



An improved simplified swarm optimization

Wei-Chang Yeh

Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan



ARTICLE INFO

Article history:

Received 3 February 2014

Received in revised form 21 January 2015

Accepted 22 February 2015

Available online 4 March 2015

Keywords:

Swarm Intelligence (SI)

Simplified Swarm Optimization (SSO)

Artificial Bee Colony Algorithm (ABC)

Soft Computing (SC)

Optimization problems

ABSTRACT

This paper introduces an improved simplified swarm optimization (iSSO) by undertaking a major revision of the update mechanism (UM) of traditional SSO. To test its performance, the proposed iSSO is compared with another recently introduced swarm-based algorithm, the Artificial Bee Colony Algorithm (ABC), on 50 different widely used multivariable and multimodal numerical test functions. Numerical examples conclude that the proposed iSSO outperforms ABC in both solution quality and efficiency. We also test the roles of the proposed UMs and iterative local search. The proposed algorithm is thus useful to both practitioners and researchers.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Since the early 1990s, Soft Computing (SC) has been utilized to obtain optimal or good-quality solutions to difficult optimization problems in fields for which exact analytical methods do not produce optimal solutions in an acceptable amount of time, especially for problems that are large in size [51]. It is well known that SC mainly involves fuzzy sets, neural networks, genetic algorithms, and rough sets. In recent years, we have seen an increasing interest in SC in creating methodologies with the aims of resembling and simulating phenomena of nature to solve larger problems in science and technology; resulting methods include Artificial Neural Network (ANN) [9,25], Simulated Annealing (SA) [23], Tabu Search (TS) [13], Genetic Algorithm (GA) [10,14], Ant Colony Optimization (ACO) [7], Particle Swarm Optimization (PSO) [21,22,28,12,40], Estimation Distribution of Algorithm (EDA) [24], Differential Evolution [31,32], Artificial Bee Colony Algorithm (ABC) [16,17,19,20], Simplified Swarm Optimization (SSO) [44–50], Firefly algorithm [41], Biogeography-based Optimization [29], Cuckoo search [42,39], Bat Algorithm [43], and Krill Herd [11,15,34–38].

Swarm Intelligence (SI) is a newly developed branch of SC that belongs to the category of population-based stochastic optimization. Swarm Intelligence (SI) was introduced by Gerardo Beni and Jing Wang in 1989 in the context of cellular robotic systems [3]. Bonabeau, Dorigo and Theraulaz defined SI as “any attempt to design algorithms or distributed problem-solving devices inspired

by the collective behavior of social insect colonies and other animal societies” [4]. SI has also received significant interest from researchers studying in a range of research areas and has been applied to several real-world problems.

ABC is one of the most recently introduced SI algorithms and was originally designed by Karabog in 2005 [19] to simulate the intelligent foraging behavior of a honeybee swarm, based on the model proposed by Tereshko and Loengarov [33]. Research in the literature indicates that the performance of the ABC is better than or similar to that of other population-based algorithms, including GA, PSO, and EDA [16,17,19,20]. Hence, ABC is selected to compare with the proposed approach.

SSO is an emerging population-based stochastic optimization method [44–48,2,1,6,5,27,30,26,8]. It is another new SI method and is also an evolutionary computation method. SSO was proposed by Yeh to overcome the drawbacks of PSO [44]. The advantages of SSO are its simplicity (easy to implement and having only a few parameters to tune), efficiency (a fast convergence rate), and flexibility. Simulation results reveal that SSO has better convergence to quality solutions than PSO, GA, EDA, and ANN [44–48]. SSO has therefore attracted considerable attention and has been widely used in a range of applications, such as redundancy allocation [44,30], data mining [45,48,27,26], health care management [6,5,8], and supply chain management [46,47,2,1].

The goal of this work is to propose an improved SSO (here termed iSSO). iSSO is able to address non-discrete data. To demonstrate the efficiency of the proposed iSSO, a comprehensive comparative study is presented on the performances of the well-known swarm-based algorithms SSO and ABC and the variant

E-mail address: yeh@iee.org

iSSO for optimizing a large set of multi-variable and multi-modal numerical test functions. For iSSO, there is also consideration of using iterative local search (ILS) to the best solution, ILS to all improved solutions, or neither.

This paper is organized as follows. Section 2 provides respective descriptions of ABC, which may be the best-known SC for solving numerical functions, and SSO, which is the basis of the proposed iSSO and related UM. The proposed UM and iSSO are discussed in Section 3. A comprehensive comparative study on the performances of the proposed UM with or without ILS and SSO based on 50 benchmark functions is given in Section 4. The complete comparison of the proposed iSSO and ABC, together with the proposed UM, is discussed in Section 5. Finally, the conclusion and discussion are given in Section 6.

2. Overview of ABC and SSO

Let $Nvar$, $Nsol$, $Ngen$, and $Nrun$ represent the number of variables (attributes, dimensions, genes), solutions, (populations chromosomes, particles), generations (called cycles in ABC), and independent replications. Most SCs, e.g., GA, SA, ACO, PSO, ABC, SSO, and the proposed iSSO, generate randomly distributed initial solutions with population size $Nsol$ inside the problem space and then search for optimal solutions by updating generations. Each solution is encoded as a finite-length string with a fitness value. Note that large $Nsol$ and $Ngen$ will slow down the optimization process significantly, and small $Nsol$ and $Ngen$ will be unable to find a good solution for the requirements.

Let $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iNvar}^t)$ be the i th solution (called a food source in ABC) at generation t , x_{ij}^t be the j th variable of X_i^t , and $f(X_i^t)$ be the fitness function of X_i^t . The major difference among all SC methods is the UM that is used to update the current solutions. Hence, we focus on the UMs of ABC, SSO, and iSSO in the rest of the study. Sometimes, ILSs are implemented after using UMs to improve solutions by moving them to their neighbors according to neighborhood structures, but this is done at the expense of running time.

2.1. ABC

In ABC, the quality of the possible solution X_i^t is denoted by $F(X_i^t)$ and defined as follows for $i = 1, 2, \dots, Nsol$ and $t = 1, 2, \dots, Ngen$ [16,17,19,20].

$$F(X_i^t) = \begin{cases} \frac{1}{f(X_i^t)+1}, & f(X_i^t) \geq 0 \\ 1 + |f(X_i^t)|, & \text{otherwise} \end{cases} \quad (1)$$

There are three UMs, called the employed bee, the onlooker (bee), and the scout (bee), in ABC. The first half of the bee colony implements employed bees, while the second half implements onlookers. ABC assumes only one employed bee per food source.

In the first UM, “the employed bees,” X_i^t is updated to X_i^{t+1} based on the following equation:

$$x_{ik}^{t+1} = x_{ik}^t + \rho_{[-1,1]} \cdot (x_{ik}^t - x_{jk}^t), \quad (2)$$

where both the variable k and solution X_j^t are selected randomly and ρ_l is a uniform variable in the interval l , where $i, j = 1, 2, \dots, Nsol$, $k = 1, 2, \dots, Nvar$, and $t = 0, 1, 2, \dots, Ngen - 1$. If x_{ik}^{t+1} falls outside the acceptable range, it is set to the corresponding extreme value in that range. Moreover, if $F(X_i^{t+1})$ is worse than $F(X_i^t)$, then X_i^{t+1} is replaced by X_i^t (called a greedy selection mechanism in ABC).

