Evolutionary Intelligence
https://doi.org/10.1007/s12065-018-0172-2

RESEARCH PAPER

Future search algorithm for optimization

M. Elsisi[1]

Received: 14 July 2018 / Revised: 2 September 2018 / Accepted: 7 September 2018

Abstract
This paper proposes a new optimization algorithm named future search algorithm (FSA). This algorithm mimics the person's life. People in the world search for the best life. If any person found that his life is not good, he tries to change it and he imitates the successful persons. According to this behavior, this algorithm is built by mathematical equations. The FSA can update the random initial. Furthermore, it uses the local search between people and the global search between the histories optimal persons to achieve the best solutions. The proposed algorithm does not have tuned parameters. In addition, it has low computational complexity, fast convergence, and high local optima avoidance. The performance of the proposed algorithm is evaluated by applying it to solve some benchmarks test functions. These test functions have various characteristics necessary to evaluate the FSA. In addition, the performance of the proposed algorithm is compared with five other well-known methods. The results confirm a better performance of the proposed algorithm to get the optimal solution with fewer iterations number than other methods.

Keywords Future search algorithm (FSA) · Benchmark functions · Constrained optimization · Meta-heuristic algorithms

1 Introduction

In the new century, the most fields need heuristic algorithms (HA) to get unknown parameters. The heuristic algorithms reduce the effort, the time, and errors of the conventional methods such as the trial–error method and the experience of the designer method. So, the most researchers concentrate their researches to find new HA. There are different types of HA used for the optimal tuning of these parameters such as genetic algorithm (GA) [1–4], particle swarm optimization (PSO) [5, 6], bacterial foraging optimization algorithm (BFOA) [7, 8], artificial bee colony (ABC) [9–11], tabu search algorithm [12–14], imperialist competitive algorithm (ICA) [15], gravitation search algorithm (GSA) [16, 17], bat inspired algorithm (BIA) [18, 19], and other techniques [20–24]. All these HA give acceptable results for the optimization of unknown parameters compared with conventional methods. The main difference between all HA is the constructing method of each one. There is some HA have results close to each other for simple applications, while the results difference may appear for the complicated applications. All these HA start with a random initial and they build its iterations based on the best solution of the random initial. If the best solution of the random initial far from the optimal solution, the HA may take long iterations number to reach the optimal solution. Another problem associated with some of these HA, the solutions are built based on the global best solution only in some of these HA and based on the local best solution only in others. Also, this problem leads to a long number of iterations. Furthermore, some of these HA are built based on more and complex mathematical equations which lead to more time and long number of iterations to reach the optimal solution. This paper proposes a new optimization algorithm named future search algorithm (FSA) to overcome the above problems. This algorithm mimics the human behavior to find the best life around the world. The space of solutions in the algorithm is represented by persons. The person which achieves the best performance in a country is the optimal local solution between the other persons. Every year, this solution may be changed and there is another solution by a different person in another country. The algorithm updates the local solutions in each iteration and it selects the best solution overall iterations which considered the global solution. This algorithm can update the random initial and it

✉ M. Elsisi
mahmoud.elsesy@feng.bu.edu.eg

[1] Electrical Engineering Department, Faculty of Engineering (Shoubra), Benha University, 108 Shoubra St., B. O. Box 11241, Cairo, Egypt

Published online: 14 September 2018

builds its iterations based on the local and global solutions to achieve the best solutions. Twenty-three standard benchmark functions are used to evaluate the performance of the proposed FSA. Furthermore, the performance of the proposed FSA is compared with the GA, PSO, GSA, firefly algorithm (FFA), and lightning search algorithm (LSA). The results proved a better performance of the proposed FSA to get the optimal solution with fewer iterations number than the GA, PSO, GSA, FFA, and LSA.

2 Future search algorithm

All people in the world look for the best life. If any person found that his life is not good, he tries to emulate the life of the best person around the world. The future search algorithm uses this behavior to find the best solutions. The FSA is formulated by mathematical equations. It can update the random initial and it utilizes the local search between people and the global search between the histories optimal persons as mentioned in the Abstract and the Introduction sections. The others HA start its steps by a random initial and it builds its iterations based on the best solution of the random initial. This best solution may be far from the optimal solution, this makes the HA take long iterations number to reach the optimal solution. The FSA can overcome this problem and it updates the random initial every each iteration. In each HA, there is a local best solution between the multi-agents and global solution between the iterations. Some of HA update its new solution based on the local best solution only. But the others update its new solution based on the global best solution only. These methods may take a long number of iterations. The FSA utilizes the local best solution and the global best solution to find the best solutions. Some of HA have more and complex mathematical equations which can lead to more time and long number of iterations to reach the optimal solution. The proposed FSA is built by simple mathematical equations. In FSA, the space of solutions is represented by person search for the best life in the countries of the world. The person which achieves the best performance in a country is representing the optimal local solution between the other persons. Every year, this solution may be changed and there is another solution by a different person in another country. The person which achieves the best performance in a country over some years is representing the optimal global solution between the other persons. If the performance of the persons in a year does not achieve a great performance from the last year, the initial positions of each person will change.

