Journal of Universal Computer Science, vol. 23, no. 7 (2017), 619-635
submitted: 31/3/17, accepted: 30/6/17, appeared: 28/7/17 © *J.UCS*

A Hybrid Social Spider Optimization Algorithm with Differential Evolution for Global Optimization

Jianfeng Qiu
(Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education and Institute of Bioinspired Intelligence and Mining Knowledge and School of Computer Science and Technology Anhui University, Hefei, 230039, China
qiujianf@ahu.edu.cn)

Juan Xie
(School of Mathematics and Physics, Anhui Jianzhu University, Hefei 230601, China
xiejuan0176@163.com)

Fan Cheng
(Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education and Institute of Bioinspired Intelligence and Mining Knowledge and School of Computer Science and Technology Anhui University, Hefei, 230039, China
chengfan@mail.ustc.edu.cn)

Xuefeng Zhang
(School of Computer Science and Technology, Anhui University, Hefei, 230039, China
1458799127@qq.com)

Lei Zhang
(Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education and Institute of Bioinspired Intelligence and Mining Knowledge and School of Computer Science and Technology Anhui University, Hefei, 230039, China
zl@ahu.edu.cn (* Corresponding author))

Abstract: Social Spider Optimization (SSO) algorithm is a swarm intelligence optimization algorithm based on the mating behavior of social spiders. Numerical simulation results have shown that SSO outperformed some classical swarm intelligence algorithms such as Particle Swarm Optimization (PSO) algorithm and Artificial Bee Colony (ABC) algorithm and so on. However, there are still some deficiencies about SSO algorithm, such as the poor balance between exploration and exploitation. To this end, an improved SSO algorithm named wDESSO is proposed for global optimization, which can balance exploration and exploitation effectively. Specifically, a weighting factor changing with iteration is introduced to control and adjust the search scope of SSO algorithm dynamically. After social-spiders have completed their search, a mutation operator is then suggested for jumping out of the potential local optimization, thus can further strengthen the ability of global search. The experimental results on a set of standard benchmark functions demonstrate the effectiveness of wDESSO in solving complex numerical optimization problems.

Key Words: social-spider algorithm; swarm intelligence algorithm; global optimization; weighting factor.

Category: I.2, I.2.2, I.2.8

1 Introduction

Many real-world engineering problems, such as bioinformatics [Zeng et al, 2016, Zou et al, 2016, Zeng and Zhang et al, 2016], economics [Gao et al, 2012] and management science [Kulkarni and Venayagamoorthy, 2011] are transformed to find a potential optimal solution in feasible search space by maximizing or minimizing one objective. Among the existing methods, evolutionary computation has become a promising tool for finding near global optimum in complex solution space [Ju et al, 2016]. Evolutionary computation usually includes two important computing models [James and Victor, 2015]: evolutionary algorithms(EA) and swarm intelligence based algorithms. Specifically, in EA, each individual is mutated and recombined with some probabilities. After that, tournament selection is adopted for comparing arbitrary two individuals from the current population, then the individuals with better fitness value are put into the mating pool. Several typical EAs including genetic algorithm (GA) [Holland, 1992], evolutionary strategy (ES) [Beyer et al, 2002] and differential evolution (DE) [Storn and Price, 1997] have been widely applied for solving multi-objective optimization [Zhang and Tian et al, 2015, Zhang et al, 2014, Zhang and Tian et al, 2016], high dimensional [Cheng and Jin et al, 2015] and non-convex optimization problems [Zhang et al, 2016, Cheng et al, 2016], overlapping community detection [Zhang and Pan et al, 2017], pattern recommendation based on multi-objective optimization [Zhang et al, 2017], etc.

The other evolutionary computation model is swarm intelligence, which has attracted more attention due to its particular mechanism. Bonabeau has defined the swarm intelligence as “any attempt to design algorithms or distributed problem solving devices inspired by the collective behavior of social insect colonies and other animal societies” [Bonabeau et al, 1999]. The term ‘swarm’ is employed to denote a group of unintelligent individuals, which complies a few simple rules and responds to local stimuli individually. A large number of individuals in the same population can perform complicated tasks by cooperative ways (self-organization and labor division) among the individuals, which can well adapt to the changes of surrounding environment. Recently, some swarm intelligence based algorithms and computing model are developed including ants [Dorigo, 1992], birds [Eberhart and Kennedy, 1995], bacterial foraging [Passino, 2002], bees [Karaboga, 2005], probe machine [Xu, 2016] and so on.

Among these algorithms, a novel swarm intelligence algorithm inspired by the unique mating behavior of social spider has been proposed by Erik Cuevas [Cuevas et al, 2016], named social spider optimization (SSO) algorithm. In SSO, the optimal solution is found by cooperation and mating among spiders. One key characteristic of SSO is that the gender difference is incorporated into labor division so as to adapt well itself to the changing of external and internal environment. Due to cooperating and mating behaviors in SSO, population diversity has been increased for most of optimization problems. However, the diversity of the population may decrease with the iterations increase in some multi-modal optimization problems. In other words, the balance between exploration and exploitation in search strategy of SSO needs to be further studied.

In this paper, we propose a novel algorithm, named Weighted Differential Evolution Social Spider Optimization (wDESSO), on basis of SSO algorithm for global optimization. To be specific, in the proposed algorithm, an adaptive weighting factor is firstly introduced to control the movement of spiders, which can guarantee good balance between exploration and exploitation in search strategy; secondly, after completing the preliminary search, one mutation operator is adopted for improving the convergence speed and avoiding premature phenomena of SSO algorithm. Finally, the proposed algorithm wDESSO is compared with the state-of-art optimization algorithms, and the experimental results on 15 standard benchmark functions demonstrate the effectiveness of wDESSO in solving complex numerical optimization problems.

The rest of the paper is organized as follows: some related works and SSO algorithm are reviewed in Section 2. In Section 3, the proposed wDESSO algorithm is illustrated in detail. Section 4 presents a series of experiments on some well known benchmark functions. Finally, some conclusions and future work are listed in Section 5.