To guide ABC toward more highly fit positions of the search space, the second UM, “the onlooker,” is a local search operator.

After a solution, say, X_i^t , is generated or updated, the onlooker bee is implemented $Nsol$ times with probability $Pr(X_i^t)$ defined as follows:

$$Pr(X_i^t) = \frac{F(X_i^t)}{\sum_{i=1}^{Nsol} F(X_i^t)}. \quad (3)$$

If a solution, say, X_i^t , does not improve for a predetermined number of iterations

$$Nlim = Nsol \times Nvar, \quad (4)$$

then the third UM, “the scout,” is a special ILS that is implemented to regenerate X_i^t randomly and repeatedly.

2.2. SSO

The proposed iSSO is based on SSO. Before discussing the proposed iSSO, basic SSO is introduced formally in this sub-section. Let $pBest_i = (p_{i1}, p_{i2}, \dots, p_{iNvar})$ be the best fitness function value of the i th solution with its own history and $gBest = (g_1, g_2, \dots, g_{Nvar})$ be the solution with the best fitness function value among all $pBest$ s, where $i = 1, 2, \dots, Nsol$.

The fundamental concept of SSO is that each variable of any solution needs to be updated to a value related to its current value, its current $pBest$ (as a local search), the $gBest$ (as a global search), or a random feasible value to maintain population diversity and enhance the capacity to escape from a local optimum [44–48]. The UM of SSO is based only on the following simple mathematical modeling after c_w , c_p , and c_g are given:

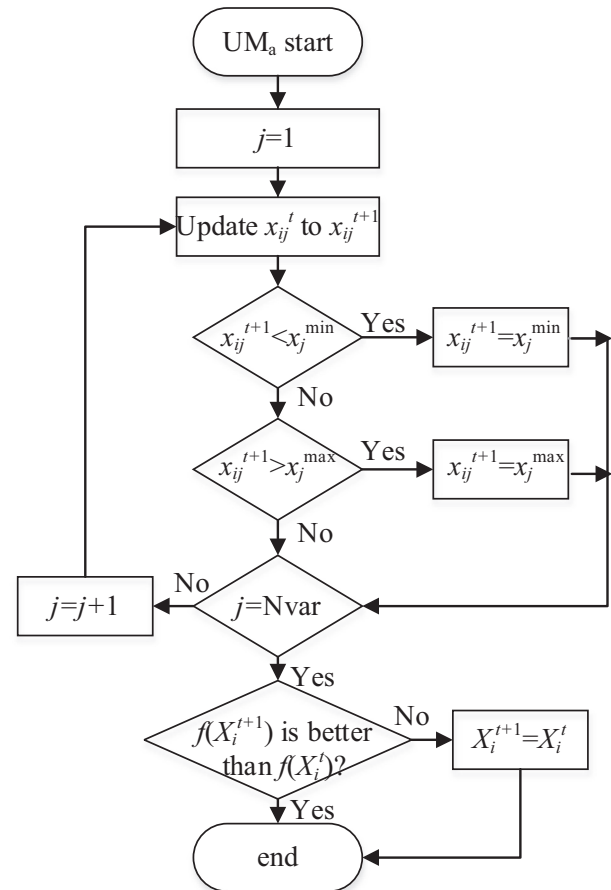


Fig. 1. The flowchart of the proposed UM_a.

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t & \text{if } \rho_{[0,1]} \in [0, C_w = c_w) \\ p_{ij}^t & \text{if } \rho_{[0,1]} \in [C_w, C_p = C_w + c_p) \\ g_j & \text{if } \rho_{[0,1]} \in [C_p, C_g = C_p + c_g) \\ x & \text{if } \rho_{[0,1]} \in [C_g, 1) \end{cases}, \quad (5)$$

where $i = 1, 2, \dots, \text{Nsol}$, $j = 1, 2, \dots, \text{Nvar}$, $t = 0, 1, 2, \dots, \text{Ngen} - 1$, and x is a random number between the lower bound and the upper bound of the j th variable. Note that c_w , c_p , and c_g can be fixed or flexible.

The UM of SSO is much simpler than those of other major soft computing techniques, such as PSO (which must calculate both the velocity and position functions), GA (which requires genetic operations such as crossover and mutation), EDA (which has difficulty building an appropriate probability model), and the Immune System Algorithm (which does not consider the interaction of variables) [44–48]. Using the above simple approaches, the diversity of a population can be maintained efficiently.

3. The proposed iSSO

In this section, iSSO is proposed together with the all-variable-UM (called UM_a hereafter). Furthermore, one ILS is proposed to help improve solution quality.

3.1. A new formulation to revise Eq. (5) for floating-point data

In the traditional SSO, all variables are changed simultaneously based on Eq. (5). In the proposed iSSO, Eq. (5) is revised for floating-point data listed as follows:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + \rho_{[-.5,.5]} \cdot u_j & \text{if } x_{ij}^t = g_j \text{ or } \rho_{[0,1]} \in [0, C_r = c_r) \\ g_j + \rho_{[-.5,.5]} \cdot u_j & \text{if } x_{ij}^t \neq g_j \text{ and } \rho_{[0,1]} \in [C_r, C_g = C_r + c_g) \\ x_{ij}^t + \rho_{[-.5,.5]} \cdot (x_{ij}^t - g_j) & \text{if } x_{ij}^t \neq g_j \text{ and } \rho_{[0,1]} \in [C_g, 1 = C_r + c_g + c_w) \end{cases}, \quad (6)$$

where

$$u_j = \frac{x_j^{\min} - x_j^{\max}}{2 \cdot \text{Nvar}}, \quad (7)$$

x_j^{\min} and x_j^{\max} are the lower bound and upper bound of the j th variable.

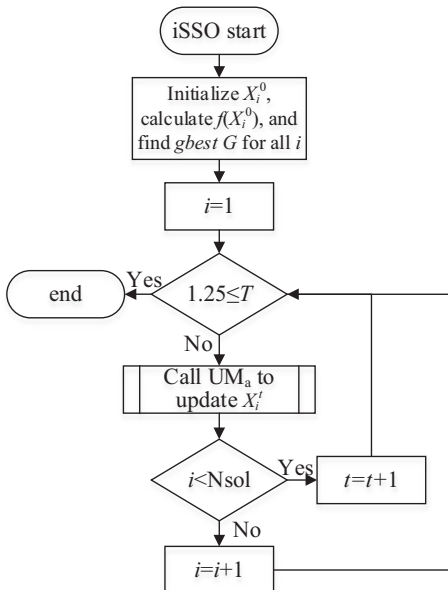


Fig. 2. The flowchart of the proposed iSSO.

