Adaptive Firefly Algorithm: Parameter Analysis and its Application

Ngaam J. Cheung[1◐], Xue-Ming Ding[2◐], Hong-Bin Shen[1]*

1 Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China, **2** School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China

Abstract

As a nature-inspired search algorithm, firefly algorithm (FA) has several control parameters, which may have great effects on its performance. In this study, we investigate the parameter selection and adaptation strategies in a modified firefly algorithm — adaptive firefly algorithm (AdaFa). There are three strategies in AdaFa including (1) a distance-based light absorption coefficient; (2) a gray coefficient enhancing fireflies to share difference information from attractive ones efficiently; and (3) five different dynamic strategies for the randomization parameter. Promising selections of parameters in the strategies are analyzed to guarantee the efficient performance of AdaFa. AdaFa is validated over widely used benchmark functions, and the numerical experiments and statistical tests yield useful conclusions on the strategies and the parameter selections affecting the performance of AdaFa. When applied to the real-world problem — protein tertiary structure prediction, the results demonstrated improved variants can rebuild the tertiary structure with the average root mean square deviation less than 0.4Å and 1.5Å from the native constrains with noise free and 10% Gaussian white noise.

Introduction

Firefly algorithm (FA) is a simple yet quite efficient nature-inspired search technique for global optimization. Since FA was developed, it has attracted a lot of attentions and becomes more popular in solving various real-world problems [1–6]. FA is a swarm-based intelligence algorithm, which mimics the flashing behavior of fireflies [7]. A firefly flashes as a signal to attract others for some purposes, e.g. predation or mating. Accordingly, this biological phenomenon is formulated as a meta-heuristic algorithm depending on following three rules [7,8]:

- All fireflies are attracted by each other without respect to their sex;
- Attractiveness is proportional to its brightness, that is, the less bright one will move towards the brighter one;
- If there are no brighter fireflies than a particular firefly, it will move randomly in the space.

Similar to other heuristic algorithms, FA also has several control parameters to be tuned, for instance, the light absorption coefficient, the randomization control factor, and the population size, for good performance on different problems. The values of these control parameters greatly determine the qualities of the

achieved solutions and the efficiency of the FA algorithm. Generally, it is a problem dependent task to select suitable control parameters for FA. As to a complex problem, it will be a hard case to deal with if the problem with numerous local optima, in which most optimization algorithms will be trapped [9–11]. Although it is highly important, there is no consistent methodology for determining the control parameters of a FA variant. Mostly, the parameters are fixed experientially or set arbitrarily within some predefined ranges.

In FA algorithm, there are two important issues: (1) the variation of light intensity; and (2) formulation of attractiveness [7]. However, these parameters are either set constants or fixed empirically in the traditional FA [7], which may make the algorithm inefficient for the problems with complex landscapes [3]. Researchers have made numerous contributions to the improvement of FA considering the alteration of the control parameters. For example, Gandomi et al. applied several chaos mechanisms to tune light absorption coefficient and attractiveness coefficient [12]. In ref. [1], a geometric progression reduction scheme for the randomization parameter was introduced in FA to enhance the solution quality. Coelho and Mariani adopted Gaussian distribution probability functions to tune the light absorption coefficient and the randomization parameter [3]. In ref. [2], a chaotic mapping operator and a chaotic component

were used to generate initial solutions and replace the random component of the standard FA, respectively. The former was to improve the quality of the initial population, while the latter was to perform ergodic search of the solution space.

Although much progress has been achieved on the FA-based algorithms since 2008, more efforts are required to further improve their performance:

- Providing the sufficient analysis for the control parameter settings;
- Efficient strategies or mechanisms for the selections of the control parameters;
- Employing heterogeneous search rules to enhance the performance of FA.

In this paper, the contributions are to develop several mechanisms and strategies for improving standard firefly algorithm, involving: (1) distance-based light absorption coefficient and gray coefficient for adaptively altering the attractiveness and enhancing difference information sharing, respectively; (2) five strategies for controlling the randomization parameter; and (3) employing heterogeneous search rules supported by the adaptively altering control parameters, to enhance the search ability of the original FA. The control parameters are adjusted over time or depending on heuristic rules, which employ the evolutionary information among the fireflies.

Methods

Firefly algorithm

The firefly algorithm (FA) is a nature-inspired optimization method [7], which maintains a population of fireflies to find the global optimum of an optimization problem.

In FA, the distance between any two fireflies i and j at $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, can be defined as the Euclidean distance r_{ij}, which is formulated as follows,

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{d=1}^{D}(x_{i,d} - x_{j,d})^2} \qquad (1)$$

where D is the dimension of an optimization problem.