The methodology of this algorithm is built based on mathematical equations as following, it starts its steps based on random solutions by this equation,

$$S(i,:) = Lb + (Ub - Lb) .* rand(1, d) \tag{1}$$

where S: Solutionm, i: Current solution of population size, Lb : Lower limit bounds, Ub: Upper limit bounds, rand: Uniformly distributed pseudo-random numbers, d: Dimensions of problem.

After finding the solutions, each solution is defined as a local solution (LS) and the best one is selected and it is defined as a global solution (GS) and then the algorithm starts its iterations to find the optimal solution. The algorithm defines the solution of each person in the population size by the following equation which depends on GS and LS.

First, the search in each country depends on the LS which support the exploitation characteristic of the proposed algorithm and it is computed by

$$S(i,:)_L = (LS(i,:) - S(i,:)) * rand \tag{2}$$

Second, the search in the overall world depends on the GS which support exploration characteristic of the proposed algorithm and it is defined as follows

$$S(i,:)_G = (GS - S(i,:)) * rand \tag{3}$$

After computing the local and the global convergences, the solution of each person is defined by

$$S(i,:) = S(i,:) + S(i,:)_L + S(i,:)_G \tag{4}$$

Then the algorithm updates the GS and LS. After finding the solutions in the current iteration and the new GS and LS, the algorithm updates the random initial of Eq. (1) and this property is added to support exploration characteristic of the proposed algorithm and it is defined by

$$S(i,:) = GS + (GS - LS(i,:)) * rand \tag{5}$$

Then, the algorithm checks the GS and LS due to the updating of initial and it updates them if they are better than the GS and LS of the main loop of the algorithm. The steps of FSA are summarized in the flowchart in Fig. 1 and in the following code.

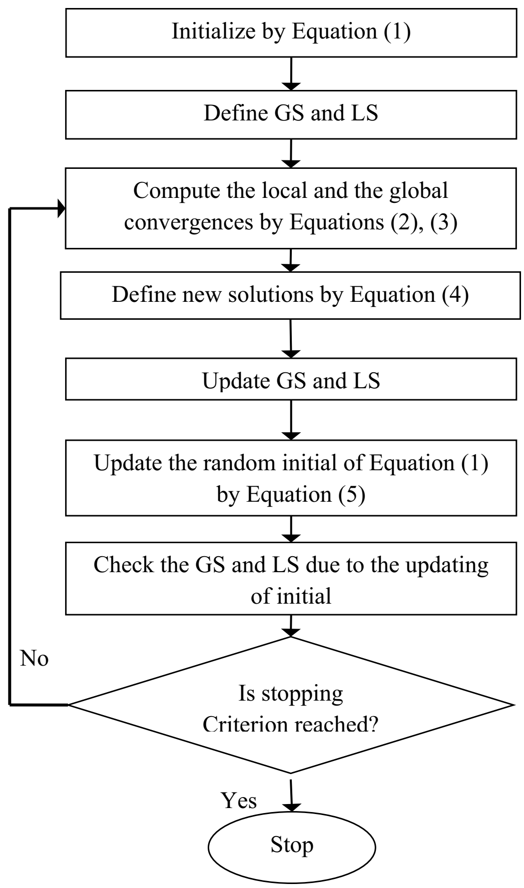


Fig. 1 The flowchart of the future search algorithm

The pseudo code of FSA.

```
Initialize by Equation (1)
Define GS and LS
while (t <Max number of iterations)
compute the local and the global convergences by Equations (2), (3) respectively
Define new solutions by Equation (4)
Update GS and LS
Update the random initial of Equation (1) by Equation (5)
Check the  GS and LS due to the updating of initial
end while
Stop
```

3 Validation and comparison

The performance of FSA is evaluated by using 23 standard benchmark functions [25]. In these benchmark functions, the ranges of their search space ‘s’, the dimension of each function ‘d’, and the minimum value of each function ‘F_{min}’ are listed in Tables 1, 2 and 3.

The proposed FSA is applied to get the optimal parameters of each objective function in Tables 1, 2 and 3 by minimizing it. The results of FSA are compared with GA, PSO, GSA, FFA, and LSA. The used settings of all algorithms are: agent size = 30 and maximum iteration number = 1000. The results of the different cases of objective functions are listed in the following sub-sections.

3.1 The case of unimodal high-dimensional functions

The unimodal functions F_1 to F_7 are listed in Table 1. In this case, the tests are carried out in Matlab over 30 runs. The best values of each unimodal function due to each optimization algorithm are listed in Table 4. The best algorithm for each benchmark function is indicated in bold type. It is clear from this table that the proposed FSA provides better results than GA, PSO, GSA, FFA, and LSA for all functions.

The effort of each optimization method to decrease each unimodal function with the variation of iteration number appears in Figs. 2, 3. These figures show that the FSA faster than other algorithms.