Eq. (6) represents a major revision of fundamental SSO to fit the continuous variables as follows:

1. A variable is updated to its neighborhood using Eq. (7) if it is selected based on c_r , i.e., the first item in Eq. (6).
2. A variable is updated to the neighborhood of $gbest$ if it is selected based on c_g , i.e., the second item in Eq. (6).
3. A variable is updated from the interval between itself and $gbest$ if it is selected based on c_w , i.e., the third item in Eq. (6).

If any variable violates the boundary condition, it is set to the nearest boundary after using Eq. (6). Moreover, unlike SSO, x_i^{t+1} is set to x_i^t after updating in the proposed iSSO if $f(x_i^{t+1})$ is not better than $f(x_i^t)$, i.e.,

$$x_i^{t+1} = \begin{cases} x_i^t & \text{if } f(x_i^{t+1}) \text{ is worse than } f(x_i^t) \\ x_i^{t+1} & \text{otherwise} \end{cases}. \quad (8)$$

3.2. The stopping criteria

There are four major stopping criteria used in SC:

1. The generation number.
2. The number to calculate the fitness function.
3. The number of unchanged best solutions.
4. The time limit.

One population (solution/particle/chromosome) calculates the fitness function only once in each generation for any SC method not using ILS, e.g., SSO, GA, PSO, etc., but that is not always true for those algorithms using ILS, e.g., ABC. Moreover, some SC methods take a long time and use complicated methods, e.g., the second UM, “the onlooker,” in ABC, before calculating the fitness function or selecting the populations to update, e.g., the roulette wheel selection in GA. Hence, to make a fair comparison of the proposed iSSO with ABC and related SSO variants, a time limit of 1.25 s is adopted here as the stopping criterion. For easy observation of the trends and changes in the entire evolution process, the 1.25 s is separated into five periods, i.e., 0 ~ .25, .25 ~ .5, .5 ~ .75, .75 ~ .1, and 1 ~ 1.25 (s).

3.3. ILS

It is well-known that ILS can improve solution quality. To test whether it is necessary to add ILS in the proposed iSSO, the first item in Eq. (6) is used as the UM in ILS, i.e.,

$$x_j = x_j + \rho_{[-.5,.5]} u_j, \quad (9)$$

where u_j is defined in Eq. (7) and x_j is a variable selected randomly from a solution, say, X . Eq. (9) is implemented repeatedly on X from the first variable to the last variable, i.e., Nvar consecutive times in total. The fitness function of the new X is calculated immediately after each implementation of Eq. (9). Note that X can only be updated to be a better value; otherwise, X retains its original value. After Nvar times, another Nvar consecutive iterations of ILS are implemented if X is improved. The above procedure is repeated until there is no improvement during the corresponding Nvar times.

3.4. UM_a : the all-variable one-by-one UM

UM_a is implemented in the proposed iSSO to search for the optimum. In UM_a , all variables are updated one-by-one in each solution, say, x_i^t , as follows:

Table 1

Summary of the experiment results of four methods (the best related value is in bold).

Statistic	Time Interval	FE				FV			
		iSSO	gILS	aILS	SSO	iSSO	gILS	aILS	SSO
AVG	T1	1335183.9	1333756.1	1806047.6	835601.8	1.3455781	5.3640956	3.5424667	47933.93
	T2	2656993.3	2654465.2	3617158.6	1673832.6	0.3831000	1.8924000	1.5560000	18712.42
	T3	3927269.6	3923229.9	5427622.4	2513421.4	0.2073040	1.0323360	1.0113460	9571.63
	T4	5198803.8	5193403.9	7238555.1	3353606.3	0.1377380	0.7175350	0.7011530	6449.81
	T5	6476050.9	6469136.6	9049906.8	4194157.9	0.1074680	0.5256640	0.5272140	4151.25
MAX	T1	1344834.2	1347033.2	1825484.7	845089.9	5.6127925	32.6709420	19.4711380	111061.45
	T2	2681551.7	2683068.2	3651585.9	1690385.1	1.6542640	7.6965330	6.3887370	40176.69
	T3	3962665.6	3964013.0	5474271.8	2535301.1	0.4323190	4.1100630	3.4669850	22335.18
	T4	5244009.7	5245987.1	7297779.0	3380110.9	0.2588170	2.6366070	2.8424040	11835.42
	T5	6532295.7	6533568.2	9121643.1	4225100.1	0.1758560	0.9947360	1.1345590	5147.19
MIN	T1	1319430.2	1316612.2	1783818.5	825257.2	0.2995758	0.1166570	0.2087804	13990.64
	T2	2625939.1	2619613.8	3567146.9	1653782.1	0.0347170	0.0749040	0.1068930	7818.31
	T3	3881801.0	3872226.0	5351721.0	2481927.9	0.0302280	0.0521730	0.0963770	3952.38
	T4	5139957.3	5125659.2	7137882.1	3311515.5	0.0131000	0.0410530	0.0600350	2827.49
	T5	6403762.8	6387105.4	8925498.7	4141046.5	0.0061480	0.0390010	0.0580640	2162.23
STDEV	T1	6137.8	7615.7	9591.8	6510.5	1.2854442	7.4729097	4.6387647	24611.19
	T2	13186.2	15646.8	19090.6	12154.0	0.3117280	1.9878590	1.5915540	8953.34
	T3	19195.6	22911.3	27794.2	16939.8	0.1101610	0.9950220	1.0220590	4090.46
	T4	24696.9	30078.8	36399.8	21033.7	0.0723610	0.6669280	0.6923780	2209.58
	T5	30367.6	36951.9	44332.7	24449.5	0.0553040	0.3054130	0.3296740	756.86

Table 2

Rankings of the experiment results listed in Table 1 (the best related value is in bold).

Statistic	Time Interval	FE				FV			
		iSSO	gILS	aILS	SSO	iSSO	gILS	aILS	SSO
AVG	T1	3	2	1	4	1	3	2	4
	T2	3	2	1	4	1	3	2	4
	T3	3	2	1	4	1	3	2	4
	T4	3	2	1	4	1	2	3	4
	T5	3	2	1	4	1	2	3	4
MAX	T1	2	3	1	4	3	1	2	4
	T2	2	3	1	4	1	2	3	4
	T3	2	3	1	4	1	2	3	4
	T4	2	3	1	4	1	2	3	4
	T5	2	3	1	4	1	2	3	4
MIN	T1	3	2	1	4	1	3	2	4
	T2	3	2	1	4	1	3	2	4
	T3	3	2	1	4	1	3	2	4
	T4	3	2	1	4	1	2	3	4
	T5	3	2	1	4	1	2	3	4
STDEV	T1	3	1	2	4	1	3	2	4
	T2	3	1	2	4	1	3	2	4
	T3	3	1	2	4	1	3	2	4
	T4	3	1	2	4	1	2	3	4
	T5	3	1	2	4	1	2	3	4

STEP 1a. Let $j = 1$.**STEP 2a.** Update x_{ij}^t to x_{ij}^{t+1} using Eq. (6).**STEP 3a.** Let $x_{ij}^{t+1} = x_{ij}^{\min}$ if $x_{ij}^{t+1} < x_{ij}^{\min}$ or $x_{ij}^{t+1} = x_{ij}^{\max}$ if $x_{ij}^{t+1} > x_{ij}^{\max}$.**STEP 4a.** If $j < Nvar$, let $j = j + 1$ and go to STEP 2a.**STEP 5a.** Let $X_i^{t+1} = X_i^t$ if $f(X_i^{t+1})$ is not better than $f(X_i^t)$.The flowchart of the proposed UM_a is shown in Fig. 1.

