Neural Computing and Applications (2020) 32:16783–16794
https://doi.org/10.1007/s00521-018-3905-3

SMART DATA AGGREGATION INSPIRED PARADIGM & APPROACHES IN IOT APPLNS

A novel numerical optimization algorithm inspired from garden balsam

Shengpu Li[1,2] · Yize Sun[3]

Received: 26 September 2018 / Accepted: 16 November 2018 / Published online: 11 December 2018

Abstract
This paper introduces a new evolutionary computing method inspired by the seed transmission process of garden balsam. Garden balsam, a beautiful and attractive flower, randomly ejects the seeds within a certain range by virtue of mechanical force originating from cracking of mature seed pods, which is different from natural expansion of most species of plants. The seeds scattered to suitable growth area will have greater reproductive capacity in the next generation, followed by iteration until the most suitable point for growth in a particular space is eventually found. This phenomenon can more intuitively show the process of searching the problem solution space in the optimization problem. The garden balsam optimization algorithm proposed in this paper incorporates two different types of search processes and has a mechanism to maintain population diversity. Through the optimization experiment on 24 constrained optimization problems, the results obtained by using this algorithm are compared with those of some known meta-heuristic search algorithms. The statistical analysis of the experimental results has been implemented by Friedman rank test and Holm–Sidak test. The comparison results verify the effectiveness of the algorithm.

Keywords Artificial intelligent · Evolutionary computing · Swarm intelligence · Garden balsam optimization algorithm · Function optimization

1 Introduction

Optimization problem runs through all aspects of human activity. Optimization idea is invariably demonstrated from the division of labor in primitive hunting, to the intensive cultivation in agricultural production, and to job scheduling in industrial production [1]. Early optimization mainly relied on empirical analysis. With the improvement in the knowledge level, people began to resort to more accurate mathematical methods to describe and solve optimization problems [2]. Since the twentieth century, new means for optimization has been available thanks to rapid development of electronic computer technology and artificial intelligence technology, enabling people to effectively deal with many complex optimization problems that could not be solved in the past, thus greatly promoting social progress and development [3].

During research on optimization problems, researchers are often inspired by nature [4]. For example, in the evolution of species, genes not adapted to the environment are gradually eliminated, while those adapted to the environment are more likely to be retained to further enhance the competitiveness of species through the optimization of combinations. Inspired by this, Holland proposed genetic algorithms (GA) [5] to solve optimization problems. In the cooling process of metal, each molecule evolves to the lowest energy state possible, making sequence of all molecules changed from disorderly to orderly. Inspired by this, Kirkpatrick et al. [6] proposed simulated annealing algorithm (SA). Based on colony cooperative foraging

✉ Yize Sun
sunyz@dhu.edu.cn

1 College of Information Science and Technology, Donghua University, Shanghai 201620, China

2 College of Information Engineering, Pingdingshan University, Pingdingshan 467002, China

3 College of Mechanical Engineering, Donghua University, Shanghai 201620, China

behavior, Dorigo et al. [7] proposed an ant colony optimization (ACO) algorithm. By simulating the flight patterns of birds, Kennedy and Eberhart [8] proposed a particle swarm optimization (PSO) algorithm. By simulating the self-organizing pattern of bee colonies, Karaboge proposed artificial bee colony (ABC) algorithm [9]. In addition, there are such optimization algorithms as differential evolution (DE) [10], teaching–learning-based optimization (TLBO) [11], invasive weed optimization (IWO) [12] and fireworks algorithm (FWA) [13], bat algorithm (BA) [14] and bacterial foraging optimization (BFO) algorithm [15].

Inspired by the seed transmission mode of garden balsam, the author proposed a new optimization algorithm—garden balsam optimization algorithm. Garden balsam is an annual herbaceous flower with English name “Touch me not” and “Don’t touch me” in American language because its fruit will crack at slight touch and eject seeds. The fruit of garden balsam known as capsule scatters the seeds around the parent through its own mechanical force of cracking at maturity. Garden balsam growing in good growth environment has robust plant, full capsules and forceful cracking, capable of producing more seeds and spreading them to a wider range [16]. This is how garden balsam optimization (GBO) algorithm comes into being. The algorithm though with relatively simple mechanism has been proved to effectively converge to obtain the optimal solution.

The remainder of this paper is organized as follows. Section 2 describes and summarizes the natural reproduction process of garden balsam. Section 3 describes GBO algorithm and its features. Section 4 describes the experimental research on the proposed algorithm and compares it with other meta-heuristic algorithms. Section 5 discusses the statistical analysis of the comparison results. Finally, Sect. 6 sums up the conclusions of this paper.

2 Garden balsam’s natural reproduction process

During the long-term interaction between plants and the environment, seeds and fruits develop a series of mechanisms suitable for transmission. The seed transmission mode and process constitute an important content of evolutionary ecology. Common transmission factors include air, water, animals and self-spread [17], where self-spread means mechanical ejection force is produced to eject seeds after fruit of certain plants matures, dries and cracks. These fruits are found in dehiscent type of dried fruits. Studies have found that second transmission is often a case in seeds relying on self-spread. Garden balsam in this paper belongs to self-spread types.

Whole plant of garden balsam, an annual herb of sapindales, balsam family and Impatiens L., is divided into six parts of roots, stems, leaves, flowers, fruits and seeds. Because its flower head, wings, tail and feet all look like phoenix shape, it is also known as Buttercup. Garden balsam has varied flower colors including pink, red, purple, pinkish purple. By smashing its petals or leaves, wrapping it around nails with leaves, the nail can be died with bright-colored red; thus, it is very popular with girls. The plant, capsule and self-spread of henbane are shown in Fig. 1.

Garden balsam fruit in oval shape is developed from the pistil, which includes a few to more than twenty seeds according to growth state and pollination. According to fruit classification knowledge in botany, the fruit of garden balsam is capsule whose most important feature is that the fruit is formed by two or more carpellate pistils, it cracks in diverse ways, and the fruit is cracked into five circinate carpels which eject seeds by mechanical force for self-spread reproduction [18].

