Commun Nonlinear Sci Numer Simulat 18 (2013) 89–98

Contents lists available at SciVerse ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns

Firefly algorithm with chaos

A.H. Gandomi [a], X.-S. Yang [b], S. Talatahari [c,*], A.H. Alavi [d]

[a] Young Researchers Club, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[b] Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, UK
[c] Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran
[d] Department of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 21 August 2011
Received in revised form 10 June 2012
Accepted 12 June 2012
Available online 21 June 2012

Keywords:
Firefly algorithm
Chaos
Metaheuristic algorithm
Global optimization

ABSTRACT

A recently developed metaheuristic optimization algorithm, firefly algorithm (FA), mimics the social behavior of fireflies based on the flashing and attraction characteristics of fireflies. In the present study, we will introduce chaos into FA so as to increase its global search mobility for robust global optimization. Detailed studies are carried out on benchmark problems with different chaotic maps. Here, 12 different chaotic maps are utilized to tune the attractive movement of the fireflies in the algorithm. The results show that some chaotic FAs can clearly outperform the standard FA.

1. Introduction

The non-linearity of many optimization problems often results in multiple local optima. To cope with this issue, global algorithms are widely used [1]. Metaheuristic techniques are well-known global optimization methods. These techniques attempt to reproduce social behavior [2] or natural phenomena [3]. Intensification and diversification are important characteristics of the metaheuristic methods. Intensification searches around the current best solutions and selects the best candidate points. The diversification process allows the optimizer to explore the search space more efficiently, mostly by randomization.

Several novel metaheuristic algorithms are proposed for global search. Such algorithms can increase the computational efficiency, solve larger problems, and implement robust optimization codes [2,3]. Recently, Yang [4] developed a promising metaheuristic algorithm, called firefly algorithm (FA) at the University of Cambridge. The FA is based on the idealized behavior of the flashing characteristics of fireflies. Preliminary studies suggest that the FA can perform superiorly, compared with genetic algorithm and particle swarm optimization [4], and it is applicable for mixed variable and engineering optimization [5].

On the other hand, recent advances in theories and applications of nonlinear dynamics, especially of chaos, have drawn more attention in many fields [6]. The one of these fields is the applications of chaos in optimization algorithms [7].

Previously, Chaotic sequences have been used together with some heuristic optimization algorithms such as genetic algorithms [8], harmony search [9], simulated annealing [10], particle swarm optimization [11], Imperialist Competitive Algorithm [12], ant and bee colony optimization [13,14] and Big Bang-Chaotic Big Crunch optimization [15]. This paper introduces chaotic FA-based methods. In these algorithms, we use different chaotic systems to replace the parameters of the FA. Thus different methods that use chaotic maps as efficient alternatives to pseudorandom sequences have been proposed. In order to evaluate the proposed algorithm, a set of unimodal and multimodal mathematical benchmarks are utilized. The results reveal the improvement of the new algorithms due to the application of deterministic chaotic signals in place of constant values.

* Corresponding author. Tel.: +98 411 4432571.
E-mail addresses: a.h.gandomi@gmail.com (A.H. Gandomi), siamak.talat@gmail.com (S. Talatahari).

1007-5704/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cnsns.2012.06.009

The rest of the paper is organized as follows. Section 2 presents the descriptions of the proposed firefly algorithm. The chaotic maps that generate chaotic sequences in the FA steps are described in Section 3. Section 4 describes how to implement the simulations. In Section 5, the tuning of the attraction parameters and finding the best chaotic FA are discussed. Section 6 presents the unique features of the chaotic FA and outlines directions for further research.

2. Firefly algorithm

2.1. Firefly metaphor

The proposed algorithm mimics the social behavior of fireflies in the tropical summer sky. Fireflies communicate, search for pray and find mates using bioluminescence with varied flashing patterns. By mimicking nature, various metaheuristic algorithms can be designed. In this paper, some of the flashing characteristics of fireflies were idealized so as to develop a firefly-inspired algorithm. For simplicity, only the following three rules were used [4]:

(1) All fireflies are unisex so that one firefly will be attracted at other fireflies regardless of their sex;
(2) Attractiveness is proportional to their brightness. For any couple of flashing fireflies, the less bright one will move towards the brighter one. Attractiveness is proportional to the brightness which decreases with increasing distance between fireflies. If there are no brighter fireflies than a particular firefly, it will move randomly in the space;
(3) The brightness of a firefly is somehow related with the analytical form of the cost function. For a maximization problem, brightness can simply be proportional to the value of the cost function. Other forms of brightness can be defined in a similar way to the fitness function in genetic algorithms.