2 Background

2.1 Related Works

Evolutionary computation is usually classified into two categories: evolutionary algorithms(EA) and swarm intelligence based algorithms. Among these methods, DE, PSO [Eberhart and Kennedy, 1995], ABC [Karaboga, 2005], and SSO algorithms are the most popular evolutionary computation methods.

In DE algorithm, the offspring is produced by competing with its parents. One notable feature in DE is that all solutions have equal chance of being selected as parents, thus it can be successfully applied into some research fields, such as numerical optimization [Qin et al, 2005] and pattern recognition [Maulik and Saha, 2009]. From then on, the researches dedicated to improve the performance of DE algorithm. In summary, these improved DE algorithms mainly adopted different mutation strategies (e.g. DE/best, DE/rand, DE/current-to-best) [Price. et al, 2005], adaptive strategies (e.g. SDE, JDE, SaDE) [Qin et al, 2009] and hybrid DE algorithm with other optimization algorithms [Biswas et al, 2007]. These achievements have greatly improved the performance of DE algorithm and broadened the application domains of DE in different complex optimization problems.

For swarm intelligence based algorithms, it is important to make the balance between exploration and exploitation. PSO and ABC algorithms are two well-known two swarm intelligence algorithms for solving complex optimization problems in real world. For PSO, after initialization, the positions of particles are changed in the search space in order to find optimal value. More specifically, each particle is changed by two factors: one is its own best position found so far (local best position) and the other is the best position in the whole search space (global best position). The performance of PSO algorithm largely depends on the setting of the initial parameters which make the algorithm easily trap into local optimal and result in premature convergence [Beak et al, 2013]. ABC algorithm adopts the collective intelligent behavior among bees through self-organization and labor division mechanisms. In other words, the whole bee colony are composed of employed bees, onlooker bees and scout bees. The global optimal position is found by cooperating among them. The adopted search strategy is in favor of enlarging the search scope at the initial stage for finding potential better solutions. However, it can easily be trapped into stagnation and slower the convergence speed in the later phase [Banharnsakun et al, 2011]. Moreover, some other optimization algorithms based on swarm intelligence have been widely studied. For example, Dorigo proposed ant colony optimization (ACO) algorithm [Dorigo, 1992] inspired by foraging behavior of ant colony and a Krill Herd (KH) algorithm based on the collective behavior of kills were proposed by Gandomi [Gandomi and Alavi, 2012].

Recently, Erik Cuevas proposed a new swarm intelligence algorithm, named social spider optimization (SSO) algorithm, inspired by the unique mating behavior of social spider [Cuevas et al, 2016]. In SSO, the optimal solution is determined by the way of cooperating and mating among spiders and the gender difference is incorporated into labor division, thus adapt well itself to the change of external and internal environment. However, in some multi-modal optimization problems, the diversity of the population may decrease with the increasing iterations which weaken the ability of jumping out of local optimal. Based on the above consideration, in this paper, by combining the advantages of DE and SSO, we proposed an improved SSO algorithm anmed wDESSO for global optimization, which can balance exploration and exploitation during optimizing.

2.2 Social-Spider Optimization Algorithm

Usually, in Social-Spider Optimization Algorithm, the population is highly female-biased and the number of female spiders reach 70% of the total colony members [Aviles, 1997]. One important characteristic of the SSO is that gender differences have been incorporated into the algorithm, which is beneficial for enlarging the population diversity and enhancing the searching capacities. The interaction between them is encoded by vibration through the communal web, which is a medium of communication. The vibration is determined by two factors: one is the distance between two spiders, and the other is the weight assigned for each spider. The social spiders receive the information transmitted by vibration and are guided to move the global optimal.

For female spiders, the movement is controlled by the vibration of other individuals around her in the communal web. While for the male spiders, they are divided into dominant and non-dominant groups. The dominant male spiders have better fitness

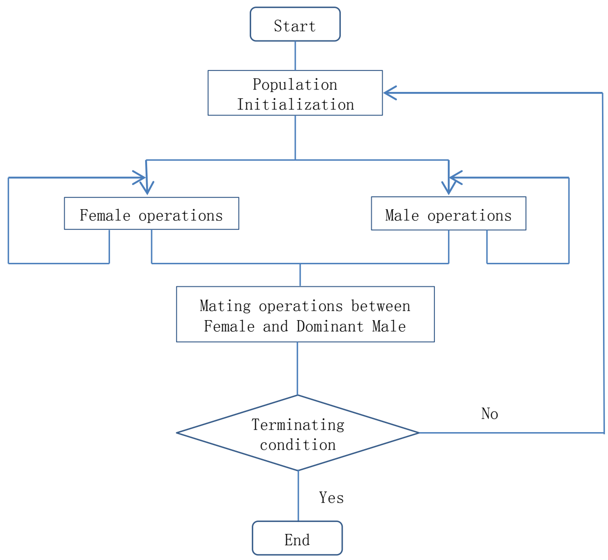

Figure 1: Schematic drawing of SSO algorithm.

values than other male spiders. Dominant males can change their positions considering the influence of the closest females while non-dominant spiders are attracted to the weighted mean of the male population which is beneficial to avoid immature convergence. After that, the mating behavior occurring between dominant males and female members is used to produce offsprings, which plays an important role in SSO algorithm. The advantage of SSO is that the whole population can find potential optimal solutions effectively by the information exchange among them. The framework of SSO algorithm is shown in Figure 1.

3 Proposed algorithm

For SSO algorithm, the movement of the spiders depends on the influence of the individual local best position and the global best position in the population. This could result in spiders being trapped into some local best position and a slower convergence speed. In order to address these issues, we introduce an adaptive weighting factor for enhancing the optimization performance and a mutation operator for avoiding premature phenomena in the framework of SSO algorithm.