Indeed, the larger the distance r_{ij} is, the less light the fireflies can see from each other. Accordingly, it is necessary to define
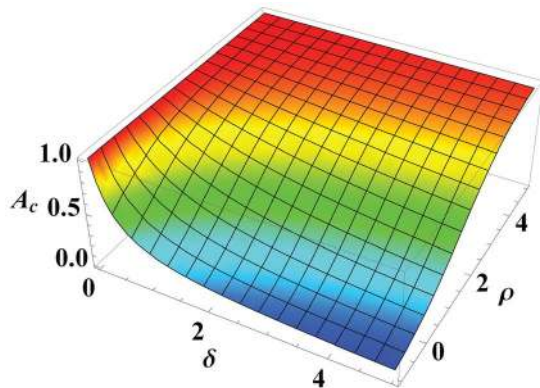

Figure 1. The relationship among A_c, σ and ρ. As σ and ρ increase, A_c will decrease while the value of A_c will sharply increase when σ and ρ are very small (or large).
doi:10.1371/journal.pone.0112634.g001

monotonically decreasing functions for light intensity and attractiveness, respectively. They are presented in Eqs. (2) and (3).

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

where I_0 is initial light intensity, and γ is the light absorption coefficient, which controls the decrease of light intensity. Accordingly, the attractiveness β of a firefly is defined as shown in Eq. (3).

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

where β_0 is a constant, which is the attractiveness at $r = 0$.

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

$$\Delta \mathbf{x}_i(t) = \beta \cdot (\mathbf{x}_i(t) - \mathbf{x}_j(t)) + \alpha(\mathbf{e}_t - \mathbf{c}) \qquad (4)$$

where $\mathbf{c}$ is a constant vector $[0.5, 0.5, \ldots, 0.5]^D$ and t is the time step, $\mathbf{e}_t$ is drawn from a normal distribution $N(0,1)$. $\Delta \mathbf{x}$ is the step size of the ith firefly moving. The first term is the attraction from the jth firefly, while the second term is randomization controlled by α, which is a constant in the range of $(0,1)$. Therefore, the update of the ith firefly is formulated as follows,

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \Delta \mathbf{x}_i(t) \qquad (5)$$

The Eqs. (4) and (5) show that the ith firefly will move towards the jth firefly, which is a more attractive one.

The procedure of FA algorithm is summarized as follows (*Algorithm S1*):

The selections of parameters are crucially important in FA, such as the light absorption coefficient γ in Eq. (3) and the randomization parameter α in Eq. (4), but the values of the control parameters are chosen in predefined ranges dogmatically [13]. During the search process, traditional FA does not alter the values of the control parameters or only use constant parameters throughout the whole process. Also the information of the search or the knowledge achieved by the fireflies are not taken into account in the selections of parameters. All these static designs may be optimal for one problem, but not efficient or even fail to guarantee convergence for another one [13]. Proper selections of these parameters highly determine the quality of the solution and the search efficiency. Although researchers have proposed many improved FA variants [1,14–16], premature convergence can still occur in the original firefly algorithm and its variants. FA may be easily trapped in local regions when it is used to deal with the complex problems with numerous local optima if the randomness is reduced to quickly [3]. To overcome these weaknesses in FA, we develop five variants of FA based on two main mechanisms, which will be described in details in following sections.

Adaptive firefly algorithm

The standard FA employs three parameters for solving the optimizations, and the parameters may result in significantly different performance of FA, such as the absorption coefficient γ and the randomization parameter α. Proper selections of these parameters can be a useful way to improve the search ability of FA. However, considering different problems with distinguish features, it is difficult to manually tune the parameters. To enhance FA, two main mechanisms and five strategies are

employed to avoid the premature convergence of the classical FA. Accordingly, five variants of FA are yielded to balance the exploitation (local search) and exploration (global search), which are denoted as AdaFa-$\mathcal{S}_s$ ($s = 1,2,\cdots,5$). The two main mechanisms are distance-based adaptive mechanism for different information sharing and gray-based coefficients for efficiently enhancing the heterogeneous search. All the two mechanisms are adaptive to exchange messages and applied to tune the control parameters in FA. Additionally, we also propose five different strategies for the selection of the randomization parameter α in Eq. (4).

Distance-based adaptive strategy. The motivation for us to investigate a distance-based strategy is that the adaptive absorption coefficient can efficiently deal with different problems whatever their landscape are, while the traditional FA uses a constant one throughout the search process.