The basic steps of the FA are summarized as the pseudo code shown in Fig. 1 which consists of the three rules discussed above.

The initial positions of agents are determined randomly in the search space, as

$$x_{i,j}^{(0)} = x_{i,min} + rand \cdot (x_{i,max} - x_{i,min}), \quad i = 1, 2, \ldots, N \tag{1}$$

where $x_{i,j}^{(0)}$ determines the initial value of the ith variable for the jth agent; $x_{i,min}$ and $x_{i,max}$ are the minimum and the maximum allowable values for the ith variable.

It should be noted that there is some conceptual similarity with other algorithms such as PSO and others [16]. However, there are some fundamental differences. First, fireflies can subgroup into different groups and thus can deal with multimodal problems naturally. Secondly, in FA, the attractiveness depends on cost function and decay monotonically with distance between fireflies. Thirdly, individuals of FA have adjustable visibility and more versatility with respect to varying attractiveness: this usually leads to higher mobility and allows the search space to be explored more efficiently. Fourthly, FA includes two important limit cases and it is possible to fine-tune the algorithm so to combine the advantages of both limit cases for exploring the search space more efficiently [4].

2.2. Distance, attractiveness and limiting cases

In the firefly algorithm, there are two important issues: the variation of light intensity and formulation of the attractiveness. For simplicity, we can always assume that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function.

```
Firefly Algorithm
begin
   Objective function f(x), x = (x1, ..., xd)T
   Generate initial population of fireflies xi (i = 1,2, ...,n)
   Light intensity Ii at xi is determined by f(xi)
   Define light absorption coefficient γ
   while (t < MaxGeneration)
   for i = 1 : n all n fireflies
      for j = 1 : i all n fireflies
         if (Ij > Ii)
         Move firefly i towards j in d-dimension via L´evy flights
         end if
         Attractiveness varies with distance r via exp[-γr2]
         Evaluate new solutions and update light intensity
      end for j
   end for i
   Rank the fireflies and find the current best
   end while
   Postprocess results and visualization
end
```

Fig. 1. Pseudo code of the firefly algorithm.

As light intensity and thus attractiveness decreases as the distance from the source increases, the variations of light intensity and attractiveness should be monotonically decreasing functions. In most applications, the combined effect of both the inverse square law and absorption can be approximated using the following Gaussian form:

$$I(r) = I_0 e^{-\gamma r^2} \tag{2}$$

where I is the light intensity, I_0 is the original light intensity, and γ is the light absorption coefficient which can be taken as a constant. As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{3}$$

where β_0 is a constant and presents the attractiveness at $r = 0$.

The distance between any two fireflies i and j at xi and xj, respectively, can be defined as the Cartesian distance rij = |xi − xj|.

The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by

$$\Delta x_i = \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha\varepsilon_i, \quad x_i^{t+1} = x_i^t + \Delta x_i, \tag{4}$$

where the first term is due to the attraction, while the second term is randomization with α being the randomization parameter. Here ε_i is a vector of random numbers which are drawn from a Gaussian distribution.

It is worth pointing out that Yang [17] has recently replaced this second term with Levy distribution and has shown that it can be further improved the FA. That is, the step size is a random number drawn from:

$$L(s) = As^{-(1+\lambda)}, \quad A = \lambda\Gamma(\lambda)\sin\left(\frac{\lambda\pi}{2}\right)\frac{1}{\pi} \tag{5}$$

where $\Gamma(\lambda)$ is a Gamma function, and λ is the exponent of the distribution.

We use a function to the geometrical annealing schedule starting from the initial α_0. That is

$$\alpha = \alpha_0\theta^t \tag{6}$$

where $0 < \theta < 1$ is the randomness reduction constant. This could form important topics for further research.

