presented as follows:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}. \quad (5)$$

For given N objects, the clustering problem is to minimize the sum squared Euclidean distance between objects and allocate each object to one of k cluster centers [55]. The clustering aims

at finding clusters center through minimizing the objective function. The objective function is defined as follows [26]:

$$J_c = \sum_{k=1}^m \sum_{X_i \in C_k} d(X_i, Z_k), \quad (6)$$

where m indicates the number of clusters, C_k indicates the k th cluster, Z_k indicates the k th center of the clustering, and $d(X_i, Z_k)$ indicates the distance of the sample to the corresponding center of clustering; namely, $d(X_i, Z_k) = \|X_i - Z_k\|$.

5.3. PGWO Algorithm on Clustering. In clustering analysis, each element in the data set is a p dimensional vector. Moreover, the actual position of a grey wolf represents the k cluster centers, so each grey wolf indicates a $k * p$ dimensional vector. For each grey wolf i , its position is denoted as a vector $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,k*p})$. In the initialization phase, we utilize maximum and minimum values of each component of the data set (which is to be grouped) as PGWO algorithm the initialization search scope of the grey wolves, and the initialization solution is randomly generated in this range. We use (6) to calculate the fitness function of grey wolves' individuals and the main steps of the fitness function are shown in Algorithm 4.

5.4. Data Clustering Experimental Results and Discussion. In order to verify performance of the proposed PGWO approach for clustering, we compare the results of the K-means, GGSA, CS, ABC, PSO, GWO, and PGWO clustering algorithms using nine different data sets that are selected from the UCI machine learning repository [56].

Artificial data set ($N = 600, d = 2$, and $k = 4$) is a two-featured problem with four unique classes. A total of 600

- (1) **For** data vector x_i
- (2) Calculate the Euclidean distance by (5)
- (3) Assign x_i to the closest cluster center
- (4) Calculate the measure function by (6)
- (5) **End For**
- (6) Return value of the fitness function

ALGORITHM 4: Main steps of the fitness function.

patterns were drawn from four independent bivariate normal distributions, where classes were distributed according to

$$N_2 \left(\mu = \begin{pmatrix} m_i \\ 0 \end{pmatrix}, \Sigma = \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.5 \end{bmatrix} \right), \quad (7)$$

where $i = 1, 2, 3, 4$, $m_1 = -3$, $m_2 = 0$, $m_3 = 3$, and $m = 6$. μ and Σ are mean vector and covariance matrix, respectively [45, 57].

Iris data ($N = 150$, $d = 4$, and $K = 3$) is a data set with 150 random samples of flowers from the *Iris* species *setosa*, *versicolor*, and *virginica* collected by Anderson [58]. From each species there are 50 observations for sepal length, sepal width, petal length, and petal width in cm. This data set was used by Fisher [59] in his initiation of the linear-discriminant-function technique [28, 56, 57].

Wisconsin breast cancer ($N = 683$, $d = 9$, and $K = 2$) consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects) [28, 56, 57].

Contraceptive method choice ($N = 1473$, $d = 10$, $K = 3$), CMC for short, is a data set that is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who either were not pregnant or did not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socioeconomic characteristics [28, 56, 57].

Seeds data ($N = 210$, $d = 7$, and $K = 3$) is a data set that consists of 210 patterns belonging to three different varieties of wheat: Kama, Rosa, and Canadian. From each species, there are 70 observations for area A , perimeter P , compactness C ($C = 4 * pi * A / p^2$), length of kernel, width of kernel, asymmetry coefficient, and length of kernel groove [56].

Statlog (Heart) data ($N = 270$, $d = 13$, $K = 2$) is a data set that is a heart disease database similar to a database already present in the repository (heart disease databases) but in a slightly different form [56].

Wine data ($N = 178$, $d = 13$, $K = 3$) is also taken from MCI laboratory. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types

of wines. There are 178 instances with 13 numeric attributes in wine data set. All attributes are continuous. There is no missing attribute value [28, 56, 57].

Balance scale data ($N = 625$, $d = 4$, and $K = 3$) is a data set that was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, or left being balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced [56].

Haberman's Survival ($N = 306$, $d = 3$, and $K = 2$) is a data set that contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. It records two survival status patients with the age of patient at time of operation, patient's year of operation, and number of positive axillary nodes detected [56].

The results of comparison of intracluster distances for the seven clustering algorithms over 20 independent runs are shown in Table 5. Table 6 reports the p values produced by Wilcoxon's rank-sum test comparing PGWO and the other algorithms over all the data sets.

For Art data set, the optimum value, the worst value, and the average value of PGWO and GGSA are all $5.1390E + 02$, while the standard deviation of GGSA is better than PGWO. For PSO, it only gets the optimum solution $5.1390E + 02$.

For Iris data set, Cancer data set, and Seeds data set, PGWO, PSO, and GGSA provide the optimum value in comparison to those obtained by other methods. However, the worst value, the average value, and the standard deviation value of PGWO are superior to those of the other methods.

For Heart data set, PGWO, PSO, and CS all find the optimum solution $1.0623E + 04$. That means they all can find the global solution. However, PGWO is slightly better for the worst value, the average value, and the standard deviation.

For CMC data set and Wine data set, Table 5 shows that the average, best, worst, and standard deviation values of the fitness function for PGWO algorithm are much smaller than those of the other six methods. The PGWO clustering algorithm is capable of providing the same partition of the data points in all runs.