3.2 The case of multimodal high-dimensional test functions

In the case of multimodal functions, there are many local minima and almost multimodal functions difficult to optimize. The better results depend on the ability of the algorithm to escape from poor local optima and locate a near-global optimum. The multimodal high-dimensional test functions F_8 to F_{13} are listed in Table 2. In this case, the tests are carried out in Matlab over 30 runs. The best values of each objective function due to each optimization

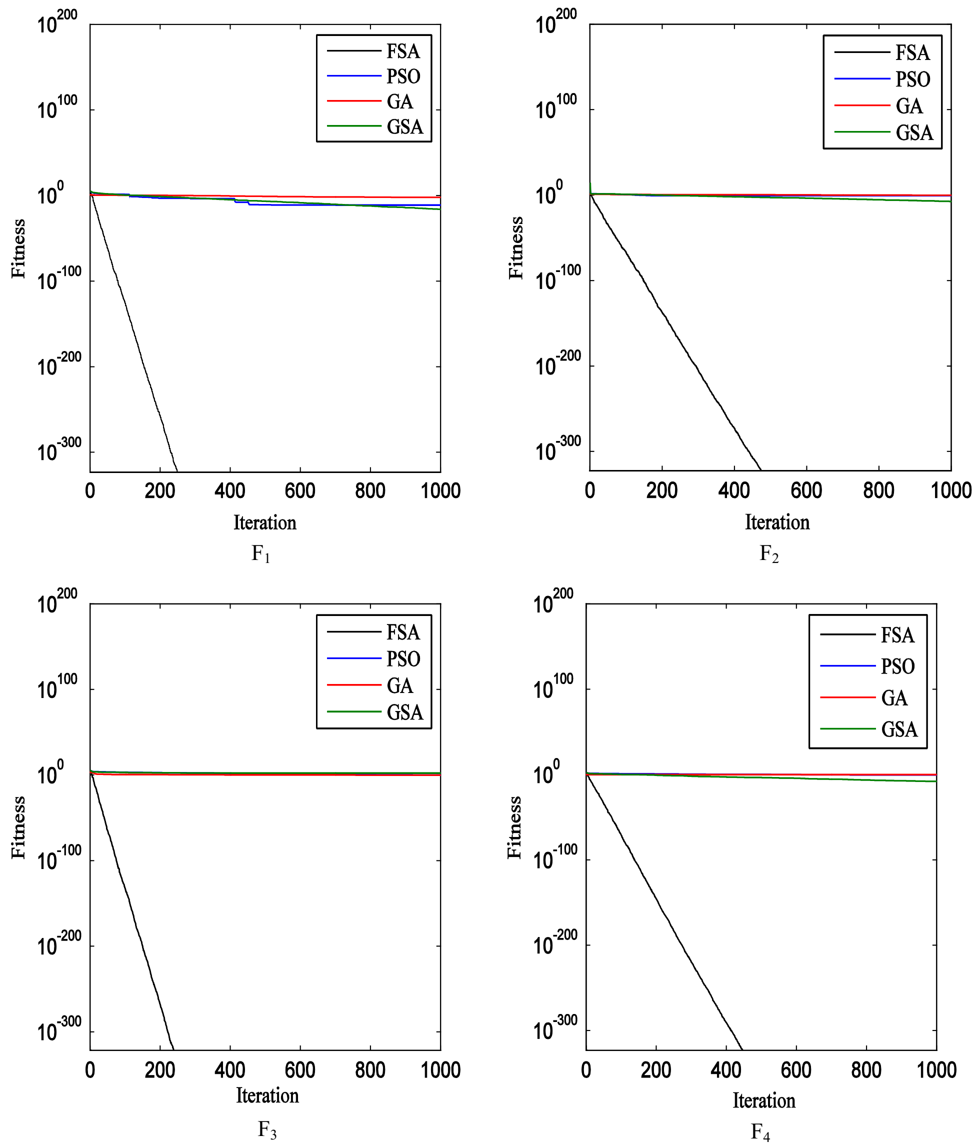


Fig. 2 Best of benchmark functions F_1–F_4

algorithm are listed in Table 5. The best algorithm for each multimodal high-dimensional function is indicated in bold type. It is clear from this table that the proposed FSA performs better solution than GA, PSO, GSA, FFA, and LSA for all functions.

The effort of each optimization method to decrease each multimodal high-dimensional function with the variation of iteration number appears in Figs. 4, 5. These figures show that the FSA still faster than other algorithms.