3.5. The expected number of comparisons for the UM in each iteration

Each solution needs to be updated based on Eq. (5) in SSO. Because $c_r < c_w < c_p < c_g$, the expected number of comparisons to each solution in Eq. (5) is

$$Nvar \cdot (c_w + 2c_p + 3c_g + 4c_r)$$

$$= Nvar \cdot (1 + c_p + 2c_g + 3c_r)$$

$$\geq Nvar \cdot (1 + c_w + c_p + c_g + 3c_r) \quad (10)$$

$$\geq Nvar \cdot (2 + 2c_r)$$

$$> 2 \cdot Nvar$$

In Eq. (6), we let $C_r = c_r$ and $C_g = C_r + c_g$ instead of $C_g = c_g$, $C_p = C_g + c_p$, $C_w = C_p + c_w$ and remove $pBest$ from the traditional SSO. The expected number of comparisons to each variable is reduced to

$$c_r + 2c_g + 3c_w = 1 + c_g + 2c_w < 2, \quad (11)$$

$$Nvar \cdot (c_r + 2c_g + 3c_w) = Nvar \cdot (1 + c_g + 2c_w) < 2 \cdot Nvar. \quad (12)$$

Both Eqs. (11) and (12) are less than that of SSO listed in Eq. (10) without considering the convergent rate.

3.6. The pseudo-code for the proposed iSSO

Let T be the current runtime. The flowchart of the proposed iSSO is shown in Fig. 2, and the overall procedure is summarized as follows:

Procedure of iSSO

STEP 1. Initialize X_i^0 randomly, calculate $f(X_i^0)$, find $gbest$ G , and let $t = 1$ for $i = 1, 2, \dots, Nsol$.

STEP 2. Let $i = 1$.

STEP 3. If $1.25 \leq T$, halt.

STEP 4. Implement UM_a to update X_i^t from STEP 1a to STEP 5a as listed in Section 3.4.

STEP 5. If $i < Nsol$, let $i = i + 1$ and go to STEP 3. Otherwise, let $t = t + 1$ and go to STEP 2.

4. Test on four SSO-based algorithms

To evaluate the quality and performance of the proposed iSSO with or without ILS, three iSSO variants and SSO are to be tested:

Table 3
Summary of the experiment results of iSSO and ABC (the best related value is in bold).

Statistic	Time interval	FE			FV		
		iSSO	ABC	iSSO/ABC	iSSO	ABC	iSSO/ABC
AVG	T1	1335183.9	872476.4	1.5303	1.34558	4.08515	0.3294
	T2	2656993.3	1696162.4	1.5665	0.38310	0.87549	0.4376
	T3	3927269.6	2519778.0	1.5586	0.20730	0.31333	0.6616
	T4	5198803.8	3343611.7	1.5548	0.13774	0.20785	0.6627
	T5	6476050.9	4168152.8	1.5537	0.10747	0.11666	0.9212
MIN	T1	1319430.2	856835.2	1.5399	0.29958	0.24393	1.2281
	T2	2625939.1	1671690.9	1.5708	0.03472	0.02865	1.2117
	T3	3881801.0	2492223.5	1.5576	0.03023	0.02516	1.2016
	T4	5139957.3	3309898.4	1.5529	0.01310	0.02043	0.6413
	T5	6403762.8	4131398.4	1.5500	0.00615	0.00827	0.7436
MAX	T1	1344834.2	877049.9	1.5334	5.61279	14.37155	0.3905
	T2	2681551.7	1702842.5	1.5748	1.65426	4.34798	0.3805
	T3	3962665.6	2528069.9	1.5675	0.43232	1.69584	0.2549
	T4	5244009.7	3353882.6	1.5636	0.25882	1.34131	0.1930
	T5	6532295.7	4179612.4	1.5629	0.17586	0.18617	0.9446
STDEV	T1	6137.8	4394.5	1.3967	1.28544	3.69092	0.3483
	T2	13186.2	7285.0	1.8100	0.31173	1.16458	0.2677
	T3	19195.6	8378.9	2.2910	0.11016	0.36862	0.2988
	T4	24696.9	9876.9	2.5005	0.07236	0.27098	0.2670
	T5	30367.6	11036.8	2.7515	0.05530	0.05045	1.0961

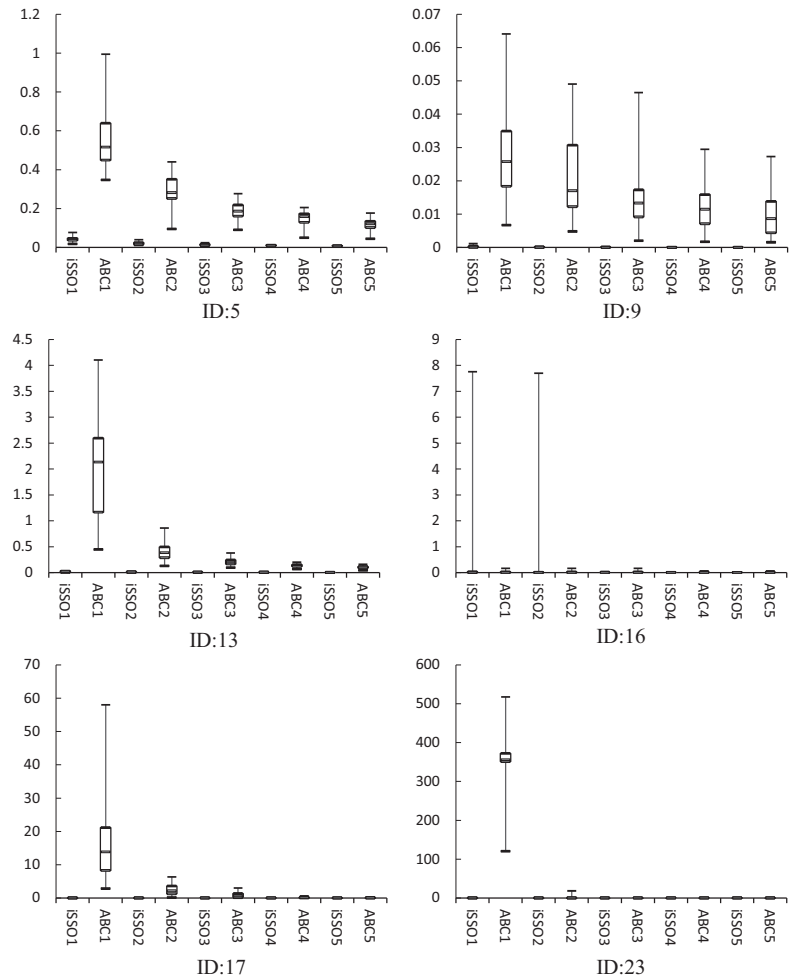


Fig. 3. Boxplots for these FV obtained from iSSO or ABC that are greater than 10^{-5} .