3.1 Weighting factor

It is obvious that the positions of the spiders in the current iteration will have a great influence on the next positions of the spiders, whether female spiders or male spiders. In our study, a weighting factor W is first introduced into the SSO algorithm which is used to describe the impact of the current position of the spider. The improved search strategies are illustrated as follows:

$$F_i^{k+1} = W \cdot F_i^k \pm \alpha \cdot Vibc_i \cdot (s_c - F_i^k) \pm \beta \cdot Vibb_i \cdot (s_b - F_i^k) \tag{1}$$

$$M_i^{k+1} = W \cdot M_i^k + \alpha \cdot Vibf_i \cdot (s_f - M_i^k) + rand \tag{2}$$

$$Vibc_i = w \cdot e^{-d_{i,c}^2} \tag{3}$$

$$Vibf_i = w \cdot e^{-d_{i,f}^2} \tag{4}$$

where α and β are random number between $[0, 1]$, k denotes the number of iteration, s_c and s_b represent local and global optimal position at the current generation respectively, whereas $Vibc_i$ and $Vibb_i$ represent the transmitted information from s_c and s_b by vibrating on the common web respectively. The exploration and exploitation of SSO can be balanced effectively by adjusting the size of the parameter W. The greater value of W means more favorable for exploration. A qualitative analysis is made about the parameter W:

1. $W = 0$: In this case, the spiders in the colony will move to the new positions which is determined by local best position s_c and global best s_b so far, while for the spider who locate itself on the global best position, its position will be remain unchanged.

2. $W \neq 0$: The update of the positions of the spiders will be affected by the current one. Thus, for exploring the potential promising area, we may attempt to assign a dynamic weighting factor W to balance exploration and exploitation well.

For some fuzzy, dynamic and complex of optimization problems, it is beneficial to assign a larger value to the parameter W at the initial stage to explore the promising area in the search space while a small value is given at the later for more sophisticated exploitation. Based on the above considerations, in this paper, a weighting factor changing with iteration is presented as follows:

$$W = W_{max} - \frac{iter}{Maxiter}(W_{max} - W_{min}) \tag{5}$$

Here, the W_{max} and W_{min} are maximum and minimum values of the weighting factor W, respectively.

3.2 Mutation strategy

Usually, the introduction of the mutation strategy is helpful for finding more promising solutions in local area of the current solution and enhancing the performance of algorithm. Here, one adaptive mutation scheme with a mutation probability (μ) is proposed to accelerate the convergence of SSO algorithm, which is described as follows:

$$x_{i,G} = \begin{cases} x_{gbest,G} + \gamma(x_{p,G} - x_{q,G}), rand \leq \mu \\ x_{i,G} \end{cases} \tag{6}$$

After completing the search implemented by mating operation between female spiders and male spiders, the mutation operator is then applied for further fine exploiting around the obtained solution with probability μ. The search process is repeated until the predefined stopping criterion is met. The complete pseudo-code for wDESSO algorithm can be found in Algorithm 1.

Algorithm 1 the pseudo-code for wDESSO

Input: The population P, The population size $\mathcal{SN}$, maximum number of generations $Maxiter$, the maximum and minimum of the weighting factor $\{W_{max}, W_{min}\}$, mutation probability p_m, threshold value PF;
Output: The final optimal solution X.
Step1: Population initialization.
$P \leftarrow Intial(\mathcal{P}, SN)$//get the initial population P with the size of SN
Step2: Female operation and male operation.
for $iter = 1$ to $Maxiter$ **do**
  for every female spider **do**
    if rand<PF **then**
      $F_i^{k+1} = W \cdot F_i^k + \alpha \cdot Vibc_i \cdot (s_c - F_i^k) + \beta \cdot Vibb_i \cdot (s_b - F_i^k)$
    else
      $F_i^{k+1} = W \cdot F_i^k - \alpha \cdot Vibc_i \cdot (s_c - F_i^k) - \beta \cdot Vibb_i \cdot (s_b - F_i^k)$
    end if
  end for
  for every male spider **do**
    if Dominant male **then**
      $M_i^{k+1} = W \cdot M_i^k + \alpha \cdot Vibf_i \cdot (s_f - M_i^k) + rand$
    else
      the non-dominant male is remained in the weighted mean of the male population Male
    end if
  end for
end for
Step3: Mating operation:
every dominant male mate with group of female spiders selected by roulette method
Step4: Mutation operation:
After female operation, male operation and mating operation, every spider in the colony is mutated by mutation operator according to the following strategy:
$x_{i,G} = \begin{cases} x_{gbest,G} + \gamma(x_{p,G} - x_{q,G}), rand \leq \mu \\ x_{i,G} \end{cases}$

Next, The steps of wDESSO are elaborated as follow. In female operation stage, for each female spider, a random number is produced firstly and compared with a predefined threshold PF to decide which update strategies are adopted. The corresponding pseudo codes are shown in line 3-9 of Algorithm 1. Different from female spiders, male spiders are mainly used to reproduce with females. The male spiders are separated into two different groups based on the size of fitness values. One is dominant group which is attracted by the closest female one on the common web while the other group (non-dominant) usually locate the center of the whole male spiders. Therefore, in the male operation stage, dominant male spiders are moved to the closest female individual s_f and influenced by its vibration $Vibf_i$ while for non-dominant males approach to the center of the whole male spiders shown in line 10-16. In social spiders, mating is an interesting way for realizing information sharing and interaction between female and male spiders. Specifically, for a dominant male, a set of female spiders are selected using a given range r and the roulette method is also used to select new offspring S_{new}. If S_{new} is better than the worst spider S_{worst} in the population, the S_{worst} is replaced by the new offspring S_{new}. After completing the three above stages, a local perturbation(mutation) for every individual is conducted for trying to find new better position, the strategy is defined as in Equation (6).

4 Simulation experiments

4.1 Benchmark Functions and Parameter Settings

In order to evaluate the performance of wDESSO completely, a set of numerical benchmark functions are adopted. The benchmark functions are listed in Table 1, where f_{min} is the global optimum value. A more detailed description about every function is given in [Cuevas et al, 2016]. For each test function, the minimum $f^*(x) = 0$, which x^* is the global optimum solution. In this paper, as suggested in [Zhang et al, 2005], a widely adopted parameter values $W_{max} = 0.9$ and $W_{min} = 0.4$ are also used for wDESSO.

