A Comparative Study of State Transition Algorithm with Harmony Search and Artificial Bee Colony

Xiaojun Zhou, David Yang Gao and Chunhua Yang

Abstract We focus on a comparative study of three recently developed nature-inspired optimization algorithms, including state transition algorithm, harmony search and artificial bee colony. Their core mechanisms are introduced and their similarities and differences are described. Then, a suit of 27 well-known benchmark problems are used to investigate the performance of these algorithms and finally we discuss their general applicability with respect to the structure of optimization problems.

Keywords State transition algorithm · Harmony search · Artificial bee colony

1 Introduction

Existing natural phenomena, such as natural selection and survival of the fittest (genetic algorithm), natural annealing process in metallurgy (simulated annealing), foraging behavior of real ant colonies (ant colony optimization), and social behavior of bird flocks and fish schools (particle swarm optimization) have inspired researchers to develop algorithms for optimization problems. These nature-inspired algorithms have received considerable attention due to their strong

X. Zhou (✉) · D. Y. Gao
School of Science, Information Technology and Engineering, University of Ballarat, Ballarat, VIC 3350, Australia
e-mail: tiezhongyu2010@gmail.com

D. Y. Gao
e-mail: d.gao@ballarat.edu.au

X. Zhou · C. Yang
School of Information Science and Engineering, Central South University, Changsha 410083, China
e-mail: ychh@csu.edu.cn

Z. Yin et al. (eds.), *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013*, Advances in Intelligent Systems and Computing 212,
DOI: 10.1007/978-3-642-37502-6_78,

adaptability and easy implementation. Inspired by the improvisation process of musicians and foraging behavior of real honeybees, harmony search (HS) [1–3] and artificial bee colony (ABC) [4, 5] have been proposed respectively in recent few years. At the same time, in terms of the concepts of state and state transition, a new heuristic random search algorithm named state transition algorithm (STA) has been introduced in order to probe into classical and intelligent optimization algorithms [6–9]. In this study, we focus on a comparative study of state transition algorithm with harmony search and artificial bee colony in their standard versions.

2 Three Stochastic Algorithms

In this section, we give a brief description of the three stochastic algorithms with respect to their mechanisms, and the similarities and differences are also discussed.

2.1 Harmony Search

In HS, there exist three possible choices to generate a new piece of music: (1) select a note stored in harmony memory at a probability of HMCR (harmony memory considerate rate); (2) adjust the pitch slightly at a probability of PAR (pitch adjusting rate); (3) compose any pitch randomly within bounds. The pitch is adjusted by

$$x_{new} = x_{old} + (2rand - 1) * b$$

where, *rand* is a random number from [0,1], and *b* is the pitch bandwidth.

2.2 Artificial Bee Colony

In ABC, the colony of artificial bees contains three groups of bees: (1) employed bees, going to the food source visited previously; (2) onlookers, making decision to choose a food source; (3) scouts, carrying out random search. A new position is produced by

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

where, i is the index of ith food position, j is the jth component of a position, φ_{ij} is a random number in [−1,1], k is a different index from i, and j, k are created randomly.

An artificial onlooker bee chooses a food source depending on a probability by

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n}$$

where, fit_i is the fitness value of the position i, SN is the number of food sources.

2.3 State Transition Algorithm

The unified framework of STA is described as follows

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k \\ y_{k+1} = f(x_{k+1}) \end{cases}$$

where, A_k, B_k are state transition matrix, u_k is the function of state x_k and historical states, and there are four special geometrical operators defined by

1. Rotation transformation

$$x_{k+1} = x_k + \alpha \frac{1}{n\|x_k\|_2} R_r x_k,$$

where, α is a positive constant, R_r is a random matrix with its entries from $[-1,1]$.

2. Translation transformation

$$x_{k+1} = x_k + \beta R_t \frac{x_k - x_{k-1}}{\|x_k - x_{k-1}\|_2},$$

where, β is a positive constant, R_t is a random variable from [0,1].

3. Expansion transformation

$$x_{k+1} = x_k + \gamma R_e x_k,$$

where, γ is a positive constant, R_e is a random diagonal matrix with its entries obeying the standard norm distribution.

4. Axesion transformation

$$x_{k+1} = x_k + \delta R_a x_k$$

where, δ is a positive constant, R_a is a random diagonal matrix with its entries obeying the standard norm distribution and only one random position having nonzero value.

2.4 Similarities and Difference

There are two main similarities among the three algorithms in the discussed versions: Firstly, a new solution is created randomly, and they are all stochastic algorithms. Second, "greedy criterion" is adopted to evaluate a solution, and it is different from simulated annealing, in which, a bad solution is accepted in probability.