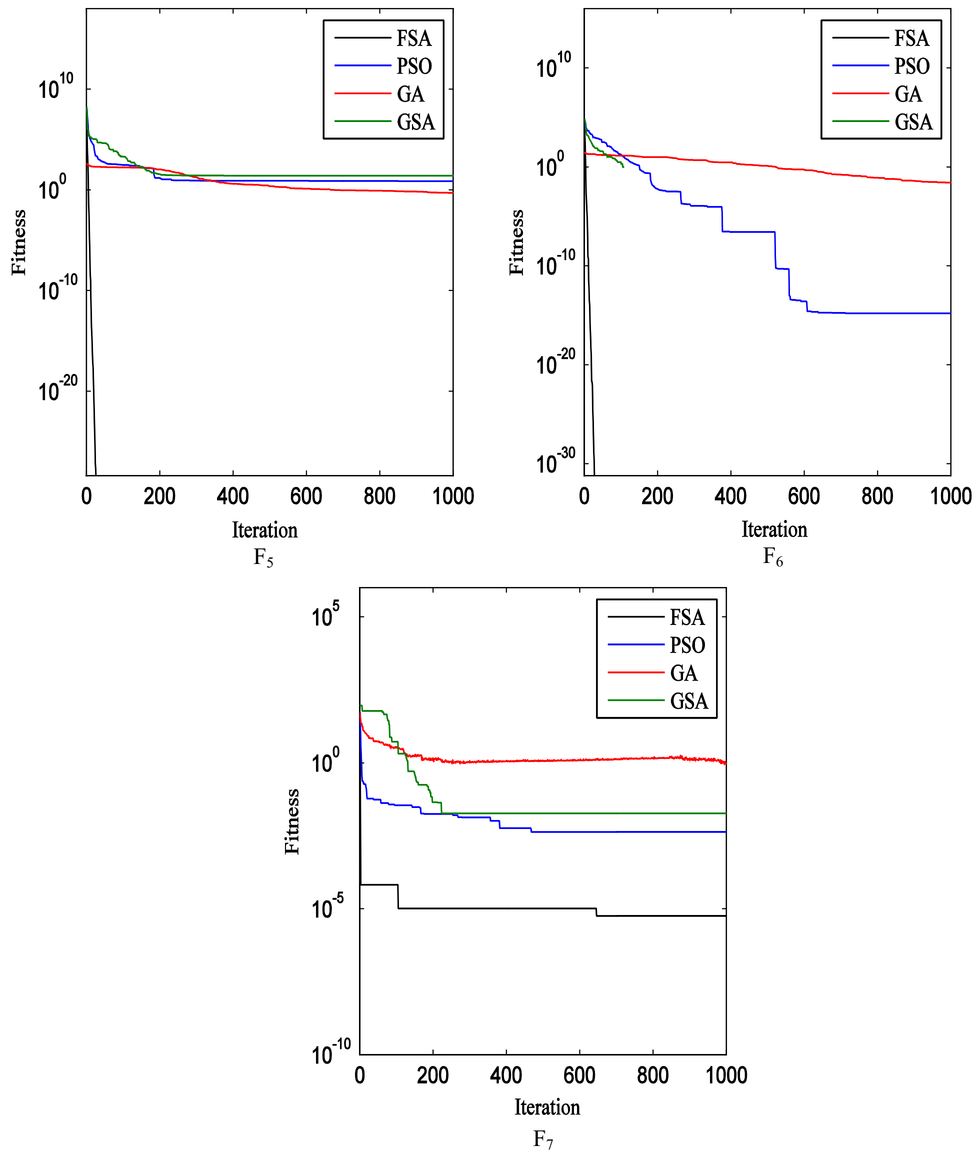


Fig. 3 Best of benchmark functions F_5–F_7

3.3 The case of multimodal low-dimensional functions

The multimodal low-dimensional test functions F_{14} to F_{23} are listed in Table 3. In this case, the tests are carried out in Matlab over 30 runs. The best values of each multimodal low-dimensional function due to each optimization algorithm are listed in Table 6. It is clear that the results of the proposed FSA, GA, PSO, GSA, FFA, and LSA have similar solutions in the most multimodal low-dimensional functions.

The effort of each optimization method to decrease each multimodal low-dimensional function with the variation of