Studies have shown aril to be the key structure for the seed transmission by elasticity. The aril mainly consists of vesicular cells which undergoes serious dehydration and contraction during maturation. The unbalanced contraction between cells produces torsion which gradually accumulates with the maturation. When the critical point is exceeded, the vesicles split and roll over the seed tip, obliquely projecting the seeds in the form of bounce. The relationship diagram between seed quality and ejection shows that, for a farther ejection distance, the seed mass is larger, and the relationship is more obvious in case of dry weight than fresh weight. The seed density distribution increases first and then decreases with the enlarging transmission distance.

The steps in the dispersion process of garden balsam population can be summarized as follows:

1. Population initialization: A few seeds are randomly scattered in a specific area, taking root and producing the first-generation population;
2. Progeny reproduction: Each individual plant in the first-generation population will demonstrate different growth conditions due to the different natural conditions in the growth area. The more robust plants will yield more fruits and then generate more seeds.
3. Mechanical transmission: According to the transmission properties of garden balsam, the seeds will be ejected by mechanical transmission to the surrounding areas of the parent after the fruit matures. The plants in good growth state will have full fruit, more powerful ejection force, and the seeds will be ejected farther.
4. Second transmission: In the real world, individual seeds will be randomly transmitted to other places by

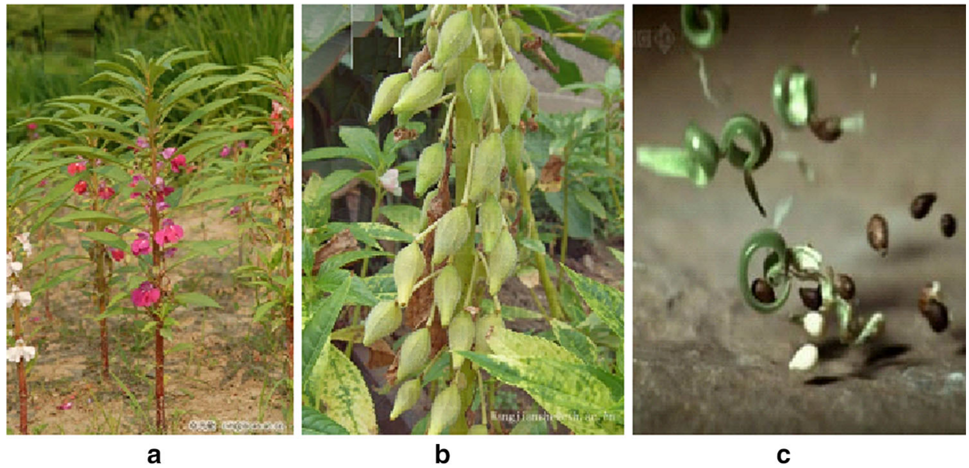

Fig. 1 **a** Plant, **b** capsule and **c** self-spread

the influence of natural forces such as animals, running water and wind to increase the population diversity.

5. Elimination through competition: There is a maximum limit for population size within a specific region. When the population reaches its maximum, individuals with poor fitness will be eliminated in competition within the population.

3 Garden balsam optimization algorithm

The garden balsam optimization algorithm establishes the corresponding mathematical model by simulating the propagation and expansion behavior of garden balsam. A parallel explosive search manner is then formed by introducing random factors and selection strategies, which then develops into a global search method for solving optimal solution to complex optimization problem.

From Fig. 2, it can be seen that the algorithm iterates from the beginning and adopts mechanical propagator, second propagator, mapping rule and selection strategy in turn until the termination condition is satisfied; that is, accuracy requirement of the problem is satisfied or the maximum number of iterations is reached.

3.1 Initialize a population

Determine the initial population number N_{init} and the maximum population size $N_{\max}$, the maximum number of iterations $\mathrm{iter}_{\max}$, the number of problem dimensions D, the upper limit $S_{\max}$ and lower limit $\mathrm{S}_{\min}$ for the possible seed number, nonlinear index n, zoom factor F, initial value A_{init} of seed diffusion amplitude, seed number N_{sec} in second transmission and search space range. The initial population is obtained via uniform distribution to ensure diversity of the initial population of GBO algorithm. The initial population generated by this method can be randomly distributed in the entire search space.

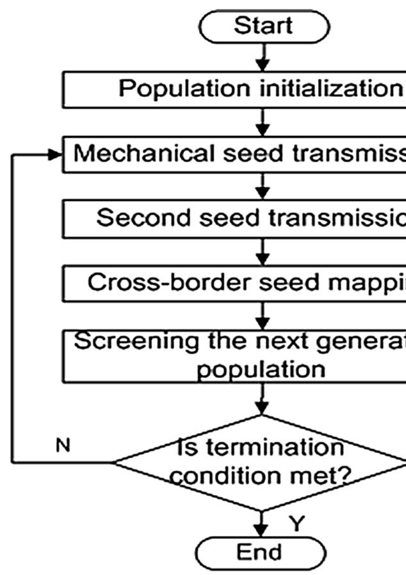

Fig. 2 Framework of garden balsam optimization algorithm

3.2 Mechanical transmission

Seeds can grow into individual plants. Plants in good growth environment (with better fitness function) have robust rhizomes, have full capsules and produce more seeds. The mechanical force is stronger when the capsule cracks at seed maturity, and the seed ejection distance is larger. Meanwhile, consideration should also be given to the balance between early global exploration capabilities and later local exploitation capabilities.

The number of seeds produced by an individual (garden balsam plant) in the reproductive process concerns the individual's fitness value. A better fitness value means more produced seeds. For minimization, let individuals with minimum fitness values reproduce as many offspring as possible, while individuals with the greatest fitness value reproduce as few offspring as possible. That is, let number of seeds produced by an individual with minimum fitness value be $S_{\max}$ and number of seeds produced by an individual with the maximum fitness value be $S_{\min}$, while the number of seeds produced by an individual between the minimum fitness value and the maximum fitness value follows a linear relationship of downward rounding with the fitness value.