The differences between STA and other two algorithms are: (1) both HS and ABC focus on updating each component of a solution, while STA treats a solution in whole for update except the axesion transformation; (2) the comparing STA is individual-based, while both HS and ABC are population-based; (3) the mutant operators are different in three algorithms; (4) in HS, there is a probability in choosing an update, while in STA, the updating procedures are determined; (5) in ABC, choosing a food source depending on a probability associated with the fitness, while in STA, a candidate solution with better fitness is preferred; (6) in ABC, the fitness is standardized, while in STA, the fitness is based on objective function.

3 Experimental Results

All these benchmark instances are taken from [10]. In our experiments, we use the codes of standard HS and ABC from [11, 12], and the STA is from Zhou et al. [7]. The size of the population is 10, and the maximum iterations are 1e3, 2e3, 4e3, 1e4, 5e4, and 1e5 for $n = 2, 3, 4, 10, (20, 24, 25)$ and 30, respectively. For each benchmark instance, the initial population is the same for three algorithms at each run, and 20 runs are performed for each algorithm. Statistics like mean, std (standard deviation), and Wilcoxon rank sum test are used to evaluate algorithms.

3.1 Benchmark Instances

The details of the benchmark instances are given as follows.

Ackley function

$$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e,\ -15 \le x_i \le 30$$

Beale function

$$f_2(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2, \ -4.5 \le x_i \le 4.5$$

Bohachevsky Function

$$f_3(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7, \ -100 \le x_i \le 100$$

Booth Function $f_4(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2, -10 \le x_i \le 10$

Branin Function

$$f_5(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \ -5 \le x_1 \le 10, \ 0 \le x_2 \le 15$$

Colville Function

$$f_6(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1), \ -10 \le x_i \le 10$$

Dixon and Price Function $f_7(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2, \ -10 \le x_i \le 10$

Easom Function $f_8(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$, $-100 \le x_i \le 100$

Goldstein and Price Function

$$f_9(x) = \left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right) \times \left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)\right), \ -2 \le x_i \le 2$$

Griewank Function $f_{10}(x) = \frac{1}{4{,}000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left|\frac{x_i}{\sqrt{i}}\right| + 1, \ -600 \le x_i \le 600$

Hartmann Function $f_{11}(x) = -\sum_{i=1}^{4} a_i \exp\left[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2\right], \ 0 < x_j < 1$

where,

$$a = [1, 1.2, 3, 3.2]^T, \ A = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, \ P = 10^{-4}\begin{bmatrix} 6890 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}$$

Hump Function $f_{12}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,\ -5 \le x_i \le 5$

Levy Function $f_{13}(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}\Big[(y_i - 1)^2(1 + 10\sin^2(\pi y_i + 1)) + (y_n - 1)^2(1 + 10\sin^2(\pi y_n))\Big]$

$$y_i = 1 + \frac{x_i - 1}{4},\ -10 \le x_i \le 10$$

Matyas Function $f_{14}(x) = 0.26\left(x_1^2 + x_2^2\right) - 0.48x_1x_2,\ -10 \le x_i \le 10$

Michalewics Function $f_{15}(x) = -\sum_{i=1}^{2}\sin(x_i)\sin(ix_i^2/\pi)^{2m},\ m = 10,\ 0 \le x_i \le \pi$

Perm Functions $f_{16}(x) = \sum_{k=1}^{n}\left[\sum_{i=1}^{n}(i^k + \beta)\Big((x_i/i)^k - 1\Big)\right]^2,\ \beta = 0.5,\ -n \le x_i \le n$

Powell Function $f_{17}(x) = \sum_{i=1}^{n/4}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4,\ -4 \le x_i \le 5$

Power Sum Function $f_{18}(x) = \sum_{k=1}^{n}\left[\left(\sum_{i=1}^{n}x_i^k\right) - b_k\right]^2,\ b = (8, 18, 44, 114),\ -4 \le x_i \le 5$

Rastrigin Function $f_{19}(x) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10),\ -5.12 \le x_i \le 5.12$

Rosenbrock Function $f_{20}(x) = \sum_{i=1}^{n}\Big(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\Big),\ -5 \le x_i \le 10$

Schwefel Function $f_{21}(x) = 418.9829n - \sum_{i=1}^{n}\left(x_i \sin\sqrt{|x_i|}\right),\ -500 \le x_i \le 500$