4.2 Performance Comparison with SSO, ABC and PSO

In order to show the effectiveness of wDESSO, we compare it with three representative algorithms SSO, ABC and PSO. To make a fair comparison, the parameters used here are the same as the recommendation in the literature [Cuevas et al, 2016]. The performance of these algorithms are evaluated on 30 dimensional minimization problems for 30 independent runs. In order to make fair analysis, we directly reference the results of SSO, ABC and PSO given by Cuevas [Cuevas et al, 2016] . The detail parameter settings for every algorithm are described in Table 2. Table 3 shows the Average Best-so-far (AB) value and the Standard Deviation (SD) of best-so-far value of the results obtained by wDESSO algorithm, where the best results are marked in boldface.

Table 1: Benchmark functions used in experiments with $f_{min} = 0$

Name	Function	Search range
Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	$-100 \leq x_i \leq 100$
Schwefel 2.22	$f_2 = \sum_{i=1}^{D} \|x_i\| + \sum_{i=1}^{D} \|x_i\|$	$-10 \leq x_i \leq 10$
Schwefel 1.2	$f_3 = \sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	$-100 \leq x_i \leq 100$
Step	$f_4 = \sum_{i=1}^{D}(\lfloor x_i + 0.5 \rfloor)^2$	$-100 \leq x_i \leq 100$
Quartic	$f_5 = \sum_{i=1}^{D} i \cdot x_i^4 + random(0,1)$	$-1.28 \leq x_i \leq 1.28$
Ackley	$f_6 = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})$ $-exp(\frac{1}{D}\sum_{i=1}^{D} cos2\pi x_i) + 20 + e$	$-32 \leq x_i \leq 32$
Griewank	$f_7 = \frac{1}{4000}\sum_{i=1}^{D} i \cdot x_i^2 - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}} + 1)$	$-600 \leq x_i \leq 600$
Levy	$f_8 = 0.1 \times \{sin^2(3\pi x_1)+$ $\sum_{i=1}^{D-1}(x_i - 1)^2 \cdot [1 + sin^2(3\pi x_i + 1)]$ $+(x_D - 1)^2)[1 + sin^2(3\pi x_D)]\}$ $+\sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$-50 \leq x_i \leq 50$
Rastrigin	$f_9 = \sum_{i=1}^{D-1}[x_i^2 + 10cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$
Rosenbrock	$f_{10} = \sum_{i=1}^{D-1}[100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$
Dixon&Price	$f_{11} = (x_i - 1)^2 + \sum_{i=1}^{D}[i \cdot (2x_i^2 - x_{i-1})^2]$	$-10 \leq x_i \leq 10$
Sum of Squares	$f_{12} = (\sum_{i=1}^{D}(i \cdot x_i^2)$	$-10 \leq x_i \leq 10$
Zakharov	$f_{13} = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5 \cdot i \cdot x_i^2)^2$ $+(\sum_{i=1}^{D} 0.5 \cdot i \cdot x_i^2)^4$	$-5 \leq x_i \leq 10$
Powell	$f_{14} = \sum_{i=1}^{D/4}(x_{4i-3} + x_{4i-2})^2 + 5(x_{4i-1} + x_{4i})^2$ $+(x_{4i-2} + x_{4i-1})^4 + 10(x_{4i-3} + x_{4i})^4$	$-5 \leq x_i \leq 10$
Schaffer	$f_{15} = 0.5 + \frac{sin^2(\sqrt{\sum_{i=1}^{D} x_i^2}) - 0.5}{(1+0.001 \cdot (\sum_{i=1}^{D} x_i^2))^2}$	$-100 \leq x_i \leq 100$

Table 2: Parameter settings

PSO	c_1=2, c_2=2; the weight factor decreases linearly from 0.9 to 0.2
ABC	limit=100
SSO	PF=0.7
wDESSO	$\omega_{max} = 0.9$ and $\omega_{min} = 0.4$

According to the results listed in Table 3, wDESSO algorithm has more powerful ability for finding global optimization and significantly better than the results obtained by SSO, ABC and PSO algorithms on most of test problems. The mutation strategy shown in Equation (6) is adopted to update the spider's position by the best-so-far spider and the distance between two randomly selected spiders. Therefore, wDESSO algorithm can find global optimum solution quickly and greatly improve convergence due to the introduction of the mutation strategy. Since wDESSO is influenced by the best-so-far individual, thus leading to the superior performance of wDESSO algorithm especial for the unimodal functions.

In order to be more intuitive to compare the convergency of wDESSO with SSO and ABC algorithms, Figure 2 and Figure 3 make some convergence curves of the best results for these algorithms based on some typical benchmark functions for 30 and 100 dimensionality. For each function, the convergence curves represent the best one of the 30 and 50 independent runs for different dimensionality. Specifically, the parameter setting for the two groups of experiments are as follows: (1) 30 dimensionality: $populationsize = 100$, $Maximumiteration = 1000$; (2) 50 dimensionality: $populationsize = 400$, $Maximumiteration = 3000$;

Table 3: Comparison of wDESSO with SSO, ABC and PSO on 30-Dimensional benchmark functions over 30 independent runs