The number of seeds produced by individual X_i :

$$S_i = \frac{f_{\max} - f(X_i)}{f_{\max} - f_{\min}} \times (S_{\max} - S_{\min}) + S_{\min} \tag{1}$$

where S_i denotes the number of seeds produced by the i-th plant; $f(X_i)$ denotes the fitness value of the i-th plant, $f_{\max}$ is the maximum fitness value in the current population, $f_{\min}$ is the minimum fitness value in the current population; $S_{\max}$ means the maximum number of seeds produced by garden balsam, $S_{\min}$ means the minimum number of seeds produced by garden balsam.

The calculation expression of the seed diffusion range is as follows:

$$A_i = \left(\frac{\mathrm{iter}_{\max} - \mathrm{iter} + 1}{\mathrm{iter}_{\max}}\right)^n \times \frac{f_{\max} - f(X_i) + 1}{f_{\max} - f_{\min}} \times A_{\mathrm{init}} \tag{2}$$

where iter is the current number of evolutionary iterations, $\mathrm{iter}_{\max}$ is the maximum number of evolutionary iterations; $f(X_i)$, $f_{\max}$ and $f_{\min}$ have the same meaning as in formula (1); n is a nonlinear harmonic factor, usually set to $n = 3$ [19]. From formula (2), it can be seen that the seed diffusion range is initially large and later smaller; seeds produced by well-adapted plants have a larger diffusion range, and smaller vice versa. This mechanism effectively guarantees the early exploration capability and later exploitation ability of the algorithm.

In the process of seed transmission, different displacement distances for different dimensions enable better seed diversity. The seed mechanical transmission mode in garden balsam optimization algorithm is shown in Algorithm 1.

where $\mathrm{U}(0,1)$ represents a random number uniformly distributed in the interval [0,1] and round() represents a rounding operation.

3.3 Second propagator

In the natural world, individual seeds are affected by natural factors such as wind, water flow and animal transport after mechanical diffusion, and then, second transmission occurs, a process that can effectively increase population diversity. The second transmission mechanism introduced by garden balsam optimization algorithm makes it possible that seeds are not only be sown in the vicinity of the plants, but also be spread farther away, thus improving the ability of the algorithm to explore the solution space. The process of second transmission is as follows: N_{sec} seeds are randomly selected and subject to mutation operations. Differential mutation is adopted here to produce mutation seeds.

Differential mutation is a mutation to improve performance of garden balsam optimization algorithm using difference information between individuals. By differential mutation method, variant individuals can be improved, population diversity can be enhanced, and the population can be prevented from falling into a local optimal solution. Its manifestation is as follows:

$$x_{i1}^k = x_B^k + F\left(x_{i2}^k - x_{i3}^k\right) \tag{3}$$

where x_{i1}^k is the position of the target individual in the k-dimension, x_B^k is the position of the best individual of the current population in the k dimension and F is the zoom factor used to zoom the difference vector, which is generally set to 0–2; x_{i2}^k and x_{i3}^k are the positions of two dissimilar individuals in the k dimension. The second transmission algorithm in garden balsam optimization algorithm is shown in Algorithm 2.

Algorithm.1. Mechanical transmission mode of GBO algorithm.

1: Initialize the plant position: $\mathrm{x_i}$
2: Let $\mathrm{z^k} = \mathrm{round}\big(\mathrm{U}(0,1)\big), \mathrm{k} = 1,2,\cdots,\mathrm{D}$
3: **while** $\mathrm{x_{ik}}$ in each dimension, where $\mathrm{z^k} = 1$ do
4: Calculate the displacement variable $\Delta\mathrm{X_{ik}} = \mathrm{A_i} \times \mathrm{U}(-1,1)$
5: $\mathrm{x'_{ik}} = \mathrm{x_{ik}} + \Delta\mathrm{X_{ik}}$
6: **if** $\mathrm{x'_{ik}}$ crosses the boundary, then
7: conduct mapping operation on $\mathrm{x'_{ik}}$, with reference to formula(5)
8: **end if**
9: **end while**

Algorithm.2. Second transmission algorithm of GBO algorithm.

Initialize the seed position: $x_{i1} = X_i$
Randomly select two dissimilar individuals x_{i2} and x_{i3} from the current population
while each dimension do
  $x_{i1}^{k\prime} = x_B^k + F(x_{i2}^k - x_{i3}^k)$
end while

3.4 Competitive exclusion rules

As the number of iterations of the algorithm continues to increase, when the sum of the population and the resulting progeny population reaches the preset maximum population size $N_{\max}$, the algorithm performs a competitive exclusion operation. The rule is to rank all individuals in the current population according to the fitness value, retain individuals with good fitness values (elite solutions), randomly select the remaining individuals and eliminate excess individuals. The number of elite solutions is calculated according to formula (4) and rounded up to an integer. The population size remains unchanged at $N_{\max}$ hereafter. That is, the algorithm first seizes suitable field by individual's rapid reproduction and then retains the more competitive individuals in the relatively stable environment to continue searching for space. The number of elite solutions has gradually increased with the evolution of iterations. That is, taking early global exploration into account, the later local exploitation capability can be guaranteed.

This mechanism gives opportunity for individuals with low fitness value to reproduce. Their offspring with better fitness values can survive. This method of first making plants rapidly reproduces and grows to adapt to the environment; then, retaining some more competitive individuals in a relatively stable environment for further environmental exploration can be also regarded as a simulation of r and k selections of organisms [20]. N_{best} indicates the number of elite solutions.

$$N_{\text{best}} = \frac{\text{iter}}{\text{iter}_{\max}} N_{\max} \tag{4}$$

3.5 Cross-border mapping rules

In the process of transmission, seeds may fall outside the scope of feasible areas. Such kind of seeds is meaningless, and they must be pulled back to the feasible area according to certain rules. The garden balsam optimization algorithm handles this situation using random mapping rule. That is, the out-of-bounds seeds are mapped using formula (5), which guarantees that all individuals remain in the feasible space.

$$x_i^{k\prime} = x_{\text{LB}}^k + U(0,1)\left(x_{\text{UB}}^k - x_{\text{LB}}^k\right) \tag{5}$$

where x_{UB}^k denotes the upper boundary of k dimension, x_{LB}^k denotes the lower boundary of k dimension and $U(0,1)$ is the same as in Algorithm 3.