For balance scale data set, the optimum values of the fitness function for PGWO, PSO, and GGSA are $1.4238E + 03$. That means they all can find the global solution. And the optimum values of the fitness function for GWO and K -means are $1.4239E + 03$; the result is close to PGWO, PSO, and GGSA. However, the standard deviation values of GWO, K -means, PGWO, PSO, GGSA, and CS are $6.8756E - 01$, $3.6565E + 00$, $8.6619E - 01$, $9.7396E - 01$, $1.6015E + 00$, and $7.3239E - 01$, respectively. From the standard deviation, we can see that the GWO is more stable than the other methods.

For Haberman's Survival data set, the optimum value, the worst value, and the average value of the fitness function for CS and PSO are almost the same. The results of CS and PSO algorithms are better than those of the other methods.

The p values in Table 6 show that the results of PGWO are significantly better in Art data set, Cancer data set,

TABLE 5: Comparison of intracluster distances for the seven clustering algorithms over 20 independent runs.

Data set	Criteria	<i>K</i> -means	GGSA	CS	ABC	PSO	GWO	PGWO
Art	Best	5.1454E + 02	5.1390E + 02	5.1391E + 02	5.1752E + 02	5.1390E + 02	5.1455E + 02	5.1390E + 02
	Worst	9.0474E + 02	5.1390E + 02	5.3213E + 02	5.5298E + 02	8.9248E + 02	5.2058E + 02	5.1390E + 02
	Mean	6.0974E + 02	5.1390E + 02	5.1597E + 02	5.2620E + 02	5.3283E + 02	5.1687E + 02	5.1390E + 02
	Std.	1.6903E + 02	<u>1.5430E - 13</u>	4.8536E + 00	9.7988E + 00	8.4652E + 01	1.8768E + 00	6.1243E - 08
Iris	Best	9.7190E + 01	9.6655E + 01	9.6659E + 01	9.7616E + 01	9.6655E + 01	9.6835E + 01	9.6655E + 01
	Worst	1.2318E + 02	1.1839E + 02	9.7684E + 01	1.0401E + 02	1.2767E + 02	1.2108E + 02	9.6655E + 01
	Mean	1.0226E + 02	9.8865E + 01	9.6862E + 01	9.9710E + 01	1.0596E + 02	9.9903E + 01	9.6655E + 01
	Std.	1.0290E + 01	5.5454E + 00	3.1103E - 01	1.8460E + 00	1.4578E + 01	6.7482E + 00	8.5869E - 10
Cancer	Best	2.9763E + 03	2.9644E + 03	2.9645E + 03	2.9887E + 03	2.9644E + 03	2.9645E + 03	2.9644E + 03
	Worst	2.9884E + 03	3.0611E + 03	2.9677E + 03	3.2650E + 03	4.7288E + 03	2.9650E + 03	2.9644E + 03
	Mean	2.9830E + 03	2.9715E + 03	2.9651E + 03	3.0760E + 03	3.4055E + 03	2.9647E + 03	2.9644E + 03
	Std.	4.8278E + 00	2.1284E + 01	7.3222E - 01	6.5283E + 01	7.8385E + 02	1.1452E - 01	1.6118E - 08
CMC	Best	5.7016E + 03	5.6938E + 03	5.7054E + 03	5.8219E + 03	5.6942E + 03	5.8236E + 03	5.6937E + 03
	Worst	5.7053E + 03	5.7111E + 03	5.7762E + 03	6.3743E + 03	7.1580E + 03	6.0911E + 03	5.6937E + 03
	Mean	5.7040E + 03	5.6971E + 03	5.7263E + 03	6.0868E + 03	5.7690E + 03	5.9239E + 03	5.6937E + 03
	Std.	1.1022E + 00	3.7872E + 00	2.1119E + 01	1.7292E + 02	3.2694E + 02	9.0635E + 01	9.8777E - 04
Seeds	Best	3.1322E + 02	3.1180E + 02	3.1187E + 02	3.1573E + 02	3.1180E + 02	3.1323E + 02	3.1180E + 02
	Worst	3.1373E + 02	3.1354E + 02	3.1746E + 02	3.4997E + 02	4.2035E + 02	3.1782E + 02	3.1180E + 02
	Mean	3.1337E + 02	3.1189E + 02	3.1347E + 02	3.3161E + 02	3.2813E + 02	3.1475E + 02	3.1180E + 02
	Std.	2.3686E - 01	3.8862E - 01	1.6383E + 00	1.0943E + 01	3.9697E + 01	1.5716E + 00	1.7815E - 09
Heart	Best	1.0682E + 04	1.0749E + 04	1.0623E + 04	1.0683E + 04	1.0623E + 04	1.0637E + 04	1.0623E + 04
	Worst	1.0701E + 04	1.2684E + 04	1.0625E + 04	1.1622E + 04	1.0626E + 04	1.0683E + 04	1.0624E + 04
	Mean	1.0693E + 04	1.1542E + 04	1.0624E + 04	1.0920E + 04	1.0624E + 04	1.0657E + 04	1.0623E + 04
	Std.	8.1672E + 00	5.7044E + 02	4.6239E - 01	2.5921E + 02	7.6866E - 01	1.3817E + 01	2.3235E - 01
Wine	Best	1.6385E + 04	1.6493E + 04	1.6296E + 04	1.6566E + 04	1.6294E + 04	1.6316E + 04	1.6292E + 04
	Worst	1.8437E + 04	2.0245E + 04	1.6311E + 04	1.7668E + 04	1.6312E + 04	1.6371E + 04	1.6294E + 04
	Mean	1.6974E + 04	1.7999E + 04	1.6301E + 04	1.7010E + 04	1.6297E + 04	1.6345E + 04	1.6293E + 04
	Std.	8.6757E + 02	1.1562E + 03	4.5677E + 00	3.6824E + 02	4.0149E + 00	1.4836E + 01	5.2338E - 01
Balance scale	Best	1.4239E + 03	1.4238E + 03	1.4256E + 03	1.4265E + 03	1.4238E + 03	1.4239E + 03	1.4238E + 03
	Worst	1.4337E + 03	1.4291E + 03	1.4285E + 03	1.4310E + 03	1.4262E + 03	1.4260E + 03	1.4257E + 03
	Mean	1.4275E + 03	1.4259E + 03	1.4268E + 03	1.4282E + 03	1.4248E + 03	<u>1.4243E + 03</u>	1.4245E + 03
	Std.	3.6565E + 00	1.6015E + 00	7.3239E - 01	1.1831E + 00	9.7396E - 01	<u>6.8756E - 01</u>	8.6619E - 01
Haberman-Survival	Best	2.6251E + 03	2.5670E + 03	2.5670E + 03	2.5671E + 03	2.5670E + 03	2.5673E + 03	2.5670E + 03
	Worst	3.1966E + 03	2.6226E + 03	2.5670E + 03	2.5709E + 03	2.5678E + 03	2.6686E + 03	2.5678E + 03
	Mean	2.6554E + 03	2.5702E + 03	<u>2.5670E + 03</u>	2.5679E + 03	2.5671E + 03	2.5898E + 03	2.5673E + 03
	Std.	1.2740E + 02	1.2354E + 01	<u>1.7590E - 06</u>	8.3107E - 01	3.0624E - 01	2.3346E + 01	4.0711E - 01