From Eq. (4), it is easy to see that there exist two limiting cases when γ is small and large. When γ tends zero, the attractiveness and brightness become constant; that is to say, a firefly can be seen by all other fireflies; this is essentially a special case of particle swarm optimization. On the other hand, when γ is very large, then the attractiveness (and thus brightness) decreases dramatically, and all fireflies are short-sighted or equivalently fly in a thick foggy sky; this means all fireflies move almost randomly, which corresponds to a random search technique. In general, the firefly algorithm corresponds to the situation between these two limiting cases, and it is thus possible to fine-tune these parameters so that FA can outperform both PSO and random search. In fact, FA can find the global optima as well as all the local optima simultaneously in a very effective manner. This advantage will be demonstrated in detail later in the implementation. A further advantage of FA is that different fireflies will work almost independently; it is thus particularly suitable for parallel implementation. It is even better than GA and PSO because fireflies aggregate more closely around each optimum (without jumping around as in the case of genetic algorithms). The interactions between different subregions are minimal in parallel implementation. For FA, similar to other meta-heuristics, penalty function method can be used to handle the constraints.

3. Chaotic maps

In random-based optimization algorithms, the methods using chaotic variables instead of random variables are called chaotic optimization algorithm (COA). In these algorithms, due to the non-repetition and ergodicity of chaos, it can carry out overall searches at higher speeds than stochastic searches that depend on probabilities [18]. To fulfill this matter, herein one-dimensional, non-invertible maps are utilized to generate chaotic sets. In the following subsections, we review some of well-known one-dimensional maps.

3.1. Chebyshev map

The family of Chebyshev map is defined as follows [19]:

$$x_{k+1} = \cos(k\cos^{-1}(x_k)) \tag{7}$$

3.2. Circle map

The Circle map [20] is represented by the following equation:

$$x_{k+1} = x_k + b - (a - 2\pi)\sin(2\pi x_k)\mathrm{mod}(1) \tag{8}$$

With $a = 0.5$ and $b = 0.2$, it generates chaotic sequence in $(0,1)$.

3.3. Gauss/Mouse map

The following equations define Gaussian map [21]:

$$x_{k+1} = \begin{cases} 0 & x_k = 0 \\ 1/k\text{mod}(1) & \text{otherwise} \end{cases} \tag{9a}$$

$$1/x_k\text{mod}(1) = \frac{1}{x_k} - \left[\frac{1}{x_k}\right] \tag{9b}$$

This map also generates chaotic sequences in (0,1).

3.4. Intermittency map

The Intermittency map has two parts, one of them is linear and another one is non-linear. It is formulated as [22]:

$$x_{k+1} = \begin{cases} \varepsilon + x_k + cx_k^n & 0 < x_k \leqslant P \\ \frac{x_k - P}{1-P} & P < x_k < 1 \end{cases} \tag{10}$$

3.5. Iterative map

The iterative chaotic map with infinite collapses [23] is expressed by:

$$x_{k+1} = \sin\left(\frac{a\pi}{x_k}\right) \tag{11}$$

where $a \in (0,1)$ is a suitable parameter.

3.6. Liebovitch map

This map is introduced by Liebovitch and Toth [19]. This map is formulated as:

$$x_{k+1} = \begin{cases} \alpha x_k & 0 < x_k \leqslant P_1 \\ \frac{P - x_k}{P_2 - P_1} & P_1 < x_k \leqslant P_2 \\ 1 - \beta(1 - x_k) & P_2 < x_k \leqslant 1 \end{cases} \tag{12}$$

where $\alpha < \beta$ and they are defined as follows:

$$\alpha = \frac{P_2}{P_1}(1 - (P_2 - P_1)) \tag{13a}$$

$$\beta = \frac{1}{P_2 - 1}((P_2 - 1) - P_1(P_2 - P_1)) \tag{13b}$$

Here three equal limits are considered for the map.

3.7. Logistic map

The Logistic map is represented by the following equation. The equation appears in nonlinear dynamics of biological population evidencing chaotic behavior [24].

$$x_{k+1} = ax_k(1 - x_k) \tag{14}$$

In this equation, x_k is the kth chaotic number, with k denoting the iteration number. Obviously, $x \in (0,1)$ under the conditions that the initial $x_0 \in (0,1)$ and that $x_0 \notin \{0.0, 0.25, 0.75, 0.5, 1.0\}$. $a = 4$ is used for the experiments.