Fig. 4 Best of benchmark functions F_8–F_{11}

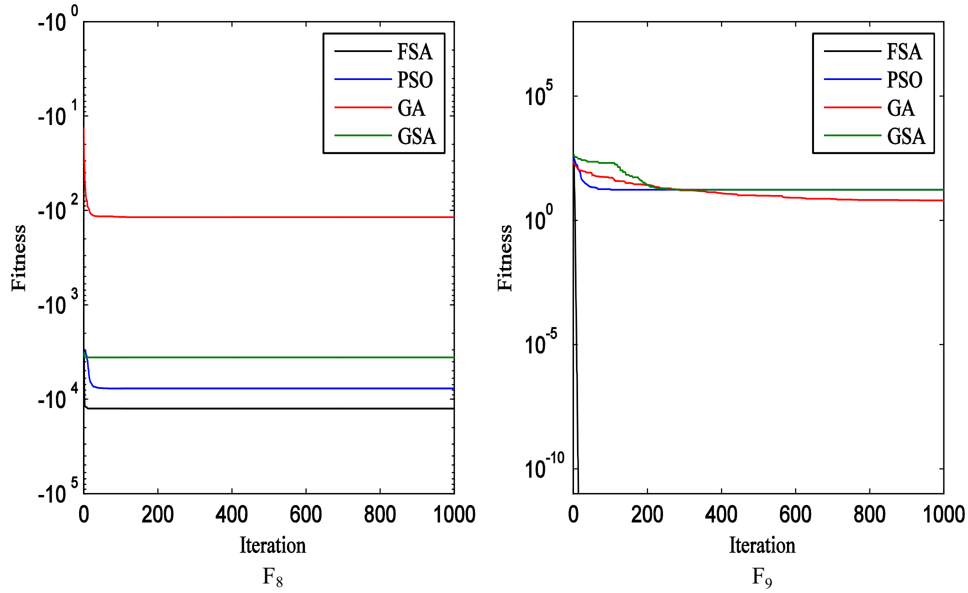


Fig. 5 Best of benchmark functions F_{12}–F_{13}

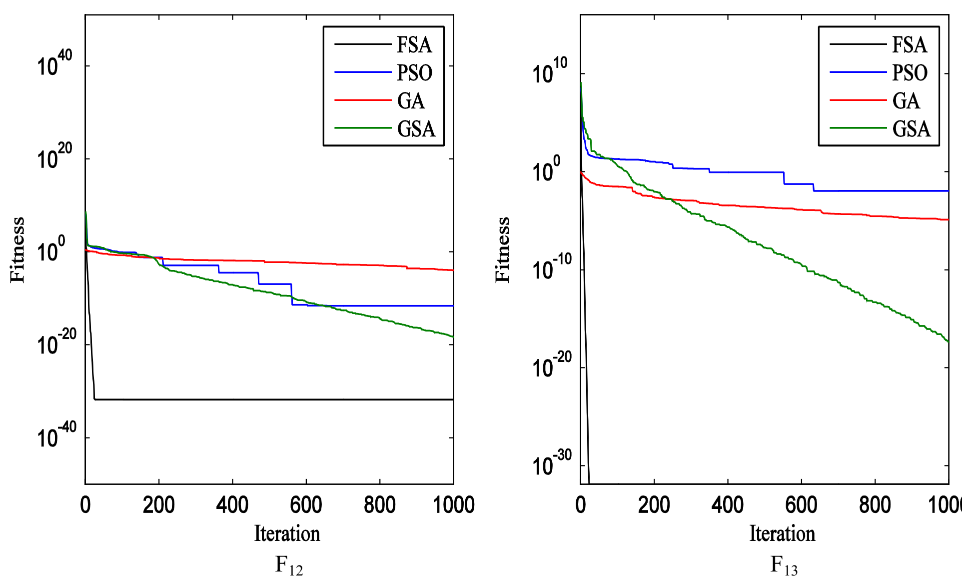
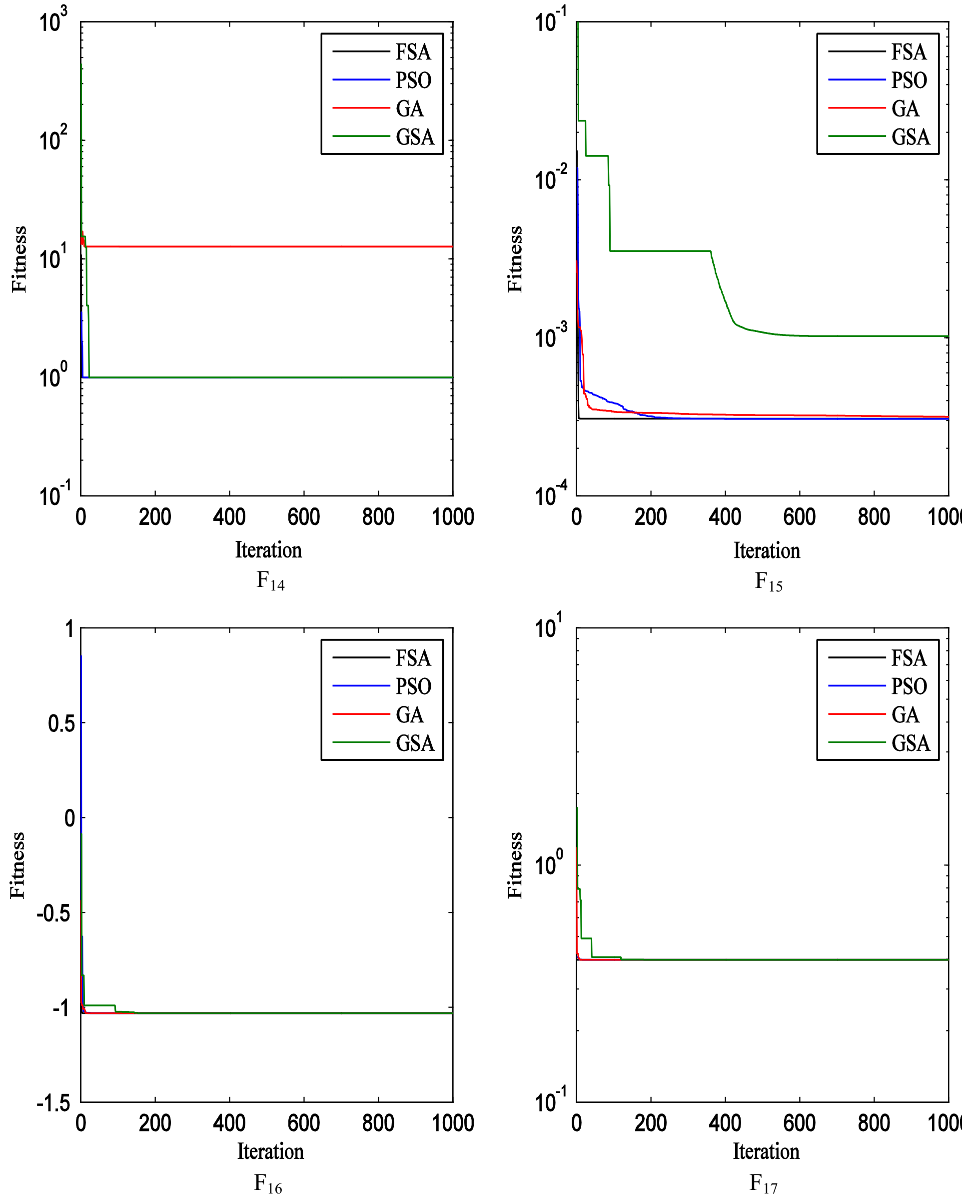