TABLE 6: *p* values produced by Wilcoxon's rank-sum test comparing PGWO versus GWO, PSO, ABC, CS, GGSA, and *K*-means over data sets (*p* ≥ 0.05 have been underlined).

PGWO vs	GWO	PSO	ABC	CS	GGSA	<i>K</i> -means
Art	6.80E - 08	1.60E - 05	6.80E - 08	6.80E - 08	6.11E - 08	4.49E - 08
Iris	6.75E - 08	1.19E - 06	6.75E - 08	6.75E - 08	<u>0.107346</u>	5.17E - 08
Cancer	6.76E - 08	6.76E - 08	6.76E - 08	6.76E - 08	1.2E - 06	5.98E - 08
CMC	6.69E - 08	6.69E - 08	6.69E - 08	6.69E - 08	6.69E - 08	3.39E - 08
Seeds	6.80E - 08	6.80E - 08	6.80E - 08	6.80E - 08	1.23E - 03	4.81E - 08
Heart	6.61E - 08	2.02E - 06	6.61E - 08	5.87E - 07	6.61E - 08	6.07E - 08
Wine	6.80E - 08	1.23E - 07	6.80E - 08	6.80E - 08	6.80E - 08	6.63E - 08
Balance scale	<u>0.067868</u>	0.007712	6.8E - 08	1.06E - 07	0.003966	6.01E - 07
Haberman-Survival	6.90E - 07	<u>0.524948</u>	1.78E - 03	<u>0.860418</u>	<u>0.090604</u>	6.26E - 08