Function	Measure	wDESSO	SSO	ABC	PSO
f1	AB	**0.00E+000**	1.96E-03	2.90E-03	1.00E+03
	SD	**0.00E+000**	9.96E-04	1.44E-03	3.05E+03
f2	AB	**8.92E-198**	1.37E-02	1.35E-01	5.17E+01
	SD	**0.00E+000**	3.11E-03	8.01E-02	2.02E+01
f3	AB	**1.99E-023**	4.27E-02	1.13E+00	8.63E+04
	SD	**1.09E-022**	3.11E-02	1.57E+00	5.56E+04
f4	AB	**3.84E-007**	2.68E-03	4.06E-03	1.00E+03
	SD	**5.61E-007**	6.05E-04	2.98E-03	3.06E+03
f5	AB	**3.20E-005**	1.20E+01	1.21E+01	1.50E+01
	SD	**2.68E-005**	5.76E-01	9.00E-01	4.75E+00
f6	AB	**8.88E-016**	1.36E-02	6.53E-01	1.14E+01
	SD	**0.00E+000**	2.36E-03	3.09E-01	8.86E+00
f7	AB	**0.00E+000**	3.29E-03	5.22E-02	1.20E+01
	SD	**0.00E+000**	5.49E-04	3.42E-02	3.12E+01
f8	AB	**1.34E-006**	6.92E-05	1.44E-04	2.47E+00
	SD	**2.06E-006**	4.02E-05	1.69E-04	3.27E+00
f9	AB	**0.00E+000**	8.59E+00	2.64E+01	1.35E+02
	SD	**0.00E+000**	1.11E+00	1.06E+01	3.73E+01
f10	AB	**3.13E-003**	1.14E+02	1.38E+02	3.34E+04
	SD	**1.00E-002**	3.90E+01	1.55E+02	4.38E+04
f11	AB	**2.40E-001**	2.14E+00	3.60E+00	3.12E+04
	SD	**7.54E-003**	1.26E+00	3.54E+00	5.74E+04
f12	AB	**0.00E+000**	4.44E-04	1.10E-01	6.93E+02
	SD	**0.00E+000**	2.90E-04	1.98E-01	6.48E+02
f13	AB	**3.04E-024**	6.81E+01	3.12E+02	4.11E+02
	SD	**1.67E-023**	3.00E+01	4.31E+01	1.56E+02
f14	AB	**8.83E-321**	1.87E+00	2.13E+00	1.26E+03
	SD	**0.00E+000**	1.20E+00	1.22E+00	1.12E+03
f15	AB	4.89E+001	**5.39E-05**	1.18E-04	4.27E+07
	SD	3.68E+000	**1.84E-05**	8.88E-05	9.70E+07

From the results shown in Figure 2 and Figure 3, the performances of wDESSO algorithm have better scalability when increasing the dimensionality from 30 to 100.

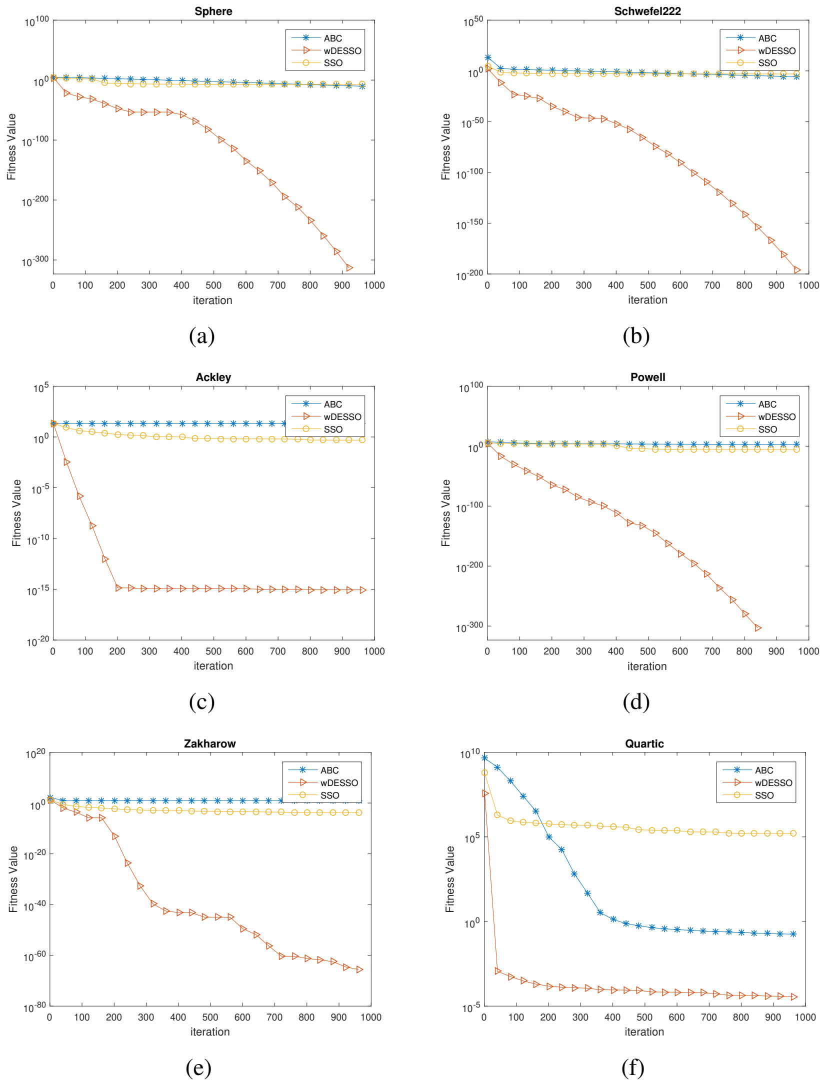

Figure 2: The demonstrations of convergence of wDESSO, SSO, and ABC algorithms on 30 dimensionality

It is noteworthy that due to the complex of some test functions, such as Zakharow, the performance of the algorithms degrade to some extent.