It is obvious that there are two limited cases for a constant light absorption coefficient γ [4], which can be concluded as follows:

- The attractiveness of other fireflies will be a constant when γ approaches 0. That is, a firefly can be seen by all the other ones. In this case, FA is the same as a classical PSO.
- If $\gamma \to \infty$, the attractiveness will be equal to 0. All the fireflies cannot take their bearings to move but in random flight. In this case, FA becomes a pure random search algorithm.

As can be seen, the parameter γ is crucially important in characterizing the variations of the attractiveness, and the speed of the convergence is also dependent on γ [12]. As a result, the performance of FA will be significantly constrained when a constant γ is used to solve the optimization problems as done in traditional FA. As is well known, the attractiveness should be linked with the distance among the fireflies, and it should also vary with the different distances among the population during the search process. The information of the distances is useful for promising search adaptively. Hence, we propose to use the distance to adaptively adjust the trajectories of the fireflies. The mean distance of the ith firefly to the other fireflies is calculated as follows,

$$dis_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \sqrt{\sum_{k=1}^{D} (x_i^k - x_j^k)^2} \qquad (6)$$

where N is the number of the fireflies.

Based on the distance calculated in Eq. (6), we then define the distance ratio as follows,

$$D_r = \frac{dis_{opt} - dis_{min}}{dis_{max} - dis_{min}} \qquad (7)$$

where dis_{opt} is the distance from the ith firefly to the global best firefly. dis_{max} and dis_{min} are the maximum and the minimum distances from the i_{th} firefly to other fireflies, respectively.

Accordingly, we define an adaptive absorption coefficient based on the distance ratio, which is to adaptively track the promising flight direction. It is defined as follows,

$$A_c = \frac{1}{1 + \sigma e^{-\rho \cdot D_r}} \qquad (8)$$

where σ denotes as amplitude factor, which controls the amplitude of A_c. ρ is called contraction index. The relationship between A_c and the two factors is illustrated in Fig. 1.

In Fig. 1, the value of D_r is set to 1, which does not have effect on the lower and upper boundaries of A_c. As can be seen, the larger the values of σ and ρ are, the smaller the value of A_c is; vice versa. If we remove the effect of D_r, that is $D_r = 1$, then A_c will be a constant. As a result, it is greatly difficult to deal with the balance between exploration and exploitation. Because the light absorption coefficient is determined by the distance information adaptively, the fireflies are able to adjust their flight directions for promising search.

Then, the constant light absorption coefficient γ in traditional FA is replaced by the adaptive coefficient A_c for efficient search. The new attractiveness is obtained in Eq. (9).

$$\tau = D_r \cdot e^{-A_c r^2} \qquad (9)$$

Gray-based coefficients. Gray relational analysis (GRA) is a similarity measure for finite sequences with incomplete information [17]. Let $\mathbf{y} = (y_1, y_2, y_3, \ldots, y_D)$ be the reference sequence, and $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD}), i = 1,2,3,\ldots,N$ be the ith comparative sequence, then the gray relational coefficient between $\mathbf{y}$ and $\mathbf{x}_i$ can be defined as

$$GRC(y_j, x_{ij}) = \frac{\delta_{min} + \xi \cdot \delta_{max}}{\delta_{ij} + \xi \cdot \delta_{max}} \qquad (10)$$

where $\delta_{ij} = |y_j - x_{ij}|$, $\delta_{min} = \min_i \quad \min_j \delta_{ij}$, $\delta_{max} = \max_i \quad \max_j \delta_{ij}$, and $\xi \in (0,1]$ is employed to distinguish δ_{max} from δ_{min} and called as distinguishing constant. Based on Eq. (10), we can calculate the corresponding gray relational grade as follows

$$GRG(\mathbf{y}, \mathbf{x}_i) = \sum_{j=1}^{N} \left[\lambda_j \cdot GRC(y_j, x_{ij}) \right] \qquad (11)$$

where λ_j is a weighted constant of the gray relational coefficient $GRC(y_j, x_{ij})$ and subjects to $\sum_{j=1}^{n} \lambda_j = 1$.