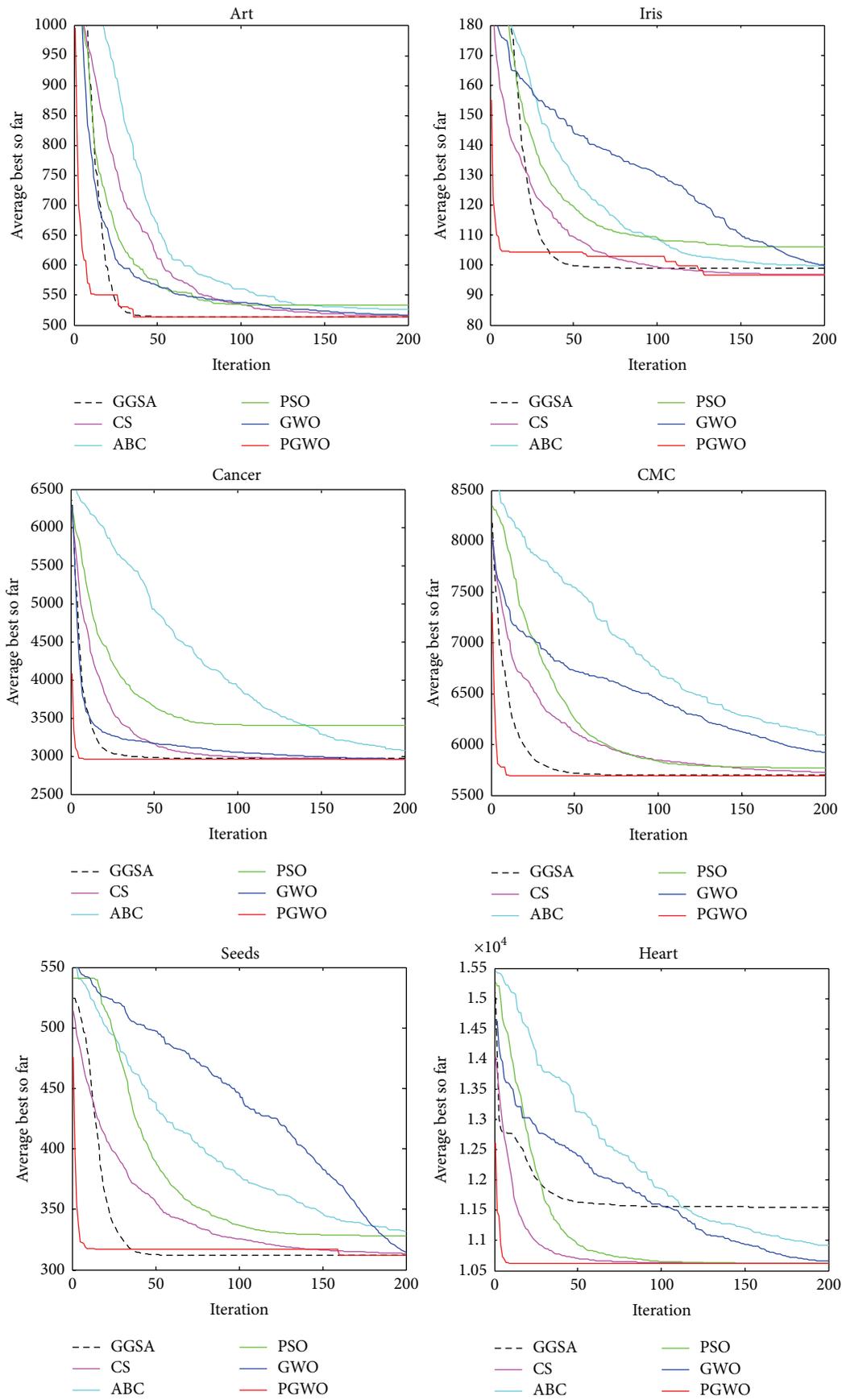


FIGURE 2: Continued.

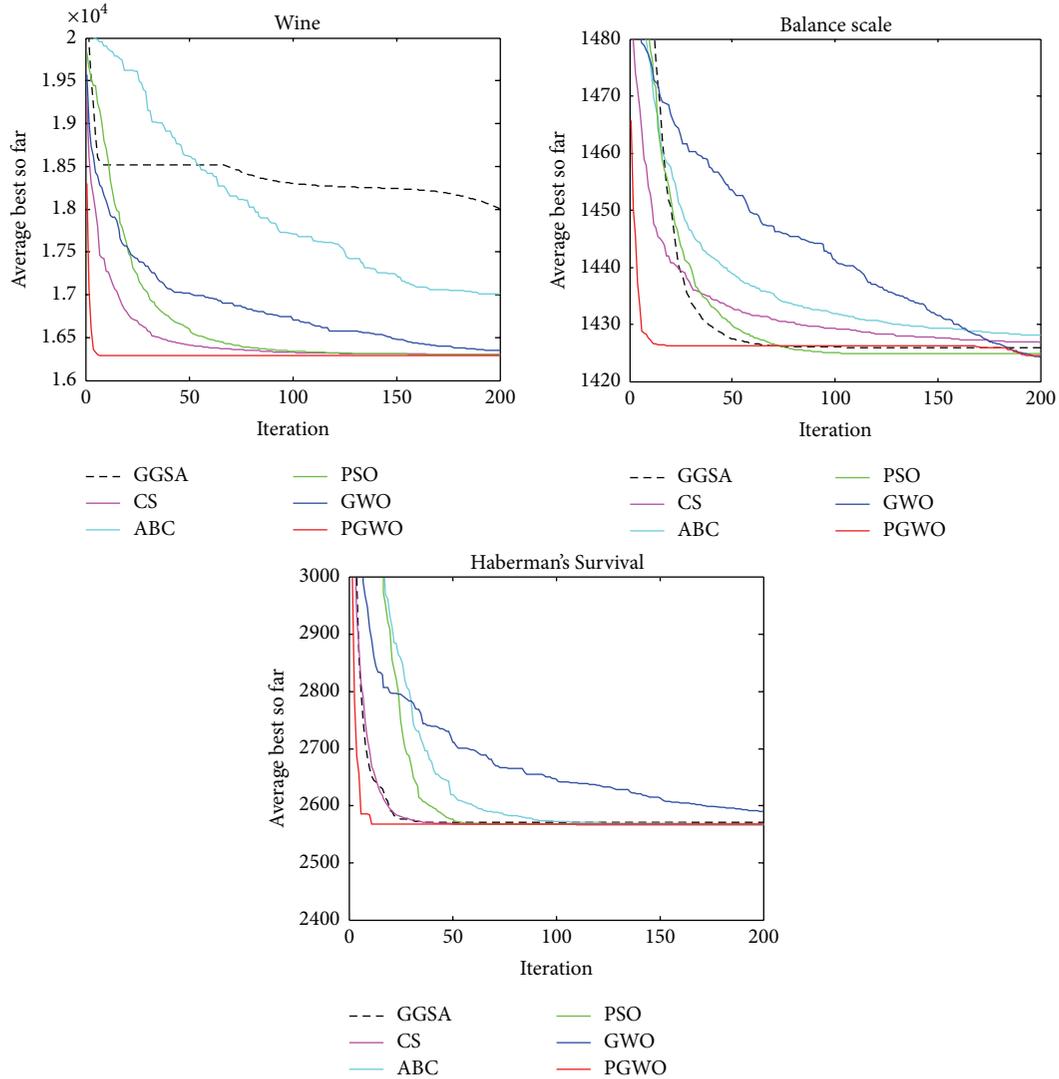


FIGURE 2: Convergence curves of clustering on data sets over 20 independent runs.

CMC data set, Seeds data set, Heart data set, and Wine data set. From Table 6, we conclude that for Iris data set PGWO is performing superior to the other algorithms except for GGSA. Since the p value for balance scale data set of PGWO versus GWO is more than 0.05, there is no statistical difference between them both. For Haberman Survival data set, while comparing PGWO and the other algorithms, we can conclude that PGWO is significantly performing better in three out of six groups compared to algorithms. So, it can be claimed that PGWO provides better results than the other algorithms across the majority of the data sets.