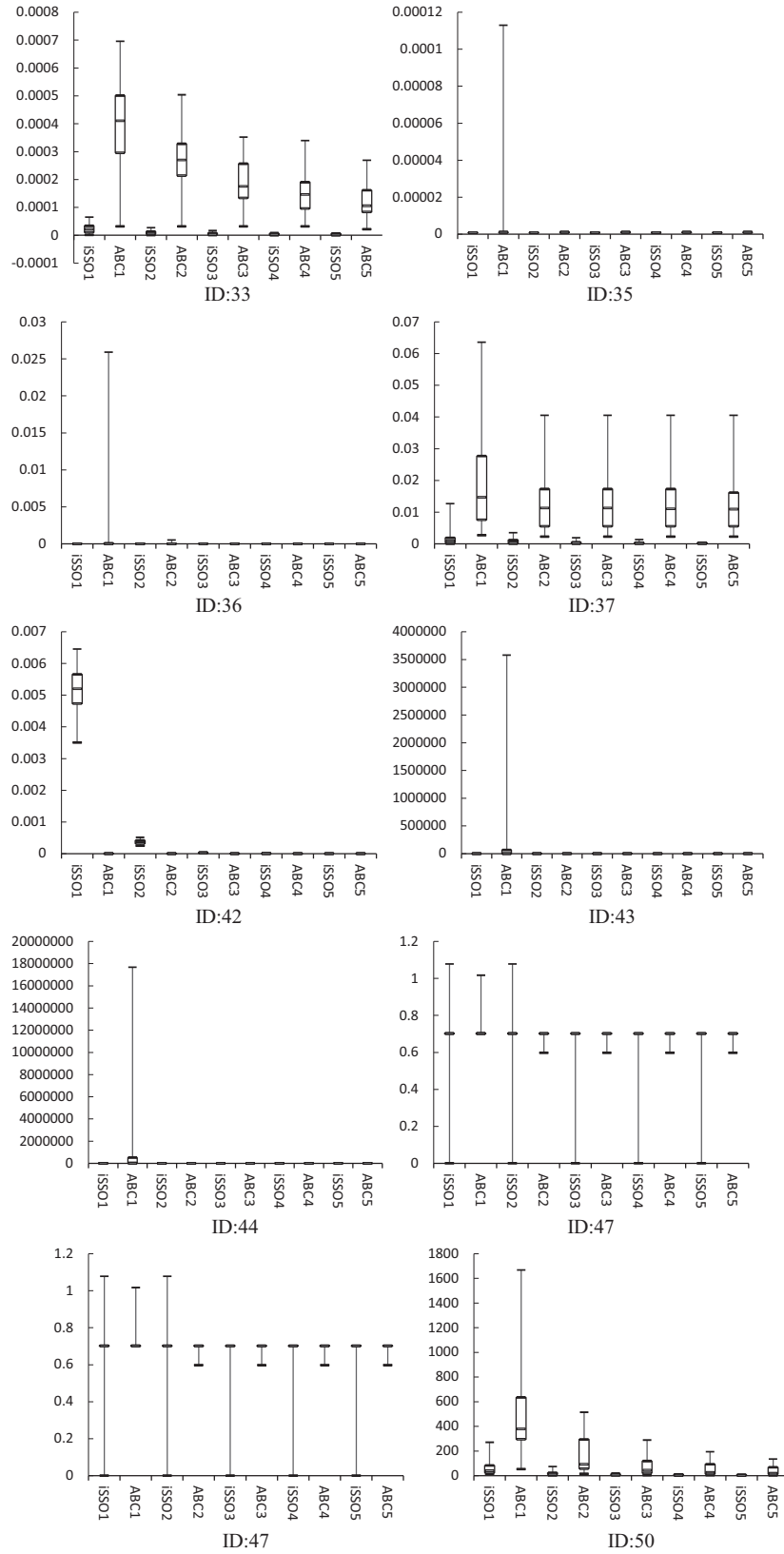


Fig. 3 (continued)

- (1) iSSO: the proposed iSSO with no use of any ILS;
- (2) gILS: the same as the proposed iSSO, but it implements Eq. (9) for each *gbest* in all generations;
- (3) aILS: the same as the proposed gILS, but it also implements Eq. (9) for any improved solutions and includes the *gbest* in the initial solutions;

- (4) SSO: the traditional SSO with $c_g = .4$, $c_p = .3$, $c_w = .2$, and $c_r = .1$.

These four methods are applied to 50 benchmark functions [20], which are presented in Appendix A. Appendix A also lists the corresponding initial ranges, formulations, characteristics, dimensions, and global optimum values of these problems. These 50 benchmark functions are well-known and have been extensively used in literature to evaluate alternative approaches in soft computing, such as ABC, GA, DEA, PSO and EA [20]. Moreover, these problems are large enough to include many different types of problems, such as unimodal, multimodal, regular, irregular, separable, non-separable and multidimensional issues.

All proposed iSSO variants and SSO are implemented in the C programming language and run on an Intel Core i7 3.07 GHz PC with 6 GB memory. The runtime unit is CPU seconds. Moreover, $c_r = .45$, $c_g = .40$, $c_w = .15$ (in Eq. (6)), $N_{sol} = 50$, and $N_{run} = 30$ (i.e., each independent replication needs to optimize 50 benchmark functions). The stopping criterion is 1.25 CPU sec as discussed in Section 3.2. The corresponding fitness function evaluation number (FE) and fitness function values (FV) are recorded at the end of each time interval, where $T_1 = (0, .25]$, $T_2 = (.25, .5]$, $T_3 = (.5, .75]$, $T_4 = (.75, 1.]$, and $T_5 = (1, 1.25]$. Let the notation AVG, MAX, MIN, and STDEV denote the average, maximal (the worst), minimal (the best), and standard deviation of the related values.

The experimental results of the four methods are summarized in Table 1 and ranked in Table 2.

It is interesting to observe that $iSSO < gILS < aILS$ (the smaller the better) for all statistics (AVG, MAX, MIN, and STDEV) from T_1 to T_5 in both FE and FV, i.e., the methods where $aILS/gILS$ are not implemented are better than those methods where ILS is implemented. The reason may be that the proposed UM_a is good enough in both global search and local search. However, $aILS$ and Eq. (9) are only effective in local search, which can be fully replaced by Eq. (6).