To properly use the gray relational analysis, the new best firefly $\mathbf{x}_{opt}$ is employed as the reference sequence, while all the fireflies are the comparative ones. Base on these assumptions, we can use the gray relational analysis to measure the similarity between them. Let the gray relational grade between the new best firefly $\mathbf{x}_{opt}$ and the ith firefly $\mathbf{x}_i$ be GRG_i. As can be seen from Eq. (11), the closer the new best firefly $\mathbf{x_n}$ and the ith firefly are, the larger the GRG_i is. Accordingly, the relational grade GRG_i can be used to control the diversity of the firefly population. Since the gray relational grade involves the information of population distribution, we define a gray coefficient η to satisfy the requirement of diversity as follows,

$$\eta = \frac{(\eta_{min} - \eta_{max}) \cdot GRG_i + \eta_{max} GRG_{max} - \eta_{min} GRG_{min}}{(GRG_{max} - GRG_{min})} \qquad (12)$$

where η_{max} and η_{min} are the upper and the lower boundaries, which ensure the population can converge in finite time. $GRG_{max} = \max\{GRG_{i|i=1,2\ldots,N}\}$ and $GRG_{min} = \min\{GRG_{i|i=1,2\ldots,N}\}$.

Update rules. Fireflies in the traditional FA and most of its modifications, follow the same search law and share similar information throughout the search process. Due to the same search characteristics, the fireflies cannot always exhibit diverse and useful information for promising search. In an optimization
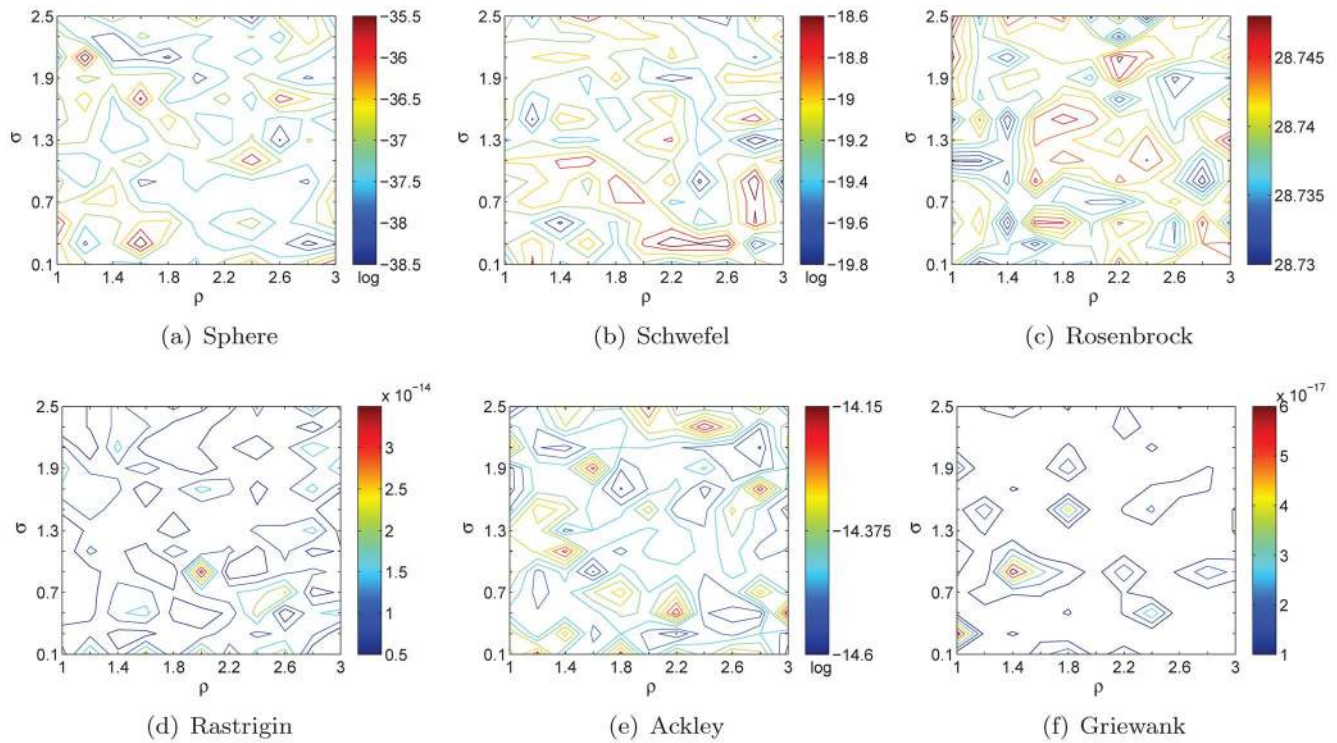
Figure 2. The relationship between the two factors (σ and ρ) and the performance of AdaFa.
doi:10.1371/journal.pone.0112634.g002

algorithm, it is important to balance the exploration and exploitation. The algorithm should make contribution to exploration initially, while focusing on exploitation and convergence in the later search process [3,18]. To accomplish the task, we introduce heterogeneous update rules to improve the search abilities of the fireflies. Two updating equations are defined and selected randomly in the search process, which are presented in Eq. (13).