Algorithm.3. Garden balsam optimization algorithm.

Initialize seed N0, randomly spread it in the solution space, and calculate the fitness value $f(x_i)$ of the resulting plant
while the termination condition do is not reached
  for all the plants do
    Calculate the number of seeds produced per plant
    Calculate the distance at which each plant spreads seeds
    Randomly proceed with mechanical seed transmission
  end for
  for $j = 1 \rightarrow N_{sec}$ do
    randomly select a seed for second transmission
  end for
  Map the cross-border seed according to the mapping algorithm
  evaluate seed's fitness value and make selections
end while

3.6 Discussion

Exploration and exploitation are two important features of population-based (or population) optimization algorithms. In optimization algorithm, exploration indicates global search capacity by investigating different unknown regions, while exploitation indicates local search capability by locally searching the optimal point. Therefore, if a population-based algorithm can achieve balance between exploration and exploitation of search space, then the algorithm is considered as effective. An inherent weakness of most population-based stochastic algorithms is premature convergence, while premature convergence and stagnation are important considerations in designing natural algorithms.

Population diversity is the key to the performance of population optimization algorithms. It can ensure that the algorithm jumps out of the local extreme points and converges to the global optimal point. Greater population diversity means wider individual distribution in the algorithm and higher possibility of finding the optimal solution without significantly affecting convergence ability of the algorithm. The diversity of garden balsam optimization algorithm is mainly reflected in the following three aspects.

1. Diversity of seed number and ejection distance
   Under the action of the ejection propagator, different parents produce different numbers of seeds according to their own fitness values, and the ejection distance also differs. For parents with good fitness values, more seeds are produced and ejected further. For those with poor fitness values, fewer seeds are produced and ejected for a smaller distance. Hence, the diversity of seed number and ejection distance is guaranteed.
2. Variety of transmission modes
   To simulate the second transmission mechanism in the natural world, the garden balsam optimization algorithm is designed with a second propagator, and a specific number of seeds are randomly selected for differential mutation operations to enable secondary displacements of these seed positions. The second propagator has nothing to do with the parent fitness value, but concerns its own coordinate value. Second propagator is essentially different from mechanical propagator, which guarantees transmission diversity.
3. Diversity of selection mode

When the population size reaches the upper limit, the algorithm initiates an elitist random selection strategy, in which the number of elite solutions gradually increases with the evolutionary iterations. This guarantees the global exploration in the early iteration and also ensures local exploitation capabilities in the later period.

4 Experimental investigation

Experimental comparison was made between garden balsam optimization algorithm and mature optimization algorithms including PSO, DE, ABC, BBO, DE, TLBO to verify the algorithm's usability and performance in terms of function optimization. The constrained optimization test set given in CEC 2006 was used in the experiment, which contained 24 constrained optimization functions. Detailed mathematical formulas and characteristics of each function are given in the literature [21]. These functions concerning continuous, unbiased constrained optimization problems have varying degrees of complexity and multimodality, each with different numbers of variables and data ranges.

Table 1 Parameter values of each algorithm

PSO algorithm	BBO algorithm	GBO algorithm
Population size: 50,	Population size: 50	Population size: 50
Inertia weight: 0.6	Immigration rate: 1	Initial size: 5
Cognitive parameter: 1.65	Emigration rate: 1	Second transmission size: 5
Social parameter: 2	Mutation factor: 0.01	Max seed size: 5
ABC algorithm	DE algorithm	TLBO algorithm
Number of employed bees: 25	Population size: 50	Population size: 50
Number of onlooker bees: 25	Crossover factor: 0.5	
Limit: number of generations	Constant factor: 0.5	
ABC algorithm	DE algorithm	TLBO algorithm
Number of employed bees: 25	Population size: 50	Population size: 50
Number of onlooker bees: 25	Crossover factor: 0.5	
Limit: number of generations	Constant factor: 0.5	