Fig. 6 Best of benchmark functions $F_{14} - F_{17}$

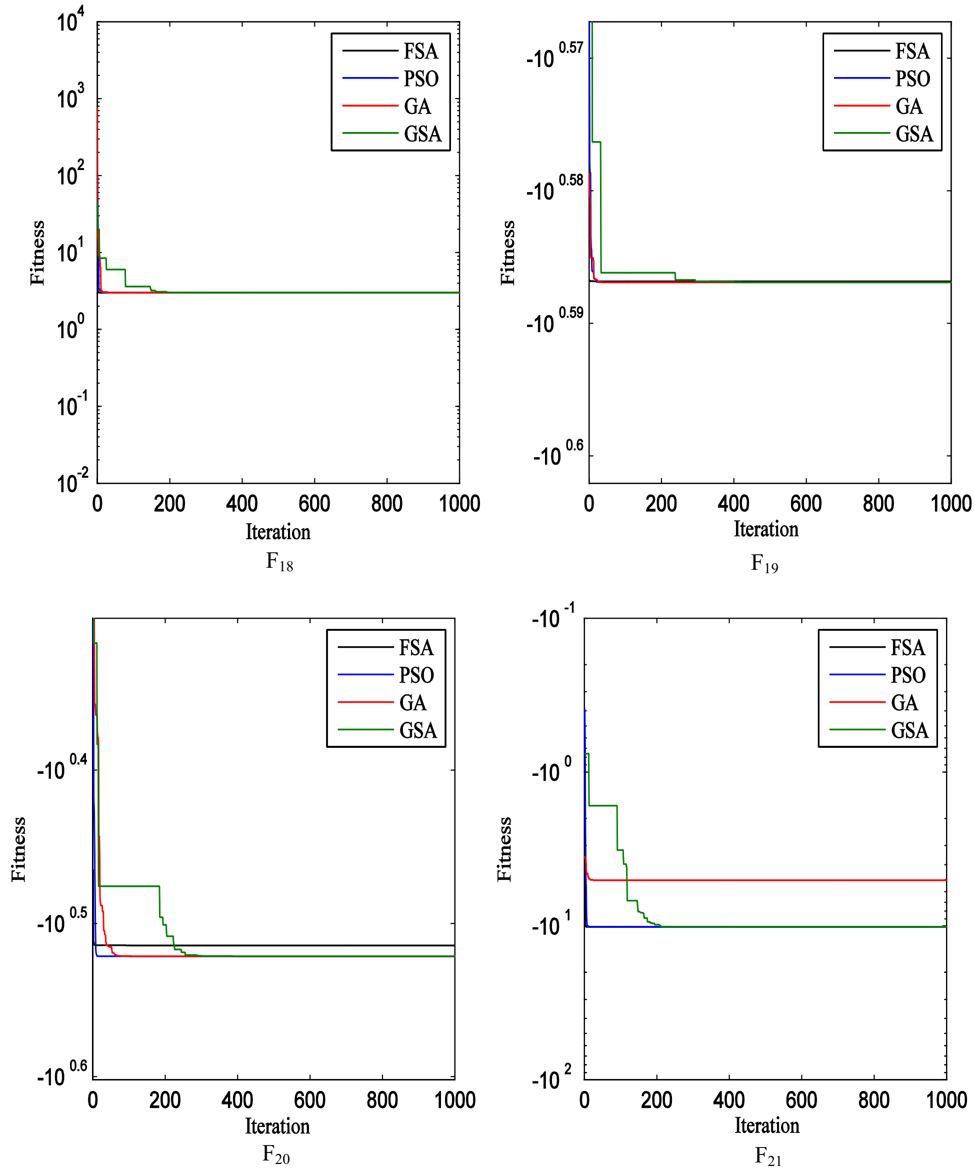


iteration number appears in Figs. 6, 7 and 8. These figures show that the results of all optimization techniques are close to each other.