$$\mathbf{x}_i(t+1) = \begin{cases} (1-\tau)\cdot\mathbf{x}_i(t) + \tau\cdot\mathbf{x}_j(t) + \mathbf{x}_r(t+1) & rand > 0.5, \\ \dfrac{N_G - t}{N_G}(1-\eta)\cdot\mathbf{x}_i(t) + \eta\cdot\mathbf{x}_{opt} & \text{else,} \end{cases} \quad (13)$$

where N_G is the maximum number of generations. $\mathbf{x}_r$ is the randomization term, which is defined based on new α-strategies as follows,

$$\mathbf{x}_r(t+1) = \alpha(t+1)\cdot(\mathbf{rand} - \mathbf{c})\cdot|\mathbf{U}_b - \mathbf{L}_b| \quad (14)$$

where $\mathbf{U}_b$ and $\mathbf{L}_b$ are the upper and lower boundaries, respectively.

In FA, the strategy ($\mathcal{S}_0$) for α is

$$\alpha(t+1) = \alpha(t)\cdot\left(\frac{10^{-4}}{0.9}\right)^{\frac{1}{N_G}} \quad (15)$$

As can be seen from Eq. (15), α decreases linearly depending on the generation number, and it does not always work well for different problems. As an important parameter controlling the randomization step, α has significant effect on the performance of

FA. To enhance its search ability, we propose five different strategies for α, which all decrease dynamically with the generation number, population size and the size of the optimized problem. These designs promise AdaFa to deal with different types of problems, and all of them increase the range of α. The strategies are presented in Eq. (16)–Eq. (20).

- strategy $\mathcal{S}_1$

$$\alpha(t) = \left(\frac{e^\delta(N_G - t)}{t\cdot\epsilon}\right)^M, \quad (16)$$

where δ is constant defined by user to clamp α, and ϵ is also a constant, which is 10^{-6}.

- strategy $\mathcal{S}_2$

$$\alpha(t) = \exp\left(\delta\left(\frac{N_G - t}{t}\right)^{\frac{1}{t}}\right)^M, \quad (17)$$

- strategy $\mathcal{S}_3$

$$\alpha(t) = \left(e^\delta t^{-\frac{N}{D}}\right)^{\frac{M(N_G - t)}{N_G}}, \quad (18)$$

where N and D are the population size and the size of an optimization problem, respectively.
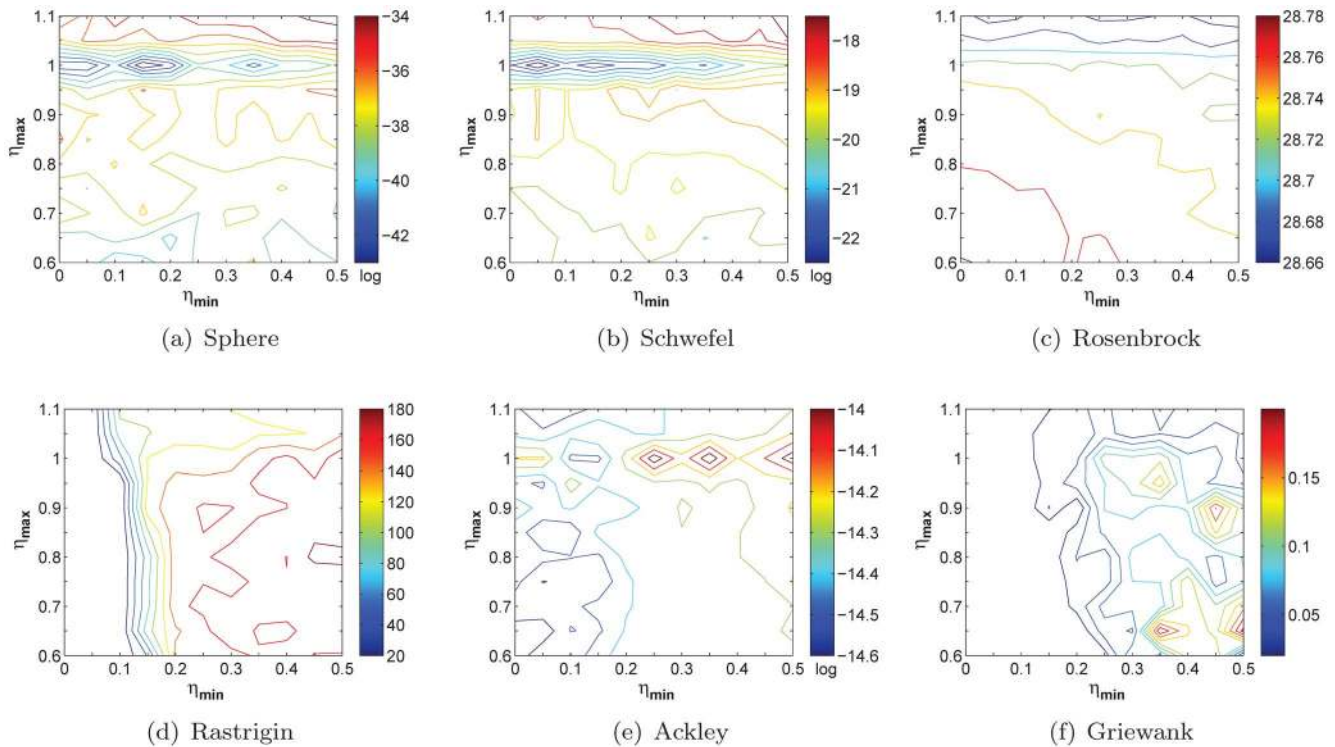
Figure 3. The relationship between η and performance of AdaFa.