3.8. Piecewise map

The piecewise map is defined as follows [25]:

$$x_{k+1} = \begin{cases} \frac{x_k}{P} & 0 \leqslant x_k < P \\ \frac{x_k - P}{0.5 - P} & P \leqslant x_k < \frac{1}{2} \\ \frac{1 - P - x_k}{0.5 - P} & \frac{1}{2} \leqslant x_k < 1 - P \\ \frac{1 - x_k}{P} & 1 - P \leqslant x_k < 1 \end{cases} \tag{15}$$

where P is the control parameter between 0 and 0.5 and $x \in (0,1)$.

3.9. Sine map

The sine map is a unimodal map and can be formulated as [26]:

$$x_{k+1} = \frac{a}{4}\sin(\pi x_k) \tag{16}$$

where $0 < a \leqslant 4$.

3.10. Singer map

Singer map is a one-dimensional system as given below [27]:

$$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.3028.75x_k^4) \tag{17}$$

where μ is parameter between 0.9 and 1.08.

3.11. Sinusoidal map

This iterator [24] is as given below:

$$x_{k+1} = ax_k^2\sin(\pi x_k) \tag{18}$$

For $a = 2.3$ and $x_0 = 0.7$ it has the following simplified form:

$$x_{k+1} = \sin(\pi x_k) \tag{19}$$

3.12. Tent map

The tent map is very similar to the logistic map. It displays some very specific chaotic effects. This map is defined by the following equation [28]:

$$x_{k+1} = \begin{cases} \dfrac{x_k}{0.7} & x_k < 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \geqslant 0.7 \end{cases} \tag{20}$$

4. Implementation

4.1. Benchmark numerical experiments

Different chaotic FAs were benchmarked using two well-know numerical examples. The first one is a unimodal Sphere model and the second one is a multimodal Griewank's function. The examples can be formulated as:

$$\text{Sphere} \quad f(X) = \|X\| \text{ where } \|X\| = \sqrt{\sum_{i=1}^{n} x_i^2} \quad X \in [-100, 100]^{10} \tag{21}$$

$$\text{Griewank} \quad f(X) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) \quad X \in [-600, 600]^{10} \tag{22}$$

Both global optima are located at the origin ($X^* = 0, 0 \ldots, 0$) with $f(X^*) = 0$. These functions have been solved for the case of 10 dimensions.

4.2. Criterion of success

There are many criteria in the literature for evaluating the performance of the algorithms. Here, the success rate is defined as

$$S_r = 100 \times \frac{N_{successful}}{N_{all}} \tag{23}$$

where N_{all} is the number of all trials, and $N_{successful}$ is the number of trials which found the solution is successful. Here, we consider a run as a successful run when it is very near to the global optimum. It should be noted that this distance is changed for different search space. The criteria for a successful run can be defined as:

$$|X^{gb} - X^*| \leqslant (UB - LB) \times 10^{-4} \tag{24}$$

where X^{gb} is the obtained global best by the proposed algorithms; *UB* and *LB* are, respectively, upper and lower bounds.

4.3. Initialization and parameter studies

We also used 100 different runs for each setting with completely different initial conditions. The final results are found to be almost independent of the initial guess. In fact, we have used statistical measures such as mean objective values and their standard deviations to measure the performance of the algorithm, rather than relying simply on a few runs. This approach is reflected in many tables provided in the main text.

For most cases in our implementation, we have carried out some extensive sensitivity studies of parameters such as the population size and attractiveness. From our simulations, we observed that the population size $n = 25 \sim 30$ will be sufficient for most problems. With a fixed number of fireflies at each run, the benchmark functions are optimized within 1000 iterations.

For the randomization term, α should ideally be related to the actual scale of each design variable, as scales vary for different problems. In this case, it is usually a good idea to replace α by α_{Sk} where the scaling parameters S_k ($k = 1, \ldots, d$) in the d dimensions should be determined by the actual scales of the problem of interest. In our simulations, when the Gaussian distribution is replaced with the Levy distribution, we used $\lambda = 3/2$.

5. Tuning of attraction parameters using chaotic maps

As described in Section 2, the FA has two terms: attraction and random terms. Different terms used are:

$$\mathrm{Dx}_i = \text{Attraction term} + \text{Random term} \tag{25}$$

The main term of the FA is the attraction term and it involves two important parameters, γ and β. The parameter γ characterizes the variations of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the FA behaves. In theory, γ should be in $[0, \infty)$. On the other hand, the parameter γ is important, which should be linked with the characteristic distance.