From Tables 1 and 2, the FE of SSO is much better than that of the rest of the methods for all time intervals and all aspects (AVG, MAX, MIN, and STDEV). The reason is that Eq. (5) is more efficient than Eq. (6). However, the FV of SSO is the worst of all four methods and shows a great difference, i.e., the proposed new UMs based on Eq. (6) are much more effective than the UM based on Eq. (5) in SSO with respect to continuous types of variables. Hence, we only focus on the comparison of the three iSSO variants in the rest of this section.

Moreover, iSSO demonstrates the best results of all three methods in AVG, MAX, MIN and STDEV from T_1 to T_6 . These observations can also be found in Table 2: iSSO is the best among all of the methods in FV.

5. Comparison between the proposed iSSO and ABC

Based on Section 4, the better average FVs in each time interval are found by iSSO. Hence, the proposed iSSO is implemented using UM_a without ILS . The results obtained from ABC outperform the results presented by DE, PSO and EA [16,17,19,20]. Additionally, ABC is the only one tested on these 50 benchmark problems. Hence, we only compare the proposed iSSO with ABC. Note that the ABC code is provided from the ABC website: <http://mf.erciyes.edu.tr/abc/>.

For fair comparison, ABC is also implemented in the C programming language and run on an Intel Core i7 3.07 GHz PC with 6 GB memory, the runtime unit is CPU seconds, and the stopping criterion is 1.25 s, as discussed in Section 3.2, just like the experiments in Section 4. The corresponding fitness function evaluation number (FE) and fitness function values (FV) are also recorded at the end of

each time interval of $T_1 = (0, .25]$, $T_2 = (.25, .5]$, $T_3 = (.5, .75]$, $T_4 = (.75, 1.]$, and $T_5 = (1, 1.25]$.

Moreover, the colony population is 100; the number of employed bees, onlooker bees and food resources are all 50; and the number of scout bees was selected to be at most one for each cycle in ABC [16,17,19,20].

5.1. General observations

In this section, we focus on comparing the average values (AVG), the worst values (MAX), the best values (MIN), and the standard deviation values (STDEV) of the obtained solutions to see the overall performance of iSSO and ABC. Table 3 lists all of the average values of iSSO and ABC in AVG, MIN, MAX, and STDEV for T_1 – T_5 ; Fig. 3 shows all of the boxplots for these FV obtained from iSSO or ABC that are greater than 10^{-5} ; Table 4 lists the number of better related values obtained from ABC and iSSO for each benchmark with 30 replicates.

From Tables 3 and 4 and Fig. 3, in general:

1. FE: ABC is always more stable and better than iSSO in FE, and the values of iSSO/ABC are almost fixed for AVG, MAX, and MIN. The reason for this is that it takes longer to calculate Eq. (3) (the “onlooker” UM of ABC) than Eq. (6) (the UM of iSSO). Hence, the UM in iSSO is more efficient than that of ABC.
2. FV: iSSO is superior to ABC in AVG, MAX, and STDEV, but ABC has better MIN in T_2 , T_3 , and T_5 . Moreover, the values of iSSO/ABC also tend to increase; hence, iSSO is more effective and robust than ABC on average, even in the worst case.

Hence, the solutions obtained from iSSO are more effective and robust than those of ABC, and the UMs in iSSO are also more efficient than those in ABC.

5.2. The nonparametric Mann Whitney test

Statistical analysis is always a powerful and useful tool for analyzing experiment results. Table 5 lists all related $\Delta(FV)$, where $\Delta(FV)$ denotes the average fitness values of the 30 final $gbest$ of ABC minus those of iSSO for each benchmark. From Table 5, the values of $\Delta(FV)$ do not form a normal distribution because many

Table 4

The number of better related values obtained from ABC and iSSO (the best related value is in bold).

Statistic	Time interval	FE			FV		
		iSSO	ABC	tie	iSSO	ABC	tie
AVG	T_1	6	44	0	24	12	14
	T_2	3	47	0	23	13	14
	T_3	2	48	0	26	10	14
	T_4	2	48	0	25	7	18
	T_5	1	49	0	16	9	25
MAX	T_1	6	44	0	23	13	14
	T_2	3	47	0	22	14	14
	T_3	1	49	0	26	10	14
	T_4	1	49	0	27	5	18
	T_5	1	49	0	16	9	25
MIN	T_1	6	44	0	16	14	20
	T_2	3	47	0	16	13	21
	T_3	2	48	0	15	11	24
	T_4	2	48	0	15	11	24
	T_5	2	48	0	16	8	26
STDEV	T_1	16	34	0	19	14	17
	T_2	15	35	0	19	14	17
	T_3	12	38	0	22	11	17
	T_4	20	30	0	23	6	21
	T_5	17	33	0	12	10	28

Table 5Statistical nonparametric Wilcoxon–Mann–Whitney test on $\Delta(FV)$.

ID	T1	T2	T3	T4	T5
1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
2	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
3	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
4	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
5	0.01097346*	0.00497643*	0.00413408*	0.00576577*	0.00417233*
6	0.00000017	0.00000013	0.00000007	0.00000000	0.00000000
7	0.04257613*	0.07100830*	0.03785980*	0.08102957*	0.00000000
8	0.00000000	0.00000000	0.00000000	0.00000017	0.00000000
9	0.12248987*	0.12744640*	0.12695574*	0.08842932*	0.01897789*
10	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
11	–0.00008874#	–0.00000023#	0.00000000	0.00000000	0.00000000
12	0.00002897	0.00000003	0.00000000	0.00000000	0.00000000
13	0.00179402	–0.00325316#	–0.00320806#	–0.00295840#	–0.00258068#
14	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
15	–0.00568260#	–0.00000329#	0.00000000	0.00000000	0.00000000
16	–0.26824253#	–0.25205709	0.02479065	0.00313598	–0.00095592
17	–0.00552204#	–0.00030762#	–0.00002902#	–0.00000327#	–0.00000038#
18	0.00000016*	0.00000016*	0.00000016*	0.00000016*	0.00000016*
19	–0.00000036#	–0.00000036#	–0.00000036#	–0.00000036#	–0.00000036#
20	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
21	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
22	–0.06644289#	–0.00000042#	0.00000000	0.00000000	0.00000000
23	10.67626112	0.00039935*	0.00040514*	0.00040517*	0.00040517*
24	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
25	0.00000018*	0.00000018*	0.00000018*	0.00000018*	0.00000018*
26	–0.00000028#	–0.00000028#	–0.00000028#	–0.00000028#	–0.00000028#
27	0.00000000	0.00000390	0.00039333*	0.00086790*	0.00000000
28	0.00000045*	0.00000045*	0.00000045*	0.00000045*	0.00000045*
29	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
30	0.00000000	0.00000000	0.00000020	0.00000000	0.00000000
31	0.00000183*	0.00000183*	0.00000183*	0.00000183*	0.00000183*
32	0.94766920*	0.90246687*	1.91248317*	0.90352647*	0.00000000
33	0.00013563*	0.00016709*	0.00015319	0.00014957*	0.00004428*
34	–0.00000032#	–0.00000032#	0.00364198	–0.00000025#	–0.00000032#
35	0.00000017*	0.00001510*	0.00000194	0.17586364*	0.00000017*
36	0.00456363*	0.00007360	0.00093097	0.00062090	0.00000000
37	0.56252259*	0.38930486*	0.26936931*	1.10254180*	0.01657989*
38	0.03635879*	0.03268729*	0.04290425	0.03797471*	0.00352764*
39	0.00000002*	0.00000002*	0.00000002*	0.00000002*	0.00000002*
40	0.00000001*	0.00000001*	0.00000001	0.00000001*	0.00000001*
41	–0.00001699#	–0.00000032#	–0.00000001*	0.00000000	0.00000000
42	–0.00522117#	–0.00036626#	–0.00002741	0.00006660*	–0.00000011#
43	–0.00005224#	–0.00000288#	–0.00000027	–0.00000003#	0.00000000
44	–0.00083896#	–0.00004146#	–0.00000424	–0.00000036#	–0.00000005#
45	0.00000013	0.00000000	0.00000010*	0.000000240	0.00000000
46	0.07712925*	0.04256432*	0.08079045	0.05438112*	0.00000022*
47	0.10441587	0.12919304	0.17383109	0.19963753*	–0.02332882#
48	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
49	0.01815883	0.02991696*	0.01450485#	0.04763507*	0.00202292*
50	124.72583350*	23.14546301	2.61127768*	0.80672547	0.44084227
Total of*	17	17	19	20	15
Total of#	12	12	8	7	9