Figure 2 shows convergence curves of clustering on data sets over 20 independent runs. As can be seen from the figure, the convergence speed of PGWO is the fastest.

Figure 3 indicates ANOVA tests of clustering on data sets over 20 independent runs. Seen from Figure 3, PGWO is very stable for the majority of the data sets. For Art, Seeds, and CMC data sets, PGWO, PSO, and GGSA can obtain the relatively stable optimal values. For Heart and Wine data

sets, the stability of PGWO, PSO, and CS is outstanding. For Cancer data set, most of the algorithms can obtain the stable optimal value except for ABC and PSO algorithms. For Iris data set, we can clearly see that PGWO is better in terms of the stability. For balance scale data set, GWO obtains the relatively stable optimal value. For Haberman's Survival data set, the stability of CS and PSO is the best, but PGWO and GGSA follow them closely.

Clustering results of Art, Iris, and Survival data sets by PGWO algorithm are presented in Figure 4 which can make it visualized clearly. It can be seen from Figure 4 that the PGWO algorithm possesses superior effect on Art, Iris, and Survival data sets.

In summary, the results show that the proposed method successfully outperforms the other algorithms across the majority of benchmark functions. Furthermore, the test results of clustering problems show that PGWO is able to provide very competitive results, including the property of this algorithm. Therefore, it appears from this comparative

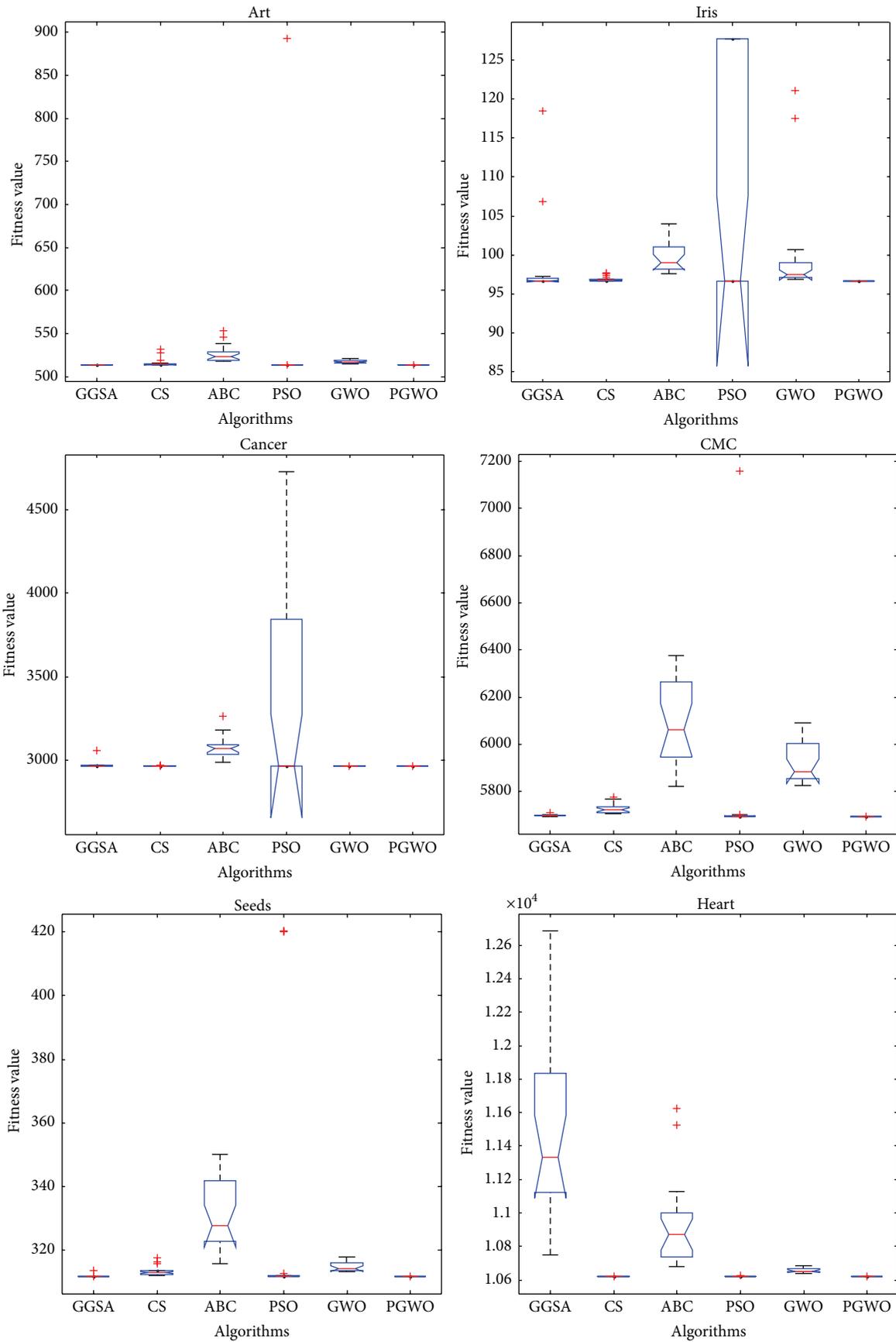


FIGURE 3: Continued.