Table 2 Comparative results of test functions obtained by different algorithms

Function		PSO	DE	ABC	BBO	TLBO	GBO
G01	Best	− 15	− 15	− 15	− 14.977	− 15	− 15
(− 15.00)	Worst	− 13	− 11.828	− 15	− 14.5882	− 6	− 15
	Mean	− 14.71	− 14.555	**− 15**	− 14.7698	− 10.782	**− 15**
G02	Best	− 0.669158	− 0.472	− 0.803598	− 0.7821	− 0.7835	− 0.7816
(− 0.803619)	Worst	− 0.299426	− 0.472	− 0.749797	− 0.7389	− 0.5518	− 0.4735
	Mean	− 0.41996	− 0.665	**− 0.792412**	− 0.7642	− 0.6705	− 0.7731
G03	Best	− 1	− 0.99393	− 1	− 1.0005	− 1.0005	− 1.0005
(− 1.0005)	Worst	− 0.464	− 1	− 1	− 0.0455	0	0
	Mean	0.764813	**− 1**	**− 1**	− 0.3957	− 0.8	− 0.9862
G04	Best	− 30,665.539	− 30,665.539	− 30,665.539	− 30,665.539	− 30,665.5387	− 30,665.5387
(− 30665.539)	Worst	− 30,665.539	− 30,665.539	− 30,665.539	− 29942.3	− 30,665.5387	− 30,665.5387
	Mean	**− 30,665.539**	**− 30,665.539**	**− 30,665.539**	− 30,411.865	**− 30,665.5387**	**− 30,665.5387**
G05	Best	5126.484	5126.484	5126.484	5134.2749	5126.486	5126.486
− 5126.486	Worst	5249.825	5534.61	5438.387	7899.2756	5127.714	5126.6876
	Mean	5135.973	5264.27	5185.714	6130.5289	5126.6184	**5126.5265**
G06	Best	− 6961.814	− 6954.434	− 6961.814	− 6961.8139	− 6961.814	− 6961.814
(− 6961.814)	Worst	− 6961.814	− 6954.434	− 6961.805	− 5404.4941	− 6961.814	− 6961.814
	Mean	**− 6961.814**	− 6954.434	**− 6961.813**	− 6181.7461	**− 6961.814**	**− 6961.814**
G07	Best	24.37	24.306	24.33	25.6645	24.3101	24.3025
− 24.3062	Worst	56.055	24.33	25.19	37.6912	27.6106	25.0079
	Mean	32.407	**24.31**	24.473	29.829	24.837	24.4051
G08	Best	− 0.095825	− 0.095825	− 0.095825	− 0.095825	− 0.095825	− 0.095825
(− 0.095825)	Worst	− 0.095825	− 0.095825	− 0.095825	− 0.095817	− 0.095825	− 0.095825
	Mean	**− 0.095825**	**− 0.095825**	**− 0.095825**	− 0.095824	**− 0.095825**	**− 0.095825**
G09	Best	680.63	680.63	680.634	680.6301	680.6301	680.6301
− 680.6301	Worst	680.631	680.631	680.653	721.0795	680.6456	680.6425
	Mean	**680.63**	**680.63**	**680.64**	692.7162	680.6336	**680.6315**
G10	Best	7049.481	7049.548	7053.904	7679.0681	7250.9704	7049.3912
− 7049.28	Worst	7894.812	9264.886	7604.132	9570.5714	7291.3779	7251.4592
	Mean	7205.5	7147.334	7224.407	8764.9864	7257.0927	**7089.5347**
G11	Best	0.749	0.752	0.75	0.7499	0.7499	0.7499
− 0.7499	Worst	0.749	1	0.75	0.92895	0.7499	0.7499
	Mean	**0.749**	0.901	**0.75**	0.83057	**0.7499**	**0.7499**
G12	Best	− 1	− 1	− 1	− 1	− 1	− 1
(− 1)	Worst	− 0.994	− 1	− 1	− 1	− 1	− 1
	Mean	− 0.998875	**− 1**	**− 1**	**− 1**	**− 1**	**− 1**
G13	Best	0.085655	0.385	0.76	0.62825	0.44015	0.2988
(− 0.05394)	Worst	1.793361	0.99	1	1.45492	0.95605	0.9372
	Mean	0.569358	0.872	0.968	1.09289	0.69055	0.5138
G14	Best	− 44.9343	54.6979	− 45.7372	− 44.6431	− 46.5903	− 47.7322
(− 47.764)	Worst	− 37.5000	257.7061	− 12.7618	− 23.3210	− 17.4780	− 46.2908
	Mean	− 40.8710	175.9832	− 29.2187	− 40.1071	− 39.9725	**− 46.6912**
G15	Best	961.715	962.664	961.715	961.7568	961.715	961.7164
− 961.715	Worst	972.317	1087.3557	962.1022	970.317	964.8922	961.7312
	Mean	965.5154	1001.4367	961.7537	966.2868	962.8641	961.7253
G16	Best	− 1.9052	− 1.9052	− 1.9052	− 1.9052	− 1.9052	− 1.9052
(− 1.9052)	Worst	− 1.9052	− 1.1586	− 1.9052	− 1.9052	− 1.9052	− 1.9052
	Mean	**− 1.9052**	− 1.6121	**− 1.9052**	**− 1.9052**	**− 1.9052**	**− 1.9052**

Table 2 (continued)

Function		PSO	DE	ABC	BBO	TLBO	GBO
G17	Best	8857.514	9008.5594	8854.6501	8859.713	8853.5396	8853.5396
– 8853.5396	Worst	8965.401	9916.7742	8996.3215	8997.145	8919.6595	8913.6934
	Mean	8899.4721	9384.268	8932.0444	8941.9245	**8876.5071**	8879.5402
G18	Best	− 0.86603	− 0.65734	− 0.86531	− 0.86603	− 0.86603	− 0.86603
(–0.86603)	Worst	− 0.51085	− 0.38872	− 0.85510	− 0.86521	− 0.86294	− 0.86607
	Mean	− 0.82760	− 0.56817	− 0.86165	− **0.86587**	− 0.86569	− 0.86605
G19	Best	33.5358	39.1471	32.6851	33.3325	32.7916	32.6912
– 32.6555	Worst	39.8443	71.3106	32.9078	38.5614	36.1935	33.1784
	Mean	36.6172	51.8769	**32.768**	36.0078	34.0792	32.2341
G20	Best	0.24743	1.26181	0.24743	0.24743	0.24743	0.24743
– 0.24979	Worst	1.8732	1.98625	0.28766	1.52017	1.84773	0.28766
	Mean	0.97234	1.43488	0.26165	0.80536	1.22037	**0.26051**
G21	Best	193.7311	198.8151	193.7346	193.7343	193.7246	193.4458
– 193.274	Worst	409.132	581.2178	418.4616	330.1638	393.8295	242.3719
	Mean	345.6595	367.2513	366.9193	275.5436	264.6092	**197.1178**
G22	Best	1.68E+22	1.02E+15	1.25E+18	2.82E+08	4.50E+17	4.96E+02
– 236.4309	Worst	3.25E+23	6.70E+16	2.67E+19	1.25E+18	4.06E+19	7.81E+17
	Mean	1.63E+23	1.41E+16	1.78E+19	4.10E+17	1.61E+19	9.58E+07
G23	Best	− 105.9826	2.3163	− 72.6420	− 43.2541	− 385.0043	− 397.9034
(– 400.055)	Worst	0	74.6089	0	0	0	− 132.0517
	Mean	− 25.9179	22.1401	− 7.2642	− 4.3254	− 83.7728	− **367.1852**
G24	Best	− 5.5080	− 5.5080	− 5.5080	− 5.5080	− 5.5080	− 5.5080
(− 5.5080)	Worst	− 5.5080	− 5.4857	− 5.5080	− 5.5080	− 5.5080	− 5.5080
	Mean	− **5.5080**	− 5.4982	− **5.5080**	− **5.5080**	− **5.5080**	− **5.5080**

The bold values indicate best result

The comparative algorithm selected in the experiment has been previously used by different people in attempts to solve various constrained optimization problems [25–37]. The results show good effect of these algorithms on constrained optimization problem. In addition, it was found in the literature survey that the algorithm under consideration was successfully applied in a variety of engineering applications, with expected results achieved.