- strategy $\mathcal{S}_4$

$$\alpha(t) = \left[t^{(N^P - 1)} \cdot (N_G - t) \right]^M, \qquad (19)$$

where P is the power of population size N.

- strategy $\mathcal{S}_5$

$$\alpha(t) = t^{\frac{M(N+D)(N_G - t)}{N_G}}. \qquad (20)$$

In Eq. (16)–Eq. (20), $M = -\exp\left(-\dfrac{N_G - t}{t}\right)$. From Eq. (16)–Eq.

(20), α varies with time t non-linearly, and the analysis of the parameters in the strategies are presented in following section.

Based on the designs of different adaptive mechanisms and strategies discussed above, the fireflies are allowed to learn more useful information from others and adjust the flight directions adaptively. The AdaFa algorithm is summarized in (*Algorithm S2*).

Results

In this section, we demonstrate the performance of AdaFa variants over twelve benchmark functions F_1–F_{12} summarized in Table S1 of Supplementary Materials (more details can be referred to the study [19,20]) and apply AdaFa variants to rebuild protein tertiary structure. Firstly, we conduct numerical experiments for parameters analysis and then compare AdaFa with standard particle swarm optimization (SPSO) [21], adaptive particle swarm optimization (APSO) [19], grey particle swarm optimization (GPSO) [20] and FA [8]. In numerical experiments, we analyze
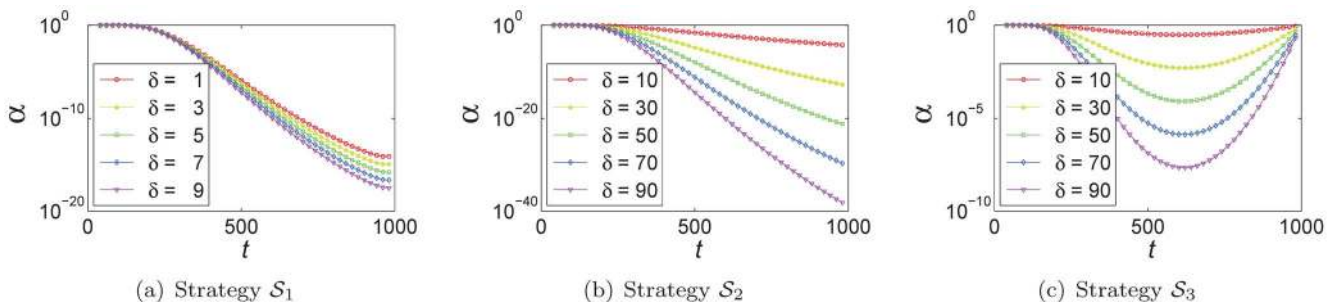