The basic firefly algorithm is very efficient, but the solutions are still changing as the optima are approaching. According to the previouly used and suggested values in [4,17], we used $\beta_0 = 1$ and $\gamma = 1/\mathrm{L}$ for the standard FA, where L is the typical length of the design variables.

In the chaotic FAs, we try to tune the attraction parameters using chaotic maps. The flowchart of a schematic chaotic FA is presented in Fig. 2. The two following subsections describe how the light absorption and attractiveness coefficients are tuned, present the simulation results, and compare different chaotic FAs with each other.

5.1. Tuning light absorption coefficient, γ, with chaotic maps

It is possible to improve the solution quality by using the chaotic maps. First, tuning light absorption coefficient is performed. Here, the value of γ is replaced with different chaotic maps introduced in Section 2. In order to fulfill this, all the maps are normalized between 0 and 2. The optimization results obtained by standard FA and chaos-FAs are presented in Tables 1 and 2.

To compare the chaotic FA with standard FA and to find the most effective chaotic map, the success rate of the FAs is presented in Table 3. As it can be seen, replacing γ with some chaotic maps improves the performance of the FA. It is worth mentioning that the chaotic maps are more effective for solving unimodal sphere functions. From this table, the Sinusoidal map has the best performance in comparison with other chaotic maps and standard algorithm in both unimodal and multimodal functions.

5.2. Tuning attractiveness coefficient, β, with chaotic maps

As it is mentioned, one of the main parameters of FA is the attractiveness coefficient. Here, this value, β, is replaced with different chaotic maps to improve the FA performance. To implement the maps, all the maps are normalized between 0 and 2. The simulation results obtained by the standard FA and chaotic FAs are presented in Tables 4 and 5 for Sphere and Griewank functions, respectively.

Table 6 compares the success rates for the standard FA and FAs with different chaotic maps. It can be observed from Table 6 that the success rates obtained by the chaotic FAs are very good. As it is seen, the success rates obtained by the chaotic FAs are far better than those of the standard FA. It seems the Gauss map has the best performance in comparison with other chaotic maps in both functions and the statistical results also confirm this.

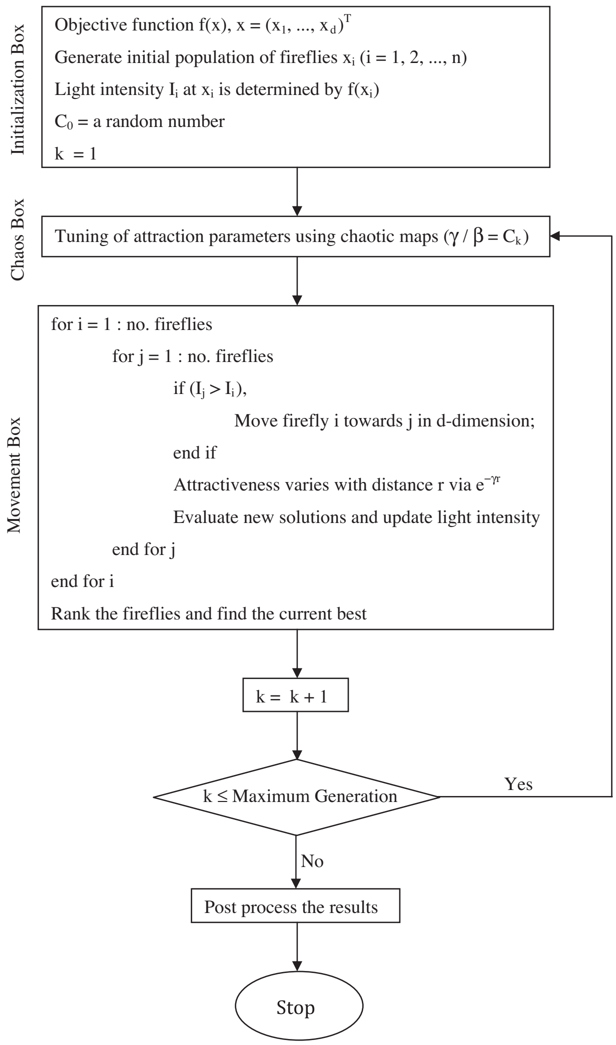

Fig. 2. Schematic flowchart of a chaotic FA.

Table 1
Statistical results for Sphere function with different light absorption coefficient.

Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time
Standard constant	2.44E−05	2.53E−03	2.07E−03	7.06E−03	1.83E−03	1.53
Chebyshev map	3.36E−01	2.24E+02	1.00E+02	1.26E+03	2.71E+02	2.15
Circle map	1.03E−04	4.17E−03	2.91E−03	1.83E−02	3.96E−03	2.02
Gauss/mouse map	2.39E−05	3.36E−03	2.79E−03	2.05E−02	3.19E−03	1.99
Intermittency map	2.01E−05	3.91E−03	2.88E−03	2.17E−02	4.24E−03	1.72
Iterative map	2.95E−06	2.28E−03	1.61E−03	9.69E−03	2.35E−03	1.99
Liebovitch map	1.90E−05	4.79E−03	1.59E−03	1.21E−01	1.29E−02	1.94
Logistic map	4.02E−05	3.66E−03	2.74E−03	1.76E−02	3.63E−03	1.97
Piecewise map	1.12E−05	2.98E−03	2.12E−03	1.82E−02	3.02E−03	1.92
Sine map	8.66E−05	3.47E−03	2.26E−03	1.85E−02	3.47E−03	2.09
Singer map	4.59E−05	2.46E−03	1.41E−03	5.98E−02	6.02E−03	1.85
Sinusoidal map	6.02E−05	1.83E−03	1.14E−03	1.03E−02	2.05E−03	1.95
Tent map	4.22E−05	2.64E−03	1.43E−03	1.57E−02	3.16E−03	2.05

5.3. Finding the best chaotic FA

As it is, tuning of the attractiveness coefficient is more effective than tuning light absorption coefficient. Thus, the best tuned FAs are those which use the Gauss map as the attractiveness coefficient. Another possible chaotic FA is that both

Table 2
Statistical results for Griewank's function with different light absorption coefficient.

Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time
Standard constant	1.23E−04	1.04E−01	7.80E−03	8.38E−01	2.40E−01	2.10
Chebyshev map	3.93E−01	2.43E+00	1.63E+00	1.11E+01	2.30E+00	2.14
Circle map	3.74E−05	1.50E−01	1.06E−02	8.71E−01	2.64E−01	2.08
Gauss/mouse map	7.06E−05	1.54E−01	1.38E−02	8.85E−01	2.61E−01	2.26
Intermittency map	3.73E−05	1.31E−01	9.94E−03	8.73E−01	2.58E−01	2.06
Iterative map	1.45E−05	1.30E−01	1.25E−02	8.52E−01	2.46E−01	1.97
Liebovitch map	5.72E−05	1.26E−01	1.05E−02	8.08E−01	2.37E−01	2.07
Logistic map	1.26E−04	1.14E−01	1.19E−02	8.89E−01	2.13E−01	2.06
Piecewise map	2.36E−05	1.41E−01	1.24E−02	8.24E−01	2.57E−01	2.02
Sine map	1.15E−04	1.10E−01	1.29E−02	8.41E−01	2.01E−01	2.20
Singer map	7.06E−05	1.50E−01	8.13E−03	1.25E+00	2.94E−01	1.83
Sinusoidal map	3.32E−05	1.06E−01	7.84E−03	8.51E−01	2.41E−01	2.14
Tent map	4.20E−05	1.13E−01	1.33E−02	8.36E−01	2.24E−01	2.03

Table 3
Success rate of FAs using standard value and different chaotic maps.

Chaotic map	Unimodal Sphere function	Multimodal Griewank's function
Standard constant	11	25
Chebyshev map	0	0
Circle map	4	24
Gauss/mouse map	8	15
Intermittency map	14	26
Iterative map	22	24
Liebovitch map	14	21
Logistic map	8	21
Piecewise map	9	15
Sine map	12	17
Singer map	14	29
Sinusoidal map	22	37
Tent map	13	18

Table 4
Statistical results for Sphere function with different attractiveness coefficient.

Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time
Standard constant	2.44E−05	2.53E−03	2.07E−03	7.06E−03	1.83E−03	1.53
Chebyshev map	2.76E−09	1.04E−08	1.04E−08	1.87E−08	2.99E−09	2.02
Circle map	4.51E−10	2.14E−05	3.14E−09	3.83E−04	6.01E−05	1.65
Gauss/mouse map	6.85E−10	5.62E−09	4.36E−09	4.46E−08	5.13E−09	1.84
Intermittency map	4.41E−09	1.21E−08	1.02E−08	4.67E−08	6.96E−09	1.98
Iterative map	3.77E−08	2.16E−04	1.65E−05	6.69E−03	8.06E−04	1.74
Liebovitch map	3.07E−09	1.12E−04	1.48E−08	9.33E−03	9.40E−04	1.88
Logistic map	2.58E−09	6.98E−07	1.94E−08	6.50E−05	6.50E−06	1.80
Piecewise map	1.48E−09	1.76E−08	1.08E−08	1.91E−07	2.43E−08	1.89
Sine map	2.60E−09	1.27E−08	1.27E−08	2.84E−08	4.70E−09	1.99
Singer map	5.61E−08	5.48E−05	1.09E−05	1.72E−03	1.85E−04	2.07
Sinusoidal map	1.18E−08	1.96E−05	3.09E−06	2.11E−04	4.04E−05	1.64
Tent map	3.25E−09	9.35E−07	3.11E−08	3.39E−05	3.94E−06	2.40

attraction parameters, γ and β, are tuned by chaotic maps. The simulation results for the algorithm using the Sinusoidal and Gauss maps, respectively, for light absorption coefficient and attractiveness coefficient are shown in Table 7. The statistical results confirm that there is no improvement when both attraction constants are replaced with the chaotic maps. From Tables 4–7, it can be observed that the best chaotic algorithm is the one using the Gauss map as its attractiveness coefficient. The performance of this algorithm is slightly better than the algorithm which uses both the Sinusoidal map and Gauss maps.

6. Discussions and conclusions

Chaos has been introduced to the standard FA to develop a chaotic FA. Twelve different chaotic maps have been investigated to tune the attraction parameters (light absorption coefficient and attractiveness coefficient) of the FA. By comparing different chaotic FAs, the algorithm which uses the Gauss map as its attractiveness coefficient is the best chaotic FA (CFA).

Table 5
Statistical results for Griewank's function with different values of the attractiveness coefficient.

Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time
Standard constant	1.23E−04	1.04E−01	7.80E−03	8.38E−01	2.40E−01	2.10
Chebyshev map	1.67E−08	8.82E−02	7.40E−03	8.94E−01	2.19E−01	2.18
Circle map	5.94E−08	1.31E−01	7.40E−03	8.77E−01	2.61E−01	1.99
Gauss/mouse map	8.43E−08	8.82E−02	2.10E−03	7.90E−01	2.09E−01	1.68
Intermittency map	4.20E−08	1.01E−01	7.60E−03	8.79E−01	2.26E−01	2.37
Iterative map	6.51E−07	1.49E−01	7.58E−03	9.11E−01	2.81E−01	1.83
Liebovitch map	3.34E−08	1.07E−01	1.42E−03	8.67E−01	2.51E−01	2.07
Logistic map	2.08E−07	1.41E−01	1.02E−02	8.83E−01	2.78E−01	1.92
Piecewise map	8.42E−08	1.22E−01	6.65E−03	8.88E−01	2.42E−01	1.85
Sine map	5.42E−08	1.24E−01	9.47E−03	8.51E−01	2.56E−01	2.05
Singer map	1.03E−06	1.49E−01	7.82E−03	1.39E+00	2.95E−01	2.38
Sinusoidal map	1.30E−06	1.57E−01	9.90E−03	8.43E−01	2.70E−01	2.02
Tent map	7.00E−08	1.92E−01	2.18E−02	9.50E−01	2.98E−01	2.32

Table 6
Success rate of FAs using standard value and different chaotic maps.

Chaotic map	Unimodal Sphere function	Multimodal Griewank's function
Standard constant	11	25
Chebyshev map	100	47
Circle map	100	45
Gauss/mouse map	100	46
Intermittency map	100	39
Iterative map	88	31
Liebovitch map	97	48
Logistic map	100	38
Piecewise map	100	41
Sine map	100	39
Singer map	97	38
Sinusoidal map	100	42
Tent map	100	29

Table 7
Statistical results for the algorithm using the Sinusoidal map and Gauss map for light absorption coefficient and attractiveness coefficient.

Function	Best	Mean	Median	Worst	Std. dev.	Ave. time	S_r
Sphere	8.23E−09	2.05E−08	1.94E−08	4.06E−08	7.12E−09	2.38	100
Griewank	5.21E−08	1.31E−01	7.40E−03	8.67E−01	2.49E−01	2.31	45

The results reveal the improvement of the new algorithm due to the application of deterministic chaotic signals in place of constant values. Statistical results and success rates of the FAs suggest that the tuned algorithms clearly improve the reliability of the global optimality and they also enhance the quality of the results.