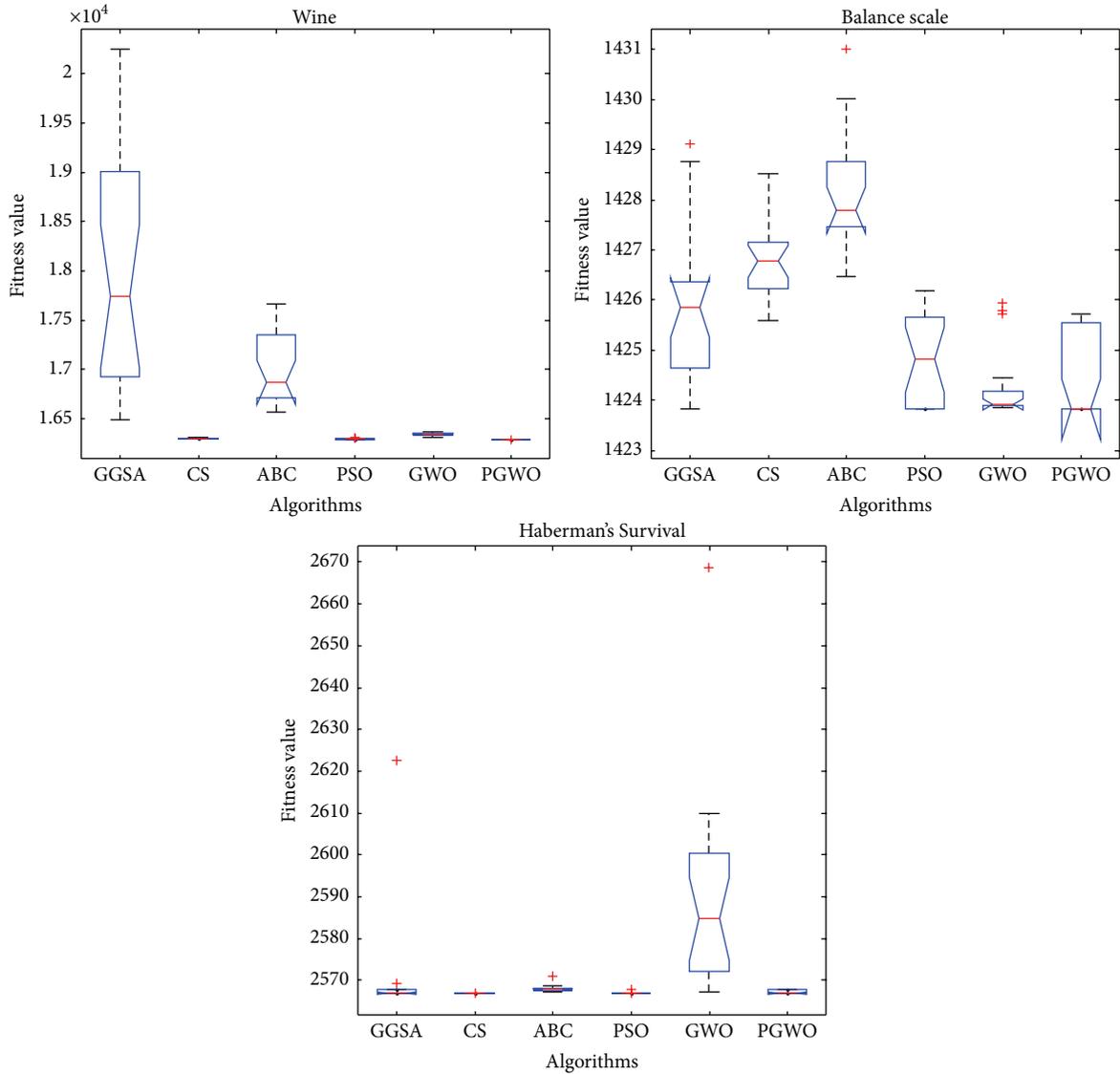


FIGURE 3: ANOVA tests of clustering on data sets over 20 independent runs.

study that the proposed method has merit in the field of evolutionary algorithm and optimization.

6. Conclusion and Future Works

In order to apply the grey wolf optimizer to solve complex optimization problems efficiently, this paper proposed a novel grey wolf optimizer based on Powell local optimization method, namely, PGWO. In PGWO, at first, original GWO algorithm is applied to shrink the search region to a more promising area. Thereafter, Powell’s method is implemented as a critical complement to perform the local search to exploit the limited area intensively to get better solutions. The PGWO makes an attempt at taking merits of the GWO and Powell’s method in order to avoid all grey wolves getting trapped in inferior local optimal regions. The PGWO enables the grey wolves to have more diverse exemplars to learn from as the grey wolves are updated each generation and

also form new grey wolves to search in a larger search space. With both techniques combined, PGWO can balance exploration and exploitation and effectively solve complex problems. The experimental results show the effectiveness of Powell’s method in terms of solution quality and convergence speed. The proposed algorithm is benchmarked on seven well-known test functions, and the results are comparative study with GGSA, CS, ABC, PSO, and GWO. The results show that the PGWO algorithm is capable of providing very competitive results compared to these famous metaheuristics. Because of the superior performance of the PGWO algorithm, we use it to solve clustering problems. The algorithm has been tested on an artificial data set and eight real data sets. To justify the performance of the PGWO algorithm on clustering problems, we compare it with the original GWO, GGSA, CS, ABC, PSO, and *K*-means. The results prove that the PGWO algorithm is able to significantly outperform others on the majority of the data sets in terms