Figure 4. Different strategies for α: (a) strategy S_1, (b) strategy S_2, (c) strategy S_3 varying with different δ.
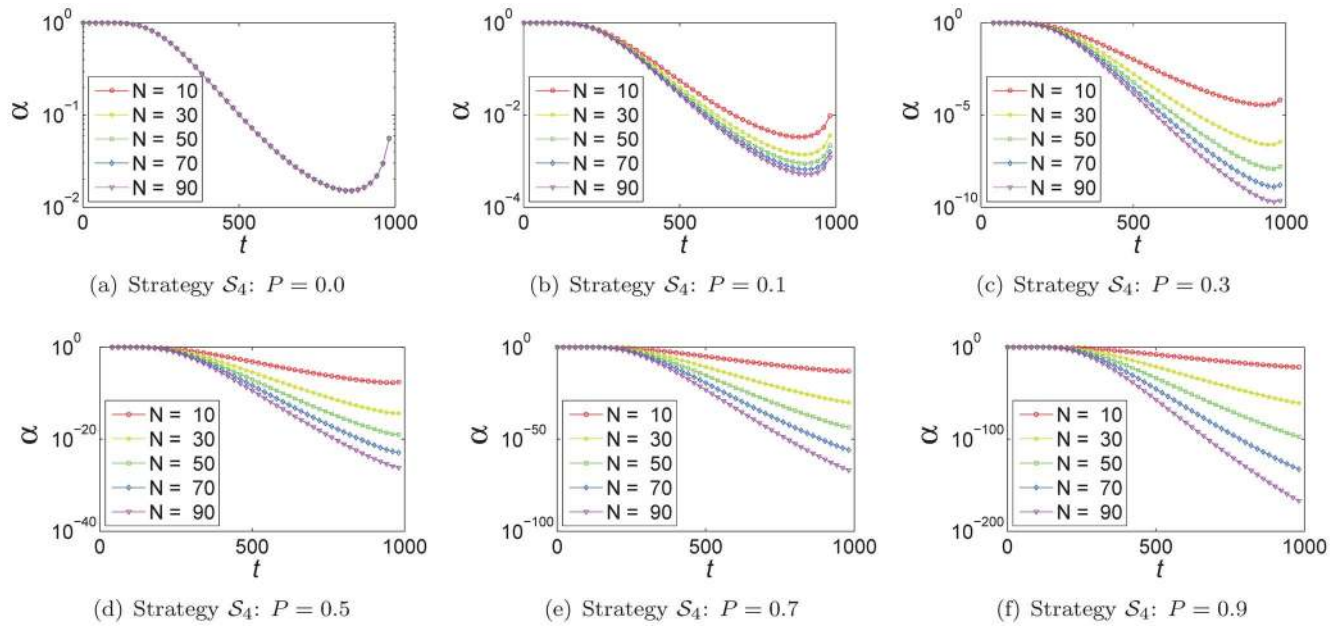
Figure 5. In strategy S_4, the relationship between α and the population size N varying with different power value P.
doi:10.1371/journal.pone.0112634.g005

the parameter sensitivity, provide the parameter settings of each algorithm, present the results of the numerical experiment, and analyze the numerical results.

Parameter sensitivity analysis

In this section, we analyze four parameters in AdaFa including the amplitude factor σ, the contraction index ρ, η_{min} and η_{max}. The different selections of the four parameters may have influence on its performance. To investigate the impacts of these parameters, we conducted the experiments on functions F_1–F_2, F_5, and F_9–F_{11}, which are Sphere, Schwefel, Rosenbrock, Rastrigin,

Ackley and Griewank functions, respectively. The maximum number of generations was set to 1,000, and the population size was 10 (This is similar to ref. [8], where the authors analyzed the effect of population size on the optimization problems). The averages of best-so-far values were used to measure the performance of AdaFa.

To investigate the amplitude factor σ and the contraction index ρ, η_{min} and η_{max} were set to $\eta_{min} = 0.05$ and $\eta_{max} = 0.95$, respectively. AdaFa was performed 20 times with different σ and ρ. σ was varied from 0.1 to 2.5 with an increment of 0.2, while ρ was changed from 1 to 3 in the same increment. Fig. 2 reveals that