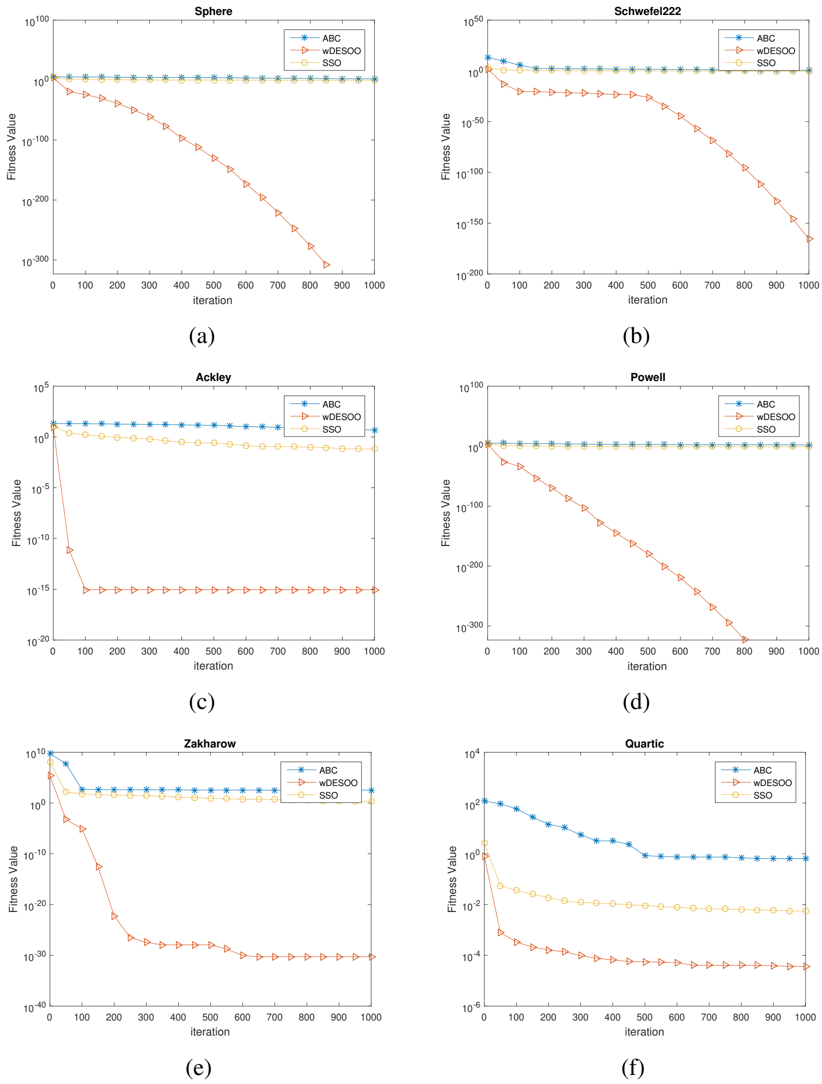

Figure 3: The demonstrations of convergence of wDESSO, SSO, and ABC algorithms on 100 dimensionality

4.3 Comparing wDESSO with Some Improved Swarm Intelligence Algorithms

For making more comprehensive comparison, wDESSO is also compared with some improved swarm intelligence optimization algorithms, including GABC [Zhu and Kwong, 2010], E-ABC [Montes and Koeppel, 2010] and

ABC/best [Gao et al, 2012] using the same benchmark functions including unimodal, multimodal problems on different dimensionality. The population size is 80 and the maximum number of generations is 5000 or until the function error dropped below e-20 (values less than e-20 were reported as 0). Every algorithm is repeated 30 times independently and the means and standard deviations are reported in Table 4. From the results, we can see that the performance of wDESSO are superior than other variants of ABC algorithms under the same experimental conditions. The reason is that for unimodal problems, the weighting factor can accelerate convergence effectively while for multimodal problems, the adopted local mutation strategy can increase the possibilities of escaping from the local optimum.

Table 4: Performance of GABC, E-ABC, ABC/best/1, ABC/best/2, and wDESSO

Algorithm	Measure	Sphere(Uni-modal)		Rosenbrock(Uni-modal)		Griewank(Multi-modal)	
		D=30	D=60	D=2	D=3	D=30	D=60
GABC	Mean	4.17E-16	1.43E-15	1.68E-04	2.65E-03	2.96E-17	7.54E-16
	SD	7.36E-17	1.37E-16	4.42E-04	2.22E-03	4.99E-17	4.12E-16
E-ABC	Mean	1.67E-16	1.41E-15	4.63E-04	1.20E-02	4.90E-14	4.19E-14
	SD	2.70E-16	1.52E-15	4.57E-04	7.06E-03	7.31E-03	9.05E-03
ABC/best/2	Mean	1.70E-126	3.72E-58	4.42E-04	9.90E-04	**0**	**0**
	SD	2.67E-58	2.39E-04	6.92E-04	0	0	3.48E-15
ABC/best/1	Mean	1.1E-150	4.40E-69	4.99E-06	5.52E-06	**0**	**0**
	SD	1.4E-150	2.56E-69	8.22E-06	3.03E-06	**0**	**0**
wDESSO	Mean	**0.00E+00**	**0.00E+00**	**5.70E-07**	**2.27E-06**	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**6.68E-07**	**2.95E-06**	**0.00E+00**	**0.00E+00**
	Measure	Ackley(Multi-modal)		Schaffer(Multi-modal)		Rastrigin(Multi-modal)	
		D=30	D=60	D=2	D=3	D=30	D=60
GABC	Mean	3.21E-14	1.66E-13	0	1.85E-18	1.32E-14	3.52E-13
	SD	3.25E-15	2.21E-14	0	1.01E-17	2.44E-14	1.24E-13
E-ABC	Mean	1.22E-10	1.55E-07	0	2.79E-07	9.97E-15	7.51E-13
	SD	4.86E-11	2.84E-08	0	2.24E-07	3.87E-15	6.15E-13
ABC/best/2	Mean	2.50E-14	7.12E-14	0	3.56E-06	0	0
	SD	3.48E-15	4.14E-15	0	1.27E-06	0	0
ABC/best/1	Mean	1.72E-14	6.62E-14	0	0	0	0
	SD	2.84E-15	1.74E-15	0	0	0	0
wDESSO	Mean	**8.87E-16**	**8.90E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Moreover, we also compare the performance of wDESSO with some other typical variants of DE algorithms including JADE [Zhang et al, 2009], JDE [Brest et al, 2006], SaDE [Qin et al, 2009] on 30 and 100 dimensionality. The parameter settings and experimental results of these algorithms are directly gained from the reference [Zhang et al, 2009] and the results are listed in Table 5 and Table 6. From these results of experiments, we can conclude that wDESSO is a competitive and scalable optimization method for solving complex numerical optimization problems. When increasing the scale of the benchmark functions from 30 to 100 dimensionality, the performance of wDESSO are not obvious influenced which is shown in Table 6 and still superior or near the other algorithms adopted in this experiment.