Shekel Function $f_{22}(x) = -\sum_{j=1}^{m}\left[\sum_{i=1}^{4}(x_i - C_{ij})^2 + \beta_j\right]^{-1},\ m = 10,\ 0 \le x_i \le 10$

$$\beta = \frac{1}{10}[1, 2, 2, 4, 4, 6, 3, 7, 5, 5]^T,\ C$$

$$= \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix}$$

Shubert Function $f_{23}(x) = \sum_{i=1}^{5} i\cos[(i+1)\cdot x_1 + i] \cdot \sum_{i=1}^{5} i\cos[(i+1)\cdot x_2 + i],\ -10 \le x_i \le 10$

Sphere Function $f_{24}(x) = \sum_{i=1}^{n}x_i^2,\ -5.12 \le x_i \le 5.12$

Sum Squares Function $f_{25}(x) = \sum_{i=1}^{n} ix_i^2,\ -10 \le x_i \le 10$

Trid Function $f_{26}(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1},\ -n^2 \le x_i \le n^2$

Zakharov Function $f_{27}(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4,\ -5 \le x_i \le 10.$

3.2 Results and Discussion

Test results are listed in Table 1. We can find that the results of HS are always not as good as that of ABC and STA, except for f_{11}, f_{15} and f_{23}. It seems that HS are capable of solving problems without much interaction between variables, and the solution accuracy and global search ability of HS are also not satisfactory.

Table 1 Results for three algorithms on benchmark instances

Functions	HS mean ± std	ABC mean ± std	STA mean ± std
$f_1(n = 2)$	0.14 ± 0.57-	8.88E−16 ± 0 ≈	8.88E−16 ± 0
$f_2(n = 2)$	0.35 ± 0.53-	3.61E−06 ± 1.38E−5-	4.31E−11 ± 4.91E−11
$f_3(n = 2)$	0.73 ± 0.62-	0 ± 0 ≈	0 ± 0
$f_4(n = 2)$	0.08 ± 0.14-	4.57E−17 ± 4.90E−17+	4.80E−11 ± 3.99E−11
$f_5(n = 2)$	0.39 ± 0.01-	0.39 ± 5.46E−16 ≈	0.39 ± 1.50E−16
$f_6(n = 4)$	7.20 ± 19.98-	0.21 ± 0.14-	0.001 ± 0.002
$f_7(n = 25)$	12.17 ± 5.19-	7.51E−15 ± 2.71E−15+	0.60 ± 0.20
$f_8(n = 2)$	−0.43 ± 0.49-	−0.9057 ± 0.27-	−1.0 ± 1.31E−11
$f_9(n = 2)$	11.48 ± 12.86-	3.002 ± 0.008-	3.00 ± 4.77E−9
$f_{10}(n = 2)$	0.16 ± 0.14-	0 ± 0 ≈	0 ± 0
$f_{11}(n = 3)$	−3.86 ± 2.88E−8 ≈	−3.86 ± 1.82E−15 ≈	−3.86 ± 2.96E−10
$f_{12}(n = 2)$	2.23E−5 ± 8.82E−5-	4.65E−8 ± 0 ≈	4.66E−8 ± 1.13E−10
$f_{13}(n = 30)$	0.90 ± 0.22-	4.98E−16 ± 5.39E−17+	3.84E−11 ± 4.80E−12
$f_{14}(n = 2)$	0.05 ± 0.06-	4.27E−10 ± 1.75E−9-	1.97E−250 ± 0
$f_{15}(n = 2)$	−1.8013 ± 5.44E−5 ≈	−1.8013 ± 6.83E−16 ≈	−1.8013 ± 1.01E−10
$f_{16}(n = 4)$	5.94 ± 9.22-	0.15 ± 0.14-	0.01 ± 0.03
$f_{17}(n = 24)$	10.27 ± 5.54-	1.88E−4 ± 5.94E−5 ≈	1.13E−4 ± 2.36E−5
$f_{18}(n = 4)$	0.29 ± 0.49-	0.02 ± 0.01-	4.33E−4 ± 5.02E−4
$f_{19}(n = 2)$	0.09 ± 0.30-	0 ± 0 ≈	0 ± 0
$f_{20}(n = 2)$	1.02 ± 1.49-	0.01 ± 0.01-	4.38E−8 ± 1.71E−7
$f_{21}(n = 2)$	0.03 ± 0.16-	2.54E−5 ± 0-	2.54E−5 ± 1.48E−12
$f_{22}(n = 4)$	−5.61 ± 3.41-	−10.53 ± 9.34E−5 ≈	−10.53 ± 3.06E−10
$f_{23}(n = 2)$	−186.73 ± 5.09E−4 ≈	−186.73 ± 3.57E−14 ≈	−186.73 ± 3.28E−8
$f_{24}(n = 30)$	0.72 ± 0.20-	5.08E−16 ± 5.69E−17-	0 ± 0
$f_{25}(n = 20)$	0.69 ± 0.50-	2.58E−16 ± 3.72E−17-	0 ± 0
$f_{26}(n = 10)$	−78.37 ± 113.69-	−210 ± 7.32E−7 ≈	−210 ± 1.86E−10
$f_{27}(n = 2)$	6.37E−4 ± 2.80E−3-	2.91E−18 ± 2.55E−18-	0 ± 0