4.1 Experimental setting

The proposed garden balsam optimization algorithm was compared with PSO, BBO, DE, ABC and TLBO under the same experimental platform. The function was evaluated for 240,000 times, and each was run 100 times [22–24]. The parameters of each algorithm in the experiment are shown in Table 1. In addition, "static penalty" method is applied to all competitive algorithms as constraint handling technology to maintain the consistency of the technology. The computational code for individual algorithms has been provided by developers of these algorithms.

4.2 Results and discussion

Targeting at the 24 constrained optimization functions in the test set, the best, worst and mean solutions are independently run by the six algorithms involved in the comparison experiment for 100 times as shown in Table 2. The data of the comparative algorithm are taken from the literature [22–24].

The convergence speed of meta-heuristic algorithm is an important standard for evaluating its performance. So far, GBO algorithm convergence performance has been compared with the other five algorithms on four functions (G1, G3, G8 and G24). The test functions selected show different objective function characteristics (that is, G1 is a quadratic type, G3 is a polynomial, G8 is nonlinear, and G24 is linear). The convergence curve is shown in Fig. 3. It can be seen from the figure that GBO algorithm has better convergence performance than other algorithms.

Table 2 shows in the last column the result of 100 independent runs of garden balsam optimization algorithm on the G01–G24 benchmark function. Each run is evaluated for 240,000 times, and the "worst," "best" and

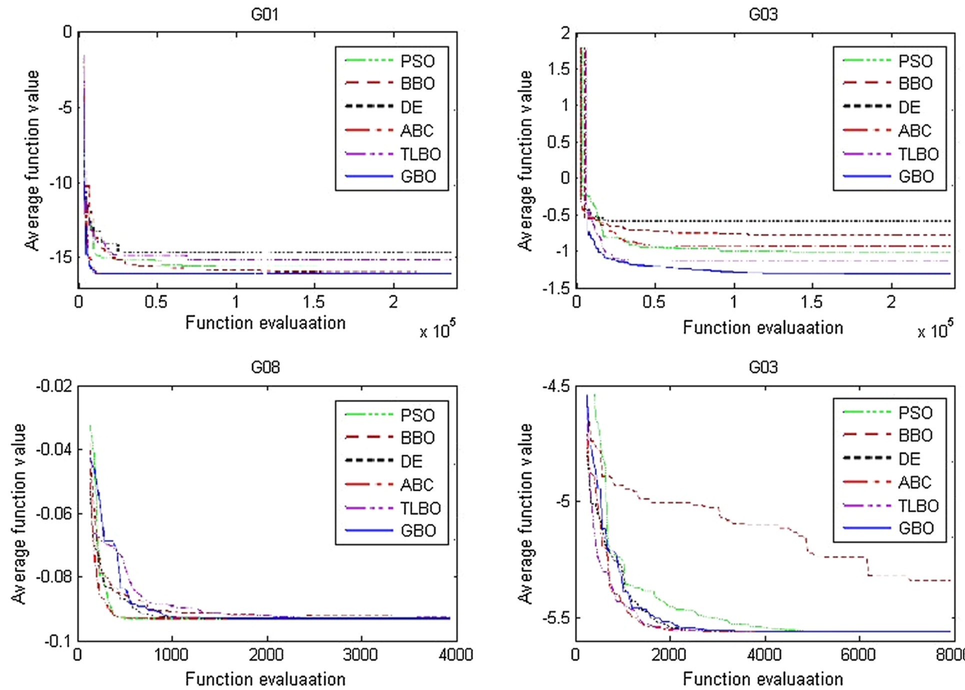

Fig. 3 Convergence of each algorithm on four functions

Table 3 Success rate of various algorithms for test functions

Function	PSO	BBO	DE	ABC	TLBO	GBO
G01	38	0	94	100	26	100
G02	0	0	0	0	0	0
G03	59	23	41	67	74	95
G04	100	16	100	100	100	100
G05	61	0	93	28	92	97
G06	100	21	100	100	100	100
G07	21	0	26	28	23	42
G08	100	94	100	100	100	100
G09	84	26	95	89	91	100
G10	0	0	0	0	0	0
G11	100	57	19	100	100	100
G12	100	100	100	100	100	100
G13	0	0	0	0	0	0
G14	0	0	0	0	0	0
G15	53	0	73	42	81	86
G16	100	18	100	100	100	100
G17	0	0	0	0	58	72
G18	56	0	61	73	64	79
G19	0	0	0	0	0	0
G20	0	0	0	0	0	0
G21	12	0	24	36	35	61
G22	0	0	0	0	0	0
G23	0	0	0	0	0	0
G24	100	27	100	100	100	100

"mean" operation results are compared with five other mature algorithms. The optimal solution was found in garden balsam optimization algorithm on 16 benchmark functions. Failure to find the optimal solution on the remaining eight benchmark functions also occurred in the other five functions. The balsam optimization algorithm outperforms the rest of the comparative algorithms in mean (M) on 15 test functions.

The success rate of the six algorithms in 100 independent runs on G01–G24 benchmark function is shown in Table 3. In the eight test functions (i.e., G02, G10, G13, G14, G19, G20, G22 and G23), all algorithms achieved a success rate of 0. In other test functions, the GBO algorithm is equal or superior to the other five algorithms.

Table 4 shows the "mean number" of function evaluation needed for the six algorithms to achieve a global optimum in 100 independent runs on G01–G24 reference functions (except G02, G10, G13, G13, G14, G19, G20, G22 and G23 functions). It can also be seen that relatively superior results are obtained in garden balsam optimization algorithm on all benchmark functions except G01 and G12. Garden balsam optimization algorithm also has good function evaluation standard deviation in evaluation of most functions.