: No significant difference between ABC and iSSO.

iSSO not significantly outperforms ABC at critical value 0.05.

* iSSO significantly outperforms ABC at critical value 0.05.

values of the FV are zero. Hence, a distribution-free test called the nonparametric Mann Whitney test [18] is used to test $\Delta(FV)$ to check whether there is a significant difference between the mean FVs obtained from ABC and iSSO at a critical value of 0.05 for T1–T5.

The statistical nonparametric Mann Whitney test results are shown in Table 5, where the superscript “*” means that the mean FV for the related benchmark function obtained from iSSO is significantly better than that of ABC. In contrast, the superscript “#” means that the mean FV obtained from ABC significantly outperforms that of iSSO. The performance of each method can be analyzed using a maximum possible improvement (MPI) calculated by $(\text{the number that iSSO is significant better than that of ABC in minus the number that ABC is significant better than that of iSSO in}) / (\text{the number that ABC is significant better than that of iSSO in}) \times 100\%$.

Generally, the proposed iSSO is better than ABC in FV for each time period. The detailed comparisons are discussed as follows.

1. In both T1 and T2: There are 17 items marked by * and 12 items marked by #, i.e., MPI is $(17/12-1) = 41.7\%$. Hence, iSSO is better than ABC at the critical value 0.05 in the earlier stages.
2. In T3: The number of items marked by * is increased from 17 in T1 and T2 to 19. In contrast, the number of items marked by # is decreased from 12 in T1 and T2 to 8 at the critical value 0.05. Hence, MPI is $(19/8-1) = 137.5\%$. Thus, compared with the solutions found by ABC, the solutions found by iSSO have relatively more significant improvement and are better than those of ABC in T3.
3. In T4: The number of items marked by * is increased from 19 in T3 to 20, and the number of items marked by # is decreased from 8 in T3 to 7. In this time period, the MPI is

$(20/7-1) = 185.7\%$, which is the highest among all periods. Hence, the overall FV of iSSO is much superior to that of ABC, and the solution quality is also further increased at the critical value 0.05.

4. In T5: The number of the items marked by * and # are decreased to 15 (from 20) and increased to 9 (from 7) at the critical value 0.05, respectively, i.e., the MPI is decreased to 66.7%. The reason is that ABC can finally obtain optimums for these problems that cannot be solved by ABC before T5. However, the solution quality of iSSO is still comparatively better than that found by ABC. Additionally, from the above, iSSO has a higher chance of obtaining the optimums more frequently and earlier than ABC.

From the nonparametric Mann Whitney test, in general, iSSO exhibits better performance than ABC in solution quality.

6. Conclusion and future work

This work proposed a continuous version of SSO with a new UM to enhance the ability of the traditional SSO to solve continuous problems, which must use continuous variables to yield better results. The UM is always the major part of soft commuting. Hence, this paper presents significant and novel modifications to SSO.

A comprehensive comparative study on the performances of the proposed iSSO and ABC together with UM_a and three ILS strategies has been made. The solution quality of the proposed iSSO is clearly superior to and more robust than that of ABC, which may be the best-known SC for solving numerical functions. Thus, the proposed iSSO based on UM_a has the ability to avoid local traps and can be efficiently used to solve larger-scale, multivariable, and multimodal functions. SSO has been used more than the other existing methods in supply chain management, scheduling, redundancy allocation problems, forecasting, data mining, etc. In future research, we will focus on strengthening SSO performance; we will also apply the resulting method to different optimization problems and attempt to solve real-life practical engineering problems with more variables or larger-scale benchmark problems. Additionally, in the present work, the proposed method is only compared with ABC. The iSSO method will be further compared with more state-of-the-art metaheuristic methods, such as the Firefly algorithm [41], Biogeography-based Optimization [29], Cuckoo search [42], Bat Algorithm [43], and Krill Herd [11], in the future. Furthermore, mathematical analysis using a dynamic system, such as Markov chains, is going to prove and explain the convergence of the proposed method in future research.

Acknowledgments

I wish to thank the anonymous editor and reviewers for their constructive comments and recommendations, which have significantly improved the presentation of this paper. This research was supported in part by the National Science Council of Taiwan, ROC under Grant NSC101-2221-E-007-079-MY3.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.knosys.2015.02.022>.

References

- [1] R. Azizipناه-Abarghoee, T. Niknam, M. Gharibzadeh, F. Golestaneh, Robust, fast and optimal solution of practical economic dispatch by a new enhanced