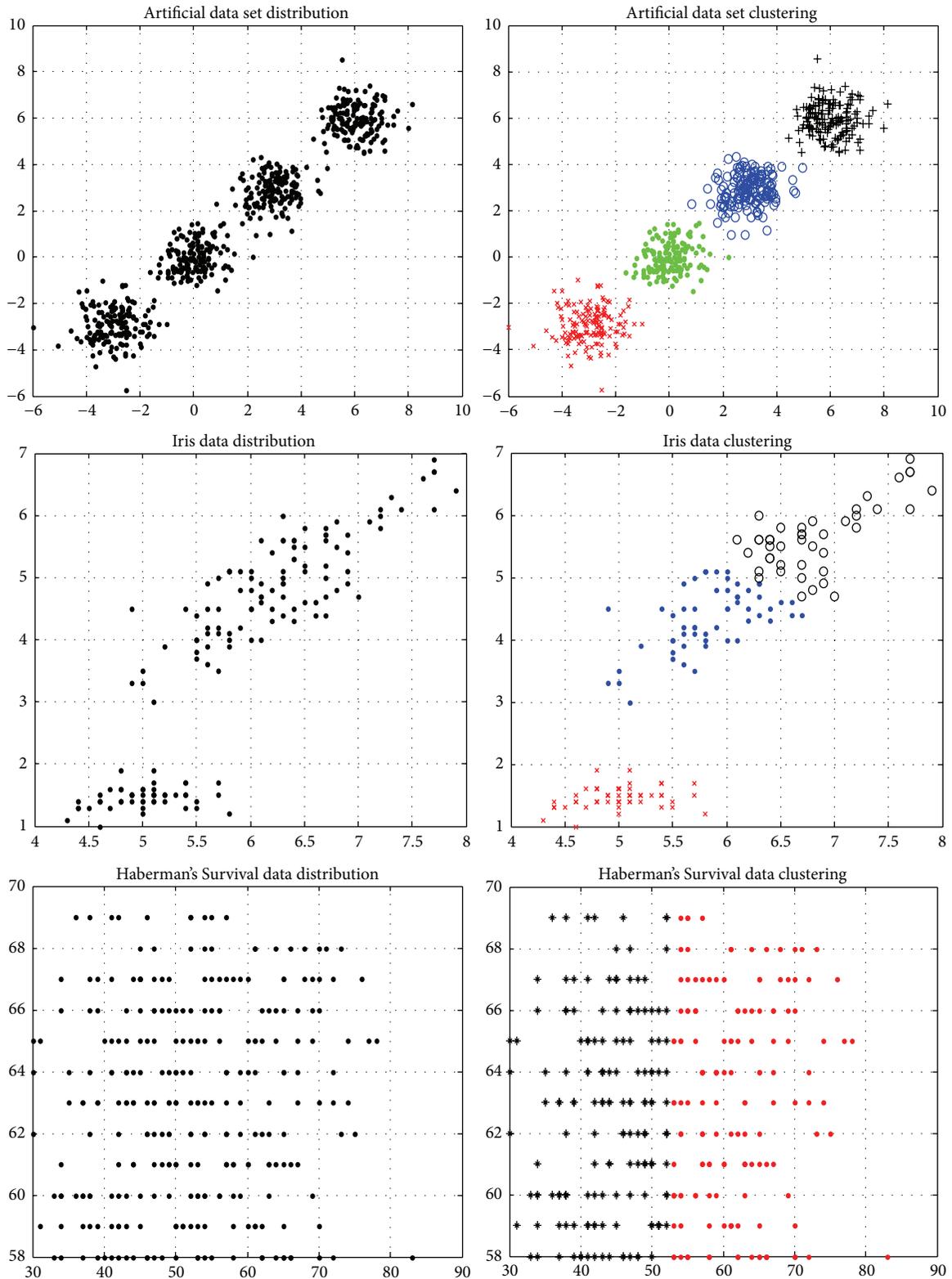


FIGURE 4: The original data distribution of Art, Iris, and Survival data sets and the clustering results by PGWO algorithm.

of average value and standard deviations of fitness function. Moreover, the experimental results demonstrate that the proposed PGWO algorithm can be considered as a feasible and efficient method to solve optimization problems.

Our future work will focus on the two issues. On one hand, we would apply our proposed PGWO algorithm to test higher dimensional problems and large number of patterns. On the other hand, the PGWO clustering algorithm will also be extended to dynamically determine the optimal number of clusters.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants nos. 61165015, 61463007, and 61563008 and the Project of Guangxi University for Nationalities Science Foundation under Grant no. 2012MDZD037.