5 Statistical tests

It can be seen from the results in Tables 2, 3 and 4 that garden balsam optimization algorithm outperforms other competitive algorithms in performance. However,

Table 4 Mean number of function evaluations required to reach global optimum value by comparative algorithms for G01–G24 over 100 independent runs

Function		PSO	BBO	DE	ABC	TLBO	GBO
G01	Mean_FE	33,750	–	6988.89	13,200	9750	12,570
	Std_FE	3872.01	–	157.674	708.676	4171.93	6312.43
G02	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G03	Mean_FE	84,610	157,950	136,350	121,950	178,083	67,400
	Std_FE	35,670	5939.7	78,988.3	64,296.1	43,819.3	29,776.2
G04	Mean_FE	14,432	189,475	14,090	29,460	5470	10,135
	Std_FE	309.23	35,390.7	1499.22	2619.25	804.225	1413.63
G05	Mean_FE	57,921	–	108,572	197,749	46,888	43,356
	Std_FE	14,277.4	–	41,757.1	20,576.8	19,623.2	34,164.3
G06	Mean_FE	14,923	140,150	17,540	69,310	11,600	15,395
	Std_FE	1789.32	22,273.9	1214.91	3753.65	2056.43	2566.61
G07	Mean_FE	97,742	–	147,650	114,351	147,550	92,916.7
	Std_FE	2984.2	–	4737.62	11,384.4	5020.46	17,237.3
G08	Mean_FE	622	4290	725	670	680	635
	Std_FE	189.78	4418.32	259.54	249.666	181.353	171.675
G09	Mean_FE	34,877	194,700	57,205	149,642	37,690	23,235
	Std_FE	12,280.1	29,557.1	10,779.1	73,436.8	26,350.6	10,806.2
G10	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G11	Mean_FE	23,312	35,490	205,250	29,140	3000	53,270
	Std_FE	1231.41	30,627.4	8273.15	12,982.5	1354.83	18,215.2
G12	Mean_FE	1204	1865	1150	1190	2480	2190
	Std_FE	341.3	2240.54	263.523	747.514	917.484	824.554
G13	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G14	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G15	Mean_FE	41,972	–	36,391.7	157,800	52,287.5	36,756.3
	Std_FE	4073.9	–	5509.21	57,558.5	47,937.1	28,670.6
G16	Mean_FE	7114	85,200	12,565	19,670	7840	13,045
	Std_FE	643.3	16,122	1155.19	714.998	2709.74	1358.6
G17	Mean_FE	–	–	–	–	126,980	65,600
	Std_FE	–	–	–	–	46,591.8	65,053.8
G18	Mean_FE	23,769	–	170,140	114,120	19,226	35,360
	Std_FE	1009.78	–	20,227.7	58,105.8	5762.16	7731.14
G19	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G20	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G21	Mean_FE	39,937	–	89,500	99,150	108,533	28,037.5
	Std_FE	4302.2	–	14,283.6	3647.94	8677.17	7032.35
G22	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G23	Mean_FE	–	–	–	–	–	–
	Std_FE	–	–	–	–	–	–
G24	Mean_FE	2469	84,625	4855	5400	2710	3715
	Std_FE	245.5	2015.25	429.761	618.241	864.677	575.929

– Indicates that algorithm is failed to obtained a global optimum value for that function, *Mean_FE* mean number of function evaluations. *Std_FE* standard deviation of function evaluations

Table 5 Friedman rank test for the "best" and "mean" solutions obtained for G01–G24 functions

Algorithms	Friedman value	Normalized value	Rank	Algorithms	Friedman value	Normalized value	Rank
PSO	82.5	2.01	3	PSO	82	2.73	3
BBO	124	3.02	5	BBO	136	4.53	6
DE	82.5	2.01	3	DE	84	2.8	4
ABC	96	2.34	6	ABC	90	3	5
TLBO	57	1.39	2	TLBO	61	2.03	2
GBO	41	1	1	GBO	30	1	1

Table 6 Friedman rank test for the "success rate" of the solutions obtained for G01–G24 functions

Algorithms	Friedman value	Normalized value	Rank
PSO	53.5	2.06	5
BBO	83	3.19	6
DE	47	1.81	4
ABC	42	1.62	2
TLBO	42.5	1.63	3
GBO	26	1	1

Table 7 Holm–Sidak test for the "best" and the "mean" solutions obtained for G01–G24 functions

Test for best solution		Test for mean solution	
Algorithm[a]	p value	Algorithm[a]	p value
1–3	0.01,204	1–3	0.01102
1–5	0.15318	1–5	0.31052
1–4	0.21983	1–4	0.44458
1–2	0.21992	1–2	0.45587
1–6	0.97641	1–6	0.87543

[a]1—GBO, 2—PSO, 3—BBO, 4—DE, 5—ABC, 6—TLBO

Friedman rank test and Holm–Sidak test are necessary to prove the significance of the proposed algorithm.

Table 5 shows Friedman rank tests in which the "best" and "mean" solutions are obtained on G01–G24 function (G22 function is excluded as no algorithm succeeds on G22 function). Therefore, it can be easily seen from Friedman rank test results in the table that the proposed garden balsam optimization algorithm ranks the first with regard to the "best" and "mean" solutions for all the considered test functions. Table 6 shows the Friedman rank test result of obtained solution "success rate." Since there was no difference in comparative algorithm between the 10 benchmark functions, only 14 samples were involved in the test, and the proposed garden balsam optimization algorithm ranked the first in the test result.

The Friedman rank test can demonstrate the significant difference between different algorithms in performance when the same problem is handled. It is used to rank algorithms according to the result data in the form of order, but cannot specify any statistical difference in the results. Holm–Sidak test as a post hoc test method can be used to determine statistical differences between algorithms. Table 7 shows the Holm–Sidak test results in which the "best" and "mean" solutions are obtained on G01–G24 function. The p values obtained by all the algorithms from Holm–Sidak test show the statistical difference between the proposed garden balsam optimization algorithm and other algorithms.