4 Conclusion

In this paper, a new algorithm is proposed and it is named FSA. The main idea of this algorithm is built based on

Fig. 7 Best of benchmark functions F_{18}–F_{21}



the behavior of people to find the best life. The FSA can update the random initial and it uses the local search between multi-agents and the global search between the histories optimal agents to find the optimal solutions. The proposed FSA is validated by applying it to solve 23 benchmark functions. The results of FSA are compared with the results of the GA, PSO, GSA, FFA, and LSA. The results show that FSA outperforms the GA, PSO, GSA, FFA, and LSA GSA, FFA, and LSA to minimize the most benchmark function. Furthermore, the results proved that

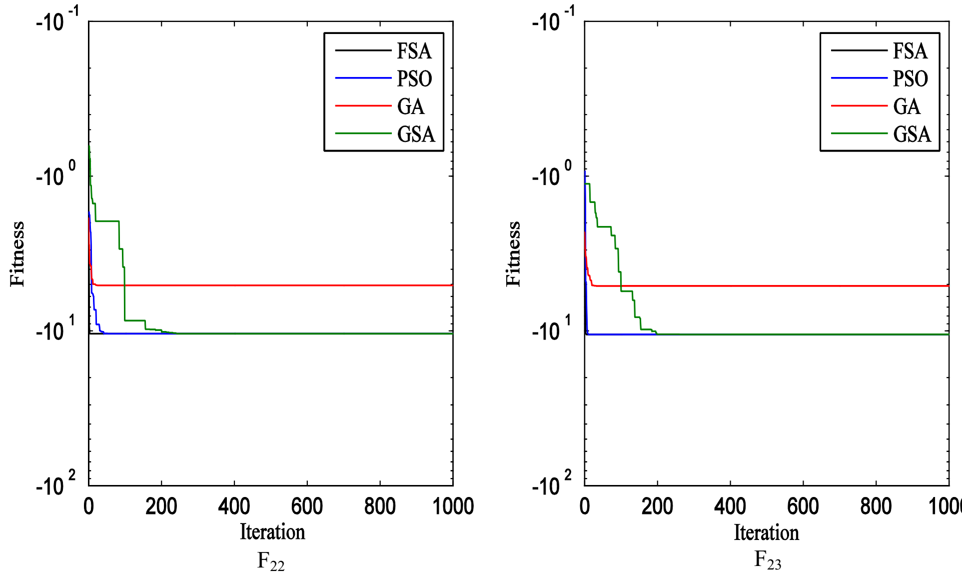


Fig. 8 Best of benchmark functions F_{22}–F_{23}

Table 1 Unimodal test functions, the ranges of their search space 's', the dimension of each function 'd', and the minimum value of each function 'F_{min}'

Unimodal test functions	d	s	F_{min}
$F_1(x) = \sum_{i=1}^{d} x_i^2$	1000	[− 100, 100]	0
$F_2(x) = \sum_{i=1}^{d} \|x_i\| + \prod_{i=1}^{d} \|x_i\|$	1000	[− 10, 10]	0
$F_3(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j\right)^2$	100	[− 100, 100]	0
$F_4(x) = \max_i\{\|x_i\|, 1 \leqslant i \leqslant n\}$	1000	[− 100, 100]	0
$F_5(x) = \sum_{i=1}^{d-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]$	30	[− 30, 30]	0
$F_6(x) = \sum_{i=1}^{d} \left(\left[x_i + 0.5\right]\right)^2$	1000	[− 100, 100]	0
$F_7(x) = \sum_{i=1}^{d} ix_i^4 + random[0, 1)$	30	[− 1.28, 1.28]	0

Table 2 Multimodal test functions, the ranges of their search space 's', the dimension of each function 'd', and the minimum value of each function 'F_{min}'

Multimodal test functions	d	s	F_{min}
$F_8(x) = \sum_{i=1}^{d} -x_i \sin\left(\sqrt{\|x_i\|}\right)$	30	[− 500, 500]	− 12,569.5
$F_9(x) = \sum_{i=1}^{d} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	30	[− 5.12, 5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos\left(2\pi x_i\right)\right) + 20 + e$	30	[− 32, 32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[− 600, 600]	0
$F_{12}(x) = \frac{\pi}{d}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{d-1}\left(y_i - 1\right)^2\left[1 + 10\sin^2\left(\pi y_{i+1}\right)\right] + \left(y_d - 1\right)^2\right\}$ $+ \sum_{i=1}^{d} u\left(x_i, 10, 100, 4\right),$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	[− 50, 50]	0
$F_{13}(x) = 0.1\left\{\sin^2\left(3\pi x_1\right) + \sum_{i=1}^{d-1}\left(x_i - 1\right)^2\left[1 + \sin^2\left(3\pi x_{i+1}\right)\right]\right.$ $\left. + \left(x_n - 1\right)^2\left[1 + \sin^2\left(2\pi x_n\right)\right]\right\} + \sum_{i=1}^{d} u\left(x_i, 5, 100, 4\right)$	30	[− 50, 50]	0

Table 3 Multimodal test functions with fix dimension, the ranges of their search space 's', the dimension of each function 'd', and the minimum value of each function 'F_{min}'

Multimodal test functions with fix dimension	d	s	F_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[− 65.536, 65.536]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[− 5, 5]	0.0003075
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[− 5, 5]	− 1.0316285
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[− 5, 10] × [0, 15]	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	[− 2, 2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[0, 1]	− 3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0, 1]	− 3.32
$F_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	− 10
$F_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	− 10
$F_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	− 10

Table 4 The best value of each objective function due to each optimization algorithm