Acknowledgements

The authors gratefully acknowledge the work and help of Engineer Parvin Arjmandi.

References

[1] Yang XS. Nature-inspired metaheuristic algorithms. Luniver Press; 2008.
[2] Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simulat 2012, in press. http://dx.doi.org/10.1016/j.cnsns.2012.05.010..
[3] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213(3–4):267–89.
[4] Yang XS. Engineering optimization: an introduction with metaheuristic applications. John Wiley & Sons; 2010.
[5] Gandomi AH, Yang XS, Alavi AH. Mixed variable structural optimization using firefly algorithm. Comput Struct 2011;89(23–24):2325–36.
[6] Pecora L, Carroll T. Synchronization in chaotic system. Phys Rev Lett 1990;64(8):821–4.
[7] Yang D, Li G, Cheng G. On the efficiency of chaos optimization algorithms for global optimization. Chaos Soliton Fract 2007;34:1366–75.
[8] Gharooni-fard G, Moein-darbari F, Deldari H, Morvaridi A. Scheduling of scientific workflows using a chaos-genetic algorithm. Procedia Comput Science 2010;1:1445–54.
[9] Alatas B. Chaotic harmony search algorithms. Appl Math Comput 2010;216:2687–99.
[10] Mingjun J, Huanwen T. Application of chaos in simulated annealing. Chaos Soliton Fract 2004;21:933–41.

[11] Alatas B, Akin E, Bedri Ozer A. Chaos embedded particle swarm optimization algorithms. Chaos Soliton Fract 2009;40:1715–34.
[12] Talatahari S, Farahmand Azar B, Sheikholeslami R, Gandomi AH. Imperialist competitive algorithm combined with chaos for global optimization. Commun Nonlinear Sci Numer Simulat 2012;17:1312–9.
[13] Gong W, Wang S. Chaos ant colony optimization and application. In: 4th International Conference on Internet Computing for Science and, Engineering, 2009, p. 301–303.
[14] Alatas B. Chaotic bee colony algorithms for global numerical optimization. Expert Syst Appl 2010;37:5682–7.
[15] Alatas B. Uniform Big Bang-Chaotic Big Crunch optimization. Commun Nonlinear Sci Numer 2011;16(9):3696–703.
[16] Glover F, Kochenberger GA. Handbook of metaheuristics. Kluwer Academic Publishers; 2003.
[17] Yang XS. Firefly algorithm, Lévy flights and global optimization, In: M. Bramer et al. editors. Research and Development in Intelligent Systems XXVI. Springer-Verlag London, 2010, p. 209–218.
[18] Coelho L, Mariani VC. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. Expert Syst Appl 2008;34:1905–13.
[19] Tavazoei MS, Haeri M. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. Appl Math Comput 2007;187:1076–85.
[20] Hilborn RC. Chaos and nonlinear dynamics: an introduction for scientists and engineers. 2nd ed. New York: Oxford Univ. Press; 2004.
[21] He D, He C, Jiang L, Zhu H, Hu G. Chaotic characteristic of a one-dimensional iterative map with infinite collapses. IEEE Trans Circuits Syst 2001;48(7):900–6.
[22] Erramilli A, Singh RP, Pruthi P, Modeling packet traffic with chaotic maps. Royal Institute of Technology, Stockholm-Kista, Sweden, 1994.
[23] May RM. Simple mathematical models with very complicated dynamics. Nature 1976;261:459–67.
[24] Li Y, Deng S, Xiao D. A novel Hash algorithm construction based on chaotic neural network. Neural Comput Applic 2011;20:133–41.
[25] Tomida AG. Matlab Toolbox and GUI for Analyzing One-Dimensional Chaotic Maps. In: International Conference on Computational Sciences and Its Applications ICCSA, IEEE Press, 2008. p. 321–330.
[26] Devaney RL. An introduction to chaotic dynamical systems. Addison-Wesley; 1987.
[27] Peitgen H, Jurgens H, Saupe D. Chaos and fractals. Berlin, Germany: Springer-Verlag; 1992.
[28] Ott E. Chaos in dynamical systems. Cambridge, UK: Cambridge University Press; 2002.