References

- [1] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, November–December 1995.
- [3] K. Price and R. Storn, "Differential evolution," *Dr. Dobbs Journal*, vol. 22, pp. 18–20, 1997.
- [4] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer, New York, NY, USA, 2005.
- [5] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [8] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, 2015.
- [9] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [10] 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, no. 3, pp. 459–471, 2007.
- [11] X.-S. Yang, "Firefly algorithm, lévy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*, pp. 209–218, Springer, London, UK, 2010.
- [12] X.-S. Yang and S. Deb, "Cuckoo Search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, pp. 210–214, IEEE, Coimbatore, India, December 2009.
- [13] R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing*, vol. 11, no. 8, pp. 5508–5518, 2011.
- [14] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [15] S. Mirjalili and A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Computing and Applications*, vol. 25, no. 7, pp. 1569–1584, 2014.
- [16] M. H. Sulaiman, Z. Mustafa, M. R. Mohamed, and O. Aliman, "Using the gray wolf optimizer for solving optimal reactive power dispatch problem," *Applied Soft Computing*, vol. 32, pp. 286–292, 2015.
- [17] X. H. Song, L. Tang, S. T. Zhao et al., "Grey Wolf Optimizer for parameter estimation in surface waves," *Soil Dynamics and Earthquake Engineering*, vol. 75, pp. 147–157, 2015.
- [18] G. M. Komaki and V. Kayvanfar, "Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.
- [19] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257–1263, 2015.
- [20] B. Mahdad and K. Srairi, "Blackout risk prevention in a smart grid based flexible optimal strategy using Grey Wolf-pattern search algorithms," *Energy Conversion and Management*, vol. 98, pp. 411–429, 2015.
- [21] O. Kramer, "Iterated local search with Powell's method: a memetic algorithm for continuous global optimization," *Memetic Computing*, vol. 2, no. 1, pp. 69–83, 2010.
- [22] M. J. D. Powell, "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, no. 1, pp. 241–254, 1977.
- [23] I. E. Evangelou, D. G. Hadjimitsis, A. A. Lazakidou, and C. Clayton, "Data mining and knowledge discovery in complex image data using artificial neural networks," in *Proceedings of the Workshop on Complex Reasoning and Geographical Data*, Paphos, Cyprus, 2001.
- [24] M. S. Kamel and S. Z. Selim, "New algorithms for solving the fuzzy clustering problem," *Pattern Recognition*, vol. 27, no. 3, pp. 421–428, 1994.
- [25] M. Omran, A. Salman, and A. P. Engelbrecht, "Image classification using particle swarm optimization," in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, Singapore, 2002.
- [26] X. J. Lei, *Swarm Intelligent Optimization Algorithms and Their Applications*, Science Press, 2012.
- [27] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [28] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 459796, 16 pages, 2010.
- [29] B. Zhang, M. Hsu, and U. Dayal, "K-Harmonic means—a data clustering algorithm," Hewlett-Packard Labs Technical Report, HPL, 1999.
- [30] S. Z. Selim and K. Alsultan, "A simulated annealing algorithm for the clustering problem," *Pattern Recognition*, vol. 24, no. 10, pp. 1003–1008, 1991.
- [31] K. S. Al-Sultan, "A Tabu search approach to the clustering problem," *Pattern Recognition*, vol. 28, no. 9, pp. 1443–1451, 1995.

- [32] C. S. Sung and H. W. Jin, "A tabu-search-based heuristic for clustering," *Pattern Recognition*, vol. 33, no. 5, pp. 849–858, 2000.
- [33] M. C. Cowgill, R. J. Harvey, and L. T. Watson, "A genetic algorithm approach to cluster analysis," *Computers and Mathematics with Applications*, vol. 37, no. 7, pp. 99–108, 1999.
- [34] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999.
- [35] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.
- [36] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 825–832, 1996.
- [37] M. Fathian, B. Amiri, and A. Maroosi, "Application of honey-bee mating optimization algorithm on clustering," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1502–1513, 2007.
- [38] D. W. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, vol. 1, pp. 215–220, IEEE, Canberra, Australia, December 2003.
- [39] A. Hatamlou, S. Abdullah, and M. Hatamlou, "Data clustering using big bang-big crunch algorithm," in *Innovative Computing Technology*, vol. 241 of *Communications in Computer and Information Science*, pp. 383–388, Springer, 2011.
- [40] D. Karaboga and C. Ozturk, "A novel clustering approach: artificial Bee Colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 652–657, 2011.
- [41] A. Hatamlou, S. Abdullah, and H. Nezamabadi-Pour, "Application of gravitational search algorithm on data clustering," in *Rough Sets and Knowledge Technology*, vol. 6954 of *Lecture Notes in Computer Science*, pp. 337–346, Springer, Berlin, Germany, 2011.
- [42] A. Hatamlou, S. Abdullah, and H. Nezamabadi-Pour, "A combined approach for clustering based on K-means and gravitational search algorithms," *Swarm and Evolutionary Computation*, vol. 6, pp. 47–52, 2012.
- [43] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187–195, 2004.
- [44] Y. Kao and K. Cheng, "An ACO-based clustering algorithm," in *Ant Colony Optimization and Swarm Intelligence*, vol. 4150 of *Lecture Notes in Computer Science*, pp. 340–347, Springer, Berlin, Germany, 2006.
- [45] Y.-T. Kao, E. Zahara, and I.-W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1754–1762, 2008.
- [46] T. Niknam, B. Amiri, J. Olamaei, and A. Arefi, "An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering," *Journal of Zhejiang University Science A*, vol. 10, no. 4, pp. 512–519, 2009.
- [47] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "A novel artificial bee colony algorithm with Powell's method," *Applied Soft Computing*, vol. 13, no. 9, pp. 3763–3775, 2013.
- [48] H. R. Lourenco, O. Martin, and T. Stützle, "A beginner's introduction to iterated local search," in *Proceedings of the 4th Metaheuristics International Conference (MIC '01)*, vol. 2, pp. 1–6, Porto, Portugal, 2001.
- [49] T. G. Stützle, *Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications*, vol. 220 of *DISKI: Dissertations on Artificial Intelligence*, Infix Publishers, Sankt Augustin, Germany, 1999.
- [50] X. S. Yang, Ed., *Test Problems in Optimization. An Introduction with Metaheuristic Applications*, Wiley, London, UK, 2010.
- [51] M. Molga and C. Smutnicki, "Test functions for optimization-needs," 2005, <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>.
- [52] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [53] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.
- [54] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [55] J. W. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2011.
- [56] C. Blake and C. J. Merz, "UCI Repository of Machine Learning Databases," 1998.
- [57] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 183–197, 2010.
- [58] E. Anderson, "The irises of the Gaspé peninsula," *Bulletin of the American Iris Society*, vol. 59, pp. 2–5, 1935.
- [59] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