Table 5: Experimental results of 30-dimensional benchmark functions, averaged over 50 independent runs

Funtions	FEs	Measures	SaDE	JDE	JADE	wDESSO
Sphere	$1.5*10^5$	Mean	4.5E-20	2.5E-28	1.8E-60	**0.00E+000**
		Std	1.9e-14	3.5E-28	8.4E-60	**0.00E+000**
Schwefel 2.22	$2.0*10^5$	Mean	1.9E-14	1.5E-23	1.8E-25	**0.00E+000**
		Std	1.1E-14	1.0E-23	8.8E-25	**0.00E+000**
Schwefel1.2	$5.0*10^5$	Mean	9.0E-37	5.2E-14	5.7E-61	**0.00E+000**
		Std	5.4E-36	1.1E-13	2.7E-60	**0.00E+000**
Rosenbrock	$2.0*10^6$	Mean	1.8E+01	8.0E-02	8.0E-02	**1.92E-005**
		Std	6.7E+00	5.6E-01	5.6E-01	**2.87E-005**
Step	$1.0*10^4$	Mean	9.3E+02	1.0E+03	2.9E+00	**1.82E-005**
		Std	1.8E+02	2.2E+02	1.2E+00	**3.00E-005**
Quartic	$3.0*10^5$	Mean	4.8E-03	3.3E-03	6.4E-04	**6.66E-006**
		Std	1.2E-03	8.5E-04	2.5E-04	**4.39E-006**
Rastrigin	$1.0*10^5$	Mean	1.2E-03	1.5E-04	1.0E-04	**0.00E+000**
		Std	6.5E-04	2.0E-04	6.0E-05	**0.00E+000**
Ackley	$5.0*10^4$	Mean	2.7E-03	3.5E-04	8.2E-10	**8.88E-016**
		Std	5.1E-04	1.0E-04	6.9E-10	**0.00E+000**
Griewank	$5.0*10^4$	Mean	7.8E-04	1.9E-05	9.9E-08	**0.00E+000**
		Std	1.2E-03	5.8E-05	6.0E-07	**0.00E+000**

Table 6: Experimental results of 100-dimensianal benchmark functions, averaged over 50 independent runs

Funtions	FEs	Measures	SaDE	JDE	JADE	wDESSO
Sphere	$8*10^5$	Mean	2.9E-08	5.0E-15	1.2E-48	**0.00E+000**
		Std	3.2E-08	1.7E-15	1.5E-48	**0.00E+000**
Schwefel 2.22	$1.2*10^6$	Mean	1.7E-05	4.1E-15	1.1E-41	**0.00E+000**
		Std	3.8E-06	1.1E-15	5.1E-41	**0.00E+000**
Schwefel1.2	$3.2*10^6$	Mean	2.4E-13	5.4E-02	1.2E-26	**0.00E+000**
		Std	5.2E-13	2.7E-02	2.0E-26	**0.00E+000**
Rosenbrock	$2.4*10^6$	Mean	9.4E+01	7.2E+01	5.6E-01	**3.8E-006**
		Std	4.0E-01	1.1E+01	1.4E+00	**1.3E-005**
Step	$6.0*10^5$	Mean	**0.0E+00**	**0.0E+00**	1.6E-01	2.0E-008
		Std	**0.0E+00**	**0.0E+00**	3.7E-01	3.7E-008
Quartic	$2.4*10^6$	Mean	1.0E-02	8.1E-03	1.1E-03	**8.8E-007**
		Std	4.9E-03	9.0E-04	2.1E-04	**6.5E-007**
Rastrigin	$1.2*10^6$	Mean	9.1E-03	2.1E-04	1.9E-01	**0.00E+000**
		Std	1.8E-03	2.0E-04	3.8E-02	**0.00E+000**
Ackley	$1.2*10^6$	Mean	2.1E-07	9.9E-14	8.9E-15	**8.9E-016**
		Std	1.0E-07	2.0E-14	2.1E-15	**0.00E+000**
Griewank	$1.2*10^6$	Mean	8.6E-13	**0.0E+00**	3.9E-04	**0.00E+000**
		Std	8.2E-13	**0.0E+00**	2.0E-03	**0.00E+000**

5 Conclusions and future work

In this paper, a novel improved social spider swarm intelligence optimization algorithm named wDESSO has been proposed for solving complex global optimization problems. The main idea is to make the use of the weighting factor to enhance the search ability of SSO. In wDESSO, the adaptive weighting factor and mutation strategies have been verified to be beneficial for jumping out of local minimum during the process of optimization.

Specifically, the introduction of the adaptive weighting factor improves the convergency speed and makes the population near to the current optimal solution quickly. In order to avoid trapping the local optimal, a random mutation strategy is adopted for escaping from the stagnation. Experimental results on a set of benchmark functions have demonstrated that the proposed wDESSO significantly outperformed or near to SSO, ABC, PSO, DE and their respective variants. In future, we would like to design some customized search strategies for specific real world problems under the framework of wDESSO to find global optimum solution.

Acknowledgement

This work is supported by National Nature Science Foundation of China (Grant No. 61502001, 61502004), Scientific Research Startup Fund for Doctors of Anhui University, by the Academic and Technology Leader Imported Project of Anhui University (No.J01006057). This work was also supported in part by the Natural Science Foundation of Anhui Province (Grant no.1708085MF166), and Key Program of Natural Science Project of Educational Commission of Anhui Province (Grant no.KJ2013A009, KJ2015A070,KJ2017A013).

References

[Aviles, 1997] Leticia Aviles. *Causes and consequences of cooperation and permanent-sociality in spiders*. 1997.

[Banharnsakun et al, 2011] Anan Banharnsakun, Tiranee Achalakul, and Booncharoen Sirinaovakul. The best-so-far selection in artificial bee colony algorithm. *Applied Soft Computing*, 11(2):2888–2901, 2011.