6 Conclusion

This paper introduces a new numerical random search algorithm that simulates natural behavior—garden balsam optimization algorithm. The process and characteristics of natural transmission of garden balsam are described in detail. The simulation process involves the design of garden balsam optimization algorithm, including mechanical propagator, second propagator, competitive selection strategy and cross-border mapping rule. At the same time, the algorithm's steps, pseudo-codes and flow charts are given. The characteristics of garden balsam optimization algorithm and the effect of each factor on the algorithm performance are analyzed. Meanwhile, comparative experiment is made. Seen from the experimental results, GBO has a good performance in the aspects of optimal solution, mean solution, success rate and convergence speed. Finally, statistical analysis is made for the experimental work. Seen from the statistical test results of Friedman rank test and Holm–Sidak test, GBO is superior to other natural optimization algorithms in terms of constrained optimization problem.

Acknowledgements This work is partially supported by Natural Science Foundation of China under Grant 5147509, the rolling support plan for Excellent Innovation team of Ministry of Education of China under Grant IRT_16R12, the Science and technology project of Henan Province under Grant 172102310249 and Key scientific research projects in Henan colleges and Universities under Grant 17B520030.

References

1. Akay B, Karaboga D (2012) A modified Artificial Bee Colony algorithm for real-parameter optimization. Inform Sci 192(3):120–142
2. Ali M, Ahn CW, Pant M (2014) Multi-level image thresholding by synergetic differential evolution. Appl Soft Comput 17(5):1–11
3. Boussad I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237(7):82–117
4. Berdahl A, Torney CJ, Ioannou CC, Faria JJ, Couzin ID (2013) Emergent sensing of complex environments by mobile animal groups. Science 339(6119):574–576
5. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
6. Tampuu A, Matiisen T, Kodelja D, Kuzovkin I et al (2017) Multiagent cooperation and competition with deep reinforcement learning. PLoS ONE 12(4):e0172395
7. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B Cybern 26(1):29–41
8. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Piscataway, pp 1942–1948
9. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Opt 39(3):459–471
10. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
11. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
12. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inf 1(4):355–366
13. Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. Adv Swarm Intell 21(7):355–364
14. Farmer JD, Packard N, Perelson A (1986) The immune system, adaptation and machine learning. Physica D 22(3):187–204
15. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst 22(3):52–67
16. Chen YL (2001) Flora of China, vol 47. Science Press, Beijing, pp 1–243 **(in Chinese)**
17. Song Y, Yuan YM (2003) Chromosomal evolution in Balsaminaceae, with cytological observations on 45 species from Southeast Asia. Caryologia 56(4):463–481
18. Shui YM, Janssens S, Huang SH, Chen WH, Yang ZG (2011) Three new species of impatiens L. from China and Vietnam: preparation of flowers and morphology of pollen and seeds. Syst Bot 36(2):428–439
19. Attanasi A, Cavagna A et al (2014) Information transfer and behavioural inertia in starling flocks. Nat Phys 10(9):691–696
20. Dawkins R (1999) The extended phenotype: the long reach of the gene. Oxford University Press, Oxford, pp 156–164
21. Vivek K, Vimal J (2015) Heat transfer search (HTS): a optimization algorithm. Inf Sci 324(10):217–246
22. Karaboga D, Akay B (2011) A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. Appl Soft Comput 11(3):3021–3031
23. Attanasi A, Cavagna A et al (2014) Collective behaviour without collective order in wild swarms of midges. PLoS Comput Biol 10(7):1–10
24. Bialek W, Cavagna A et al (2012) Statistical mechanics for natural flocks of birds. Proc Natl Acad Sci 109(13):4786–4791
25. Mustaffa Z, Yusof Y, Kamaruddin SS (2014) Enhanced artificial bee colony for training least squares support vector machines in commodity price forecasting. J Comput Sci 5(2):196–205
26. Boedecker J, Obst O, Lizier JT, Mayer NM, Asada M (2012) Information processing in echo state networks at the edge of chaos. Theory Biosci 131(3):205–213
27. Buid T, Tuan TA, Hoang ND et al (2017) Spatial prediction of rainfall-induced landslides for the Lao Cai area (Vietnam) using a hybrid intelligent approach of least squares support vector machines inference model and artificial bee colony optimization. Landslides 14(2):447–458
28. Degertekin SO, Lamberti L, Hayalioglu MS (2017) Heat transfer search algorithm for sizing optimization of truss structures. Latin Am J Solids Struct 14(3):373–397
29. Abdollahi M, Isazadeh A, Abdollahi D (2013) Imperialist competitive algorithm for solving systems of nonlinear equations. Comput Math Appl 65(4):1894–1908
30. Esmaeili R, Dashtbayazi MR (2015) Modelling and optimization for micro structural properties of Al/SiC nanocomposite by artificial neural network and genetic algorithm. Expert Syst Appl 41(5):5817–5831
31. Butail S, Ladu F, Spinello D, Porfiri M (2014) Information flow in animal–robot interactions. Entropy 16(3):1315–1330
32. Kumar V, Chhabra JK, Kumar D (2014) Automatic cluster evolution using gravitational search algorithm and its application on image segmentation. Eng Appl Artif Intell 29(5):93–103
33. Kuo RJ, Hung SY, Cheng WC (2014) Application of an optimization artificial immune network and particle swarm optimization-based fuzzy neural network to an RFID-based positioning system. Inf Sci 262(3):78–98
34. Lobato FS, Sousa MN, Silva MA, Machado AR (2014) Multi-objective optimization and bio-inspired methods applied to machinability of stainless steel. Appl Soft Comput 22(5):261–271
35. Kuo RJ, Tseng YS, Chen ZY (2016) An RFID indoor positioning system by using particle swarm optimization-based artificial neural network. J Intell Manuf 27(6):1191–1207
36. Patel VK, Savsani VJ (2014) Optimization of a plate-fin heat exchanger design through an improved multi-objective teaching-learning based optimization (MO-ITLBO) algorithm. Chem Eng Res Des 92(11):2371–2382
37. Zhang W, Qu Z, Zhang K et al (2017) A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting. Energy Convers Manag 136(4):439–451