Unimodal test functions	GA-Best	PSO-Best	GSA-Best	FFA-Best	LSA-Best	FSA-Best
F_1	0.00448	8.3377e−012	5.8560e−017	0.005027795	1.06220e−19	**0**
F_2	0.7845	0.21472	3.6345e−008	0.172154000	2.21758e−07	**0**
F_3	0.717	0.59714	197.5886	716.4716944	9.201365586	**1.535e−49**
F_4	0.755	0.24196	1.1475e−008	0.055032609	0.118431936	**0**
F_5	0.5024	7.2414	24.8082	27.85966457	0.560036507	**0**
F_6	0.0254	1.5613e−015	**0**	**0**	**0**	**0**
F_7	0.9137	0.0042529	0.0187	0.008299594	0.016268885	**5.6194e−006**

Table 5 The best value of each objective function due to each optimization algorithm

Multimodal high-dimensional function	GA-Best	PSO-Best	GSA-Best	FFA-Best	LSA-Best	FSA-Best
F_8	− 118.3588	− 7713.2454	− 3.6142e+003	− 7397.28009	−9193.9122	**− 12569.4866**
F_9	6.2594	16.9149	16.9143	11.3594362	40.7932709	**0**
F_{10}	0.0499	1.778	5.5264e−009	0.022344835	8.73041e−08	**8.8818e−016**
F_{11}	0.001	0.0050629	2.1599	0.002458997	2.22045e−16	**0**
F_{12}	2.5596e−04	2.4046e−012	4.4266e−019	9.14932e−05	2.23008e−15	**1.5705e−032**
F_{13}	1.229e−05	0.010988	4.5069e−018	0.000727556	2.75229e−19	**1.3498e−032**

Table 6 The best value of each objective function due to each optimization algorithm

Multimodal low-dimensional functions	GA-Best	PSO-Best	GSA-Best	FFA-Best	LSA-Best	FSA-Best
F_{14}	12.6705	**0.998**	0.9990	0.998003838	0.998003838	**0.998**
F_{15}	3.1625e−04	**0.00030749**	0.001	0.000443132	0.000307486	0.00030827
F_{16}	**− 1.0316**	**− 1.0316**	**− 1.0316**	**− 1.0316**	**− 1.0316**	**− 1.0316**
F_{17}	**0.3978**	**0.3978**	0.3979	**0.3978**	**0.3978**	**0.3978**
F_{18}	**3**	**3**	**3**	**3**	**3**	**3**
F_{19}	**− 3.862**	**− 3.862**	**− 3.862**	**− 3.862**	**− 3.862**	**− 3.862**
F_{20}	− 3.3219	− 3.322	− 3.322	− 3.322	− 3.322	**− 3.2689**
F_{21}	− 5.0551	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**	**− 10.1532**
F_{22}	− 5.0876	**− 10.4029**	**− 10.4029**	**− 10.4029**	**− 10.4029**	**− 10.4029**
F_{23}	− 5.1284	**− 10.5364**	**− 10.5364**	**− 10.5364**	**− 10.5364**	**− 10.5364**

The best algorithm for each multimodal low-dimensional function is indicated in bold type

FSA is faster than other methods for the most benchmark functions.

Compliance with ethical standards

Conflict of interest Authors state that there are no conflicts of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Mitchell M (1998) An introduction to genetic algorithms. MIT, London
2. Basnet C, Weintraub A (2009) A genetic algorithm for a bicriteria supplier selection problem. Int Trans Oper Res 16(2):173–187
3. Baker BM, Ayechew MA (2003) A genetic algorithm for the vehicle routing problem. Comput Oper Res 30(5):787–800
4. Gehring H, Bortfeldt A (1997) A genetic algorithm for solving the container loading problem. Int Trans Oper Res 4(5-6):401–418
5. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. Swarm Intell 1(1):33–57
6. Van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evolut Comput 8(3):225–239
7. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst 22(3):52–67
8. Das S, Biswas A, Dasgupta S, Abraham A (2009) Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: Foundations of computational intelligence, vol 3. Springer, Berlin, pp 23–55
9. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
10. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697
11. Zhu G, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl Math Comput 217(7):3166–3173
12. Montané FAT, Galvao RD (2006) A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Comput Oper Res 33(3):595–619
13. Grabowski J, Wodecki M (2004) A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. Comput Oper Res 31(11):1891–1909
14. Brandão J (2011) A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Comput Oper Res 38(1):140–151
15. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In Evolutionary computation, 2007. CEC 2007. IEEE congress on. IEEE, pp 4661–4667
16. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
17. Xing B, Gao WJ (2014) Gravitational search algorithm. In: Innovative computational intelligence: a rough guide to 134 clever algorithms. Springer, Cham, pp 355–364
18. Sharafi Y, Khanesar MA, Teshnehlab M (2016) COOA: competitive optimization algorithm. Swarm Evolut Comput 30:39–63
19. Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. Appl Soft Comput 36:315–333
20. Lin D, He L, Feng X, Luo W (2018) Niching pareto ant colony optimization algorithm for bi-objective pathfinding problem. IEEE Access 6:21184–21194
21. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
22. Xia X, Gui L, He G, Xie C, Wei B, Xing Y et al (2018) A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm. J Comput Sci 26:488–500
23. Brabazon A, Cui W, O’Neill M (2016) The raven roosting optimisation algorithm. Soft Comput 20(2):525–545
24. Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 3(1):24–36
25. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evolut Comput 3(2):82–102

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.