[Beak et al, 2013] Hyunwook Beak, Tae Hyoung Kim, Jaena Ryu, and Jeill Oh. Diversity-enhanced particle swarm optimizer and its application to optimal flow control of sewer networks. In *Science and Information Conference*, pages 977–978, 2013.

[Beyer et al, 2002] Hans-Georg Beyer and Hans Paul Schwefel. Evolution strategies - a comprehensive introduction. nat comput. *Natural Computing*, 1(1):3–52, 2002.

[Biswas et al, 2007] Arijit Biswas, Sambarta Dasgupta, Swagatam Das, and Ajith Abraham. A synergy of differential evolution and bacterial foraging algorithm for global optimization. *Neural Network World*, 17(6):607–626, 2007.

[Bonabeau et al, 1999] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Inc., 1999.

[Brest et al, 2006] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *Evolutionary Computation IEEE Transactions on*, 10(6):646–657, 2006.

[Cheng and Jin et al, 2015] Ran Cheng and Yaochu Jin. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291(6):43–60, 2015.

[Cheng et al, 2015] Ran Cheng, Yaochu Jin, K Narukawa, and B Sendhoff. A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. *Evolutionary Computation IEEE Transactions on*, XX(1):838–856, 2015.

[Cheng et al, 2016] Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016.

[Cuevas et al, 2016] Erik Cuevas, Miguel Cienfuegos, Zald, Daniel Var, and Marco Rez-Cisneros. A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications An International Journal*, 40(16):6374–6384, 2016.

[Dorigo, 1992] M Dorigo. Optimization, learning and natural algorithms. *Thesis Politecnico Di Milano Italy*, 1992.

[Eberhart and Kennedy, 1995] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *International Symposium on MICRO Machine and Human Science*, pages 39–43, 1995.

[Gandomi and Alavi, 2012] Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12):4831–4845, 2012.

[Gao et al, 2012] Weifeng Gao, Sanyang Liu, and Lingling Huang. A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics*, 236(11):2741–2753, 2012.

[Holland, 1992] John Holland. Genetic algorithms. *Scientific American*, 267(1):66–72, 1992.

[Ju et al, 2016] Y. Ju, S. Zhang, N. Ding, X. Zeng, and X. Zhang. Complex network clustering by a multi-objective evolutionary algorithm based on decomposition and membrane structure. *Scientific Reports*, 6:33870, 2016.

[Karaboga, 2005] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. 2005.

[Kulkarni and Venayagamoorthy, 2011] R. V Kulkarni and G. K Venayagamoorthy. Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems Man and Cybernetics Part C*, 41(2):262–267, 2011.

[Zhang and Pan et al, 2017] Zhang L, Pan H, Su Y, Zhang X, and Niu Y. A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection, In press, 2017.

[Maulik and Saha, 2009] Ujjwal Maulik and Indrajit Saha. Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recognition*, 42(9):2135–2149, 2009.

[Montes and Koeppel, 2010] E Mezura-Montes and R. E Velez-Koeppel. Elitist artificial bee colony for constrained real-parameter optimization. In *Evolutionary Computation*, pages 1–8, 2010.

[Passino, 2002] K.M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, 22(3):52–67, 2002.

[Price. et al, 2005] K. V. Price, R. M. Storn, and J. A. Lampinen. Differential evolution-a practical approach to global optimization. 141(2):1–24, 2005.

[Qin et al, 2009] A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2):398–417, 2009.

[Qin et al, 2005] A. K Qin and P. N Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *IEEE Congress on Evolutionary Computation, Cec 2005, 2-4 September 2005, Edinburgh, Uk*, pages 1785–1791, 2005.

[Storn and Price, 1997] Rainer Storn and Kenneth Price. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[Zhang et al, 2017] Xingyi Zhang, Fuchen Duan, Lei Zhang, Fan Chen, and Yaochu Jin Ke Tang. Pattern recommendation in task-oriented applications: a multi-objective perspective. *IEEE Computational Intelligence Magazine*, 12(3):43–53, 2017.

[Xu, 2016] J. Xu. Probe machine. *IEEE Trans Neural Netw Learn Syst*, 27(7):1405–1416, 2016.

[James and Victor, 2015] James J. Q. Yu and Victor O. K. Li. A social spider algorithm for global optimization. *Applied Soft Computing*, 30(C):614–627, 2015.

[Zeng et al, 2016] X. Zeng, Y. Liao, Y. Liu, and Q. Zou. Prediction and validation of disease genes using hetesim scores. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2016.

[Zeng and Zhang et al, 2016] X. Zeng, X. Zhang, and Q. Zou. Integrative approaches for predicting microrna function and prioritizing disease-related microrna using biological interaction networks. *Briefings in Bioinformatics*, 17(2), 2016.

[Zhang et al, 2009] Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, 2009.

[Zhang and Tian et al, 2015] X Zhang, Y Tian, R Cheng, and Y Jin. An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *Evolutionary Computation IEEE Transactions on*, 19(2):201–213, 2015.

[Zhang et al, 2014] X Zhang, Y Tian, and Y Jin. A knee point-driven evolutionary algorithm for many-objective optimization. *Evolutionary Computation IEEE Transactions on*, 19(6):761–776, 2014.

[Zhang et al, 2016] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin. A decision variable clustering based evolutionary algorithm for large-scale many-objective optimization. 2016.

[Zhang and Tian et al, 2016] Xingyi Zhang, Ye Tian, and Yaochu Jin. Approximate non-dominated sorting for evolutionary many-objective optimization. *Information Sciences*, 369:14–33, 2016.

[Zhang et al, 2005] Xuanping Zhang, Yuping Du, Zheng Qin, Guoqiang Qin, and Jiang Lu. *A Modified Particle Swarm Optimizer*. Springer Berlin Heidelberg, 2005.

[Zhu and Kwong, 2010] Guopu Zhu and Sam Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7):3166–3173, 2010.

[Zou et al, 2016] Quan Zou, Shixiang Wan, Ying Ju, Jijun Tang, and Xiangxiang Zeng. Pretata: predicting tata binding proteins with novel features and dimensionality reduction strategy. *Bmc Systems Biology*, 10(4):401–412, 2016.