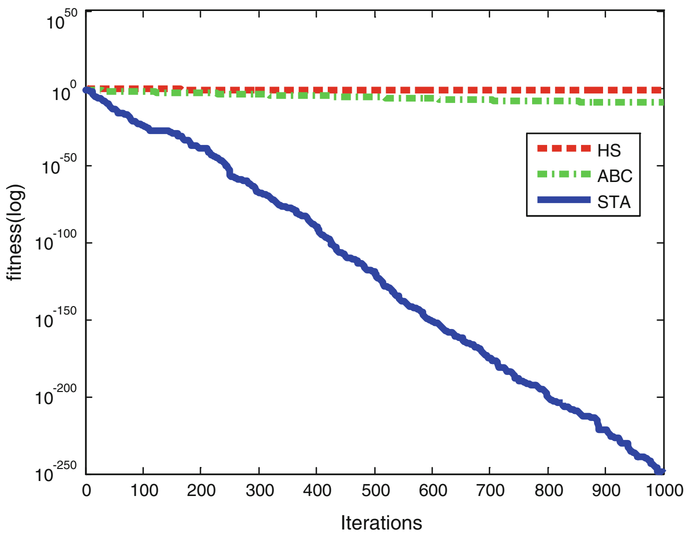

Fig. 1 The average fitness curve of Matyas function by HS, ABC and STA

For ABC and STA, we can find their results are much more satisfactory, and they are able to obtain the global solutions for the majority of the test problems. To be more specific, we can find that ABC outperforms STA for f_4, f_7 and f_{13}, and it can gain higher precision than STA, especially for f_7, which indicates that ABC are more suitable for problems with strongly interacted structure. On the other hand, for f_2, f_6, f_8, f_9, f_{14}, f_{16}, f_{18}, f_{20}, f_{24}, f_{25} and f_{27}, STA outperforms ABC in terms of solution accuracy, which indicates STA has stronger local exploitation ability than that of ABC.

Figure 1 gives the average fitness curve of Matyas function by the three algorithms. We can find that STA is more capable of searching in depth.

4 Conclusion

In this paper, we investigate the mechanisms and performances of state transition algorithm, harmony search and artificial bee colony. Similarities and differences of the algorithms are mainly focused. A suit of unconstrained optimization problems have been used to evaluate these algorithms. Experimental results show that both state transition algorithm and artificial bee colony have better global search capability and can achieve higher solution accuracy than harmony search, artificial bee colony is more capable of solving problems with strongly interacted variables, and state transition algorithm has the potential ability to search in depth.

Acknowledgments Xiaojun Zhou's research is supported by China Scholarship Council, David Yang Gao is supported by US Air Force Office of Scientific Research under the grant AFOSR FA9550-10-1-0487 and Chunhua Yang is supported by the National Science Found for Distinguished Young Scholars of China (Grant No. 61025015).

References

1. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
2. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933
3. Yang XS (2009) Harmony search as a metaheuristic algorithm. In: Music-inspired harmony search algorithm: theory and application. Springer, pp 1–14
4. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471
5. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214:108–132
6. Zhou XJ, Yang CH, Gui WH (2011) Initial version of state transition algorithm. In: 2nd international conference on digital manufacturing and automation (ICDMA), pp 644–647
7. Zhou XJ, Yang CH, Gui WH (2011) A new transformation into state transition algorithm for finding the global minimum. In: 2nd international conference on intelligent control and information processing (ICICIP), pp 674–678
8. Zhou XJ, Yang CH, Gui WH (2012) State transition algorithm. J Ind Manag Optim 8(4):1039–1056
9. Yang CH, Tang XL, Zhou XJ, Gui WH State transition algorithm for traveling salesman problem. To be published in the 31st Chinese control conference, arXiv: 1206.0329
10. http://www-optima.amp.i.kyotou.ac.jp/member/student/hedar/Hedar files/TestGO files/Page 364
11. https://sites.google.com/a/hydroteq.com/www/
12. http://mf.erciyes.edu.tr/abc/