- gradient-based simplified swarm optimisation algorithm, *Gener. Transm. Distrib.* 7 (2013) 620–635.
- [2] R. Azizipناه-Abarghoee, A new hybrid bacterial foraging and simplified swarm optimization algorithm for practical optimal dynamic load dispatch, *Int. J. Electr. Power Energy Syst.* 49 (2013) 414–429.
- [3] G. Beni, J. Wang, *Cellular Robotic Systems*, 1989. <http://en.wikipedia.org/wiki/Swarm_intelligence>.
- [4] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- [5] P.C. Chang, X. He, Macroscopic Indeterminacy Swarm Optimization (MISO) algorithm for real-parameter search, in: *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC2014)*, Beijing, China, 2014, pp. 1571–1578.
- [6] C.H. Chou, C.L. Huang, P.C. Chang, A RFID network design methodology for decision problem in health care, in: *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC2014)*, Beijing, China, 2014, pp. 1586–1592.
- [7] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evol. Comput.* 1 (1997) 53–66.
- [8] N. Esfandiari, M.R. Babavalian, A.E. Moghadam, V.K. Tabar, Knowledge discovery in medicine: current issue and future trend, *Exp. Syst. Appl.* 41 (2014) 4434–4463.
- [9] B. Farley, W.A. Clark, Simulation of self-organizing systems by digital computer, *IRE Trans. Inform. Theory* 4 (1954) 76–84.
- [10] A.S. Fraser, Simulation of genetic systems by automatic digital computers. I. Introduction, *Austr. J. Biol. Sci.* 10 (1957) 484–491.
- [11] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, *Commun. Nonlinear Sci. Numer. Simul.* 17 (2012) 4831–4845.
- [12] Y. Gao, G. Zhang, J. Lu, H.M. Wei, Particle swarm optimization for bi-level pricing problems in supply chains, *J. Global Optim.* 51 (2011) 245–254.
- [13] F. Glover, Future paths for integer programming and links to artificial intelligence, *Comput. Oper. Res.* 13 (1986) 533–549.
- [14] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Professional, Reading, MA, 1989.
- [15] L. Guo, G.G. Wang, A.H. Gandomi, A.H. Alavi, H. Duan, A new improved krill herd algorithm for global numerical optimization, *Neurocomputing* 138 (2014) 392–402.
- [16] T.J. Hsieh, W.C. Yeh, Knowledge discovery employing grid scheme least squares support vector machines based on orthogonal design bee colony algorithm, *IEEE Trans. Syst. Man Cybernet. Part B: Cybernet.* 41 (2011) 1198–1212.
- [17] T.J. Hsieh, W.C. Yeh, Penalty guided bees search for redundancy allocation problem with a mix of components in series-parallel systems, *Comput. Oper. Res.* 39 (2012) 2688–2704.
- [18] M. Hollander, D.A. Wolfe, *Nonparametric Statistical Methods*, John Wiley & Sons, Inc., 1999.
- [19] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report TR06, Engineering Faculty, Erciyes University, 2005.
- [20] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, *Appl. Math. Comput.* 214 (2009) 108–132.
- [21] J. Kennedy, R.C. Eberhard, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, Publishing, Piscataway, NJ, USA, 1995, pp. 1942–1948.
- [22] J. Kennedy, R.C. Eberhard, Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, CA, 2001.
- [23] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
- [24] P.L.J.A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer, Boston MA, 2002.
- [25] W. McCulloch, W. Pitts, A logical calculus of ideas immanent in nervous activity, *Bull. Math. Biophys.* 5 (1943) 115–133.
- [26] H. Quan, D. Srinivasan, A. Khosravi, Short-term load and wind power forecasting using neural network-based prediction intervals, *IEEE Trans. Neural Networks Learn. Syst.* 25 (2014) 303–315.
- [27] F. Rossi, D. Velázquez, I. Monedero, F. Biscarri, Artificial neural networks and physical modeling for determination of baseline consumption of CHP plants, *Exp. Syst. Appl.* 41 (2014) 4568–4669.
- [28] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: *Proceedings of the 1999 Congress on Evolutionary Computation*, Publishing, Washington, DC, 1999.
- [29] D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.* 12 (2008) 702–713.
- [30] R. Soltani, Reliability optimization of binary state non-repairable systems: a state of the art survey, *Int. J. Ind. Eng. Comput.* 5 (2014) 339–364.
- [31] R. Storn, On the usage of differential evolution for function optimization, in: *Proceedings of the 1996 Biennial Conference of the North American Fuzzy Information Processing Society*, Publishing, 1996, pp. 519–523.
- [32] R. Storn, K. Price, Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (1997) 341–359.
- [33] V. Tereshko, A. Loengarov, Collective decision-making in honey bee foraging dynamics, *Comput. Inform. Syst.* 9 (2005) 1–7.
- [34] G. Wang, L. Guo, H. Wang, H. Duan, L. Liu, J. Li, Incorporating mutation scheme into krill herd algorithm for global numerical optimization, *Neural Comput. Appl.* 24 (3–4) (2014) 853–871.
- [35] G.G. Wang, A.H. Gandomi, A.H. Alavi, An effective krill herd algorithm with migration operator in biogeography-based optimization, *Appl. Math. Model.* 38 (9–10) (2014) 2454–2462.

- [36] G.G. Wang, L. Guo, A.H. Gandomi, G.S. Hao, H. Wang, Chaotic krill herd algorithm, *Inform. Sci.* 274 (2014) 17–34.
- [37] G.G. Wang, A.H. Gandomi, A.H. Alavi, Stud krill herd algorithm, *Neurocomputing* 128 (2014) 363–370.
- [38] G.G. Wang, A.H. Gandomi, A.H. Alavi, G.S. Hao, Hybrid krill herd algorithm with differential evolution for global numerical optimization, *Neural Comput. Appl.* 25 (2) (2014) 297–308.
- [39] G.G. Wang, A.H. Gandomi, X. Zhao, H.C.E. Chu, Hybridizing harmony search algorithm with cuckoo search for global numerical optimization, *Soft Comput.* (2014), <http://dx.doi.org/10.1007/s00500-014-1502-7>.
- [40] G.G. Wang, A.H. Gandomi, X.S. Yang, A novel improved accelerated particle swarm optimization algorithm for global numerical optimization, *Eng. Comput.* 31 (7) (2014) 1198–1220.
- [41] X.S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Frome, 2008.
- [42] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, IEEE Publications, 2009, pp. 210–214.
- [43] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: J.R. Gonzalez et al. (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)*, *Studies in Computational Intelligence*, vol. 284, Springer, Berlin, 2010, pp. 65–74.
- [44] W.C. Yeh, A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems, *Exp. Syst. Appl.* 36 (2009) 9192–9200.
- [45] W.C. Yeh, Novel swarm optimization for mining classification rules on thyroid gland data, *Inform. Sci.* 197 (2012) 65–76.
- [46] W.C. Yeh, Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization, *IEEE Trans. Syst. Man Cybernet. – Part A: Syst. Humans* 42 (2012) 250–261.
- [47] W.C. Yeh, Simplified swarm optimization in disassembly sequencing problems with learning effects, *Comput. Oper. Res.* 39 (2012) 2168–2177.
- [48] W.C. Yeh, A new parameter-free simplified swarm optimization for artificial neural network training and its application in prediction of time-series, *IEEE Trans. Neural Networks Learn. Syst.* 24 (2013) 661–665.
- [49] W.C. Yeh, Orthogonal simplified swarm optimization for the series-parallel redundancy allocation problem with a mix of components, *Knowl.-Based Syst.* 64 (2014) 1–12.
- [50] W.C. Yeh, Y.M. Yeh, P.C. Chang, Y.C. Ke, V. Chung, Forecasting wind power in the Mai Liao Wind Farm based on the multi-layer perceptron artificial neural network model with improved simplified swarm optimization, *Int. J. Electr. Power Energy Syst.* 55 (2014) 741–748.
- [51] L.A. Zadeh, Fuzzy logic, neural networks, and soft computing, *Commun. ACM* 37 (1994) 77–84.