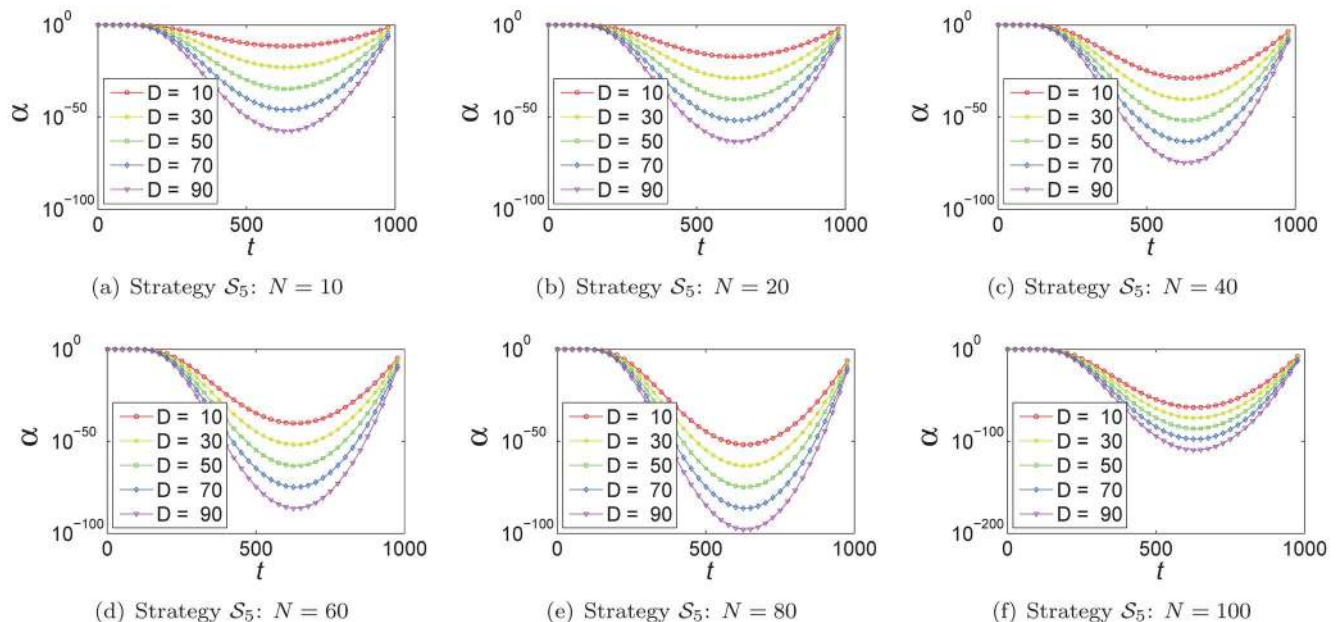**Figure 6. In strategy S_5, the relationship between α and the size of problem D with different population size N.**
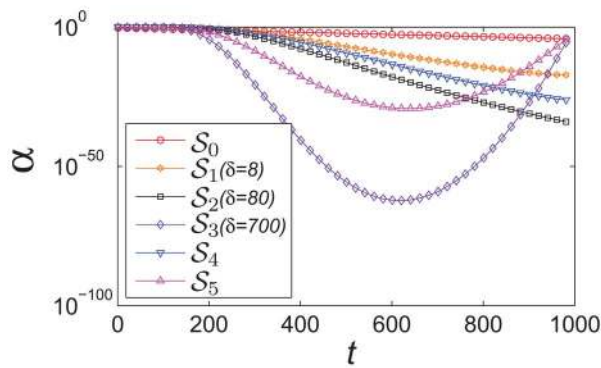doi:10.1371/journal.pone.0112634.g006

Figure 7. The comparison of different strategies for the randomization parameter α.
doi:10.1371/journal.pone.0112634.g007

- SPSO [21]: $\omega = \dfrac{1}{2log(2)}$, $c_1 = c_2 = 0.5 + log(2)$.
- APSO [19]: $\omega_{start} = 0.9$, $c_1 = c_2 = 2.0$.
- GPSO [20]: $\xi = 1$, $\omega_{min} = 0.4$, $\omega_{max} = 0.9$, $c_{min} = 1.5$, $c_{max} = c_{final} = 2.5$
- FA [8]: $\alpha = 0.5$, $\beta_{min} = 0.2$, $\gamma = 1.0$.
- IFA: $\phi = 0.05$, $\alpha = 0.5$, $\beta_0 = 1.0$, $\beta_{min} = 0.2$, $\gamma = 1.0$.
- AdaFa-$\mathcal{S}_1$: $\delta = 8$, $\sigma = 1.5$, $\rho = 2.5$, $\eta_{min} = 0.1$, $\eta_{max} = 0.95$.
- AdaFa-$\mathcal{S}_2$: $\delta = 80$, $\sigma = 1.5$, $\rho = 2.5$, $\eta_{min} = 0.1$, $\eta_{max} = 0.95$.
- AdaFa-$\mathcal{S}_3$: $\delta = 700$, $\sigma = 1.5$, $\rho = 2.5$, $\eta_{min} = 0.1$, $\eta_{max} = 0.95$.
- AdaFa-$\mathcal{S}_4$: $P = 0.75$, $\sigma = 1.5$, $\rho = 2.5$, $\eta_{min} = 0.1$, $\eta_{max} = 0.95$.
- AdaFa-$\mathcal{S}_5$: $\sigma = 1.5$, $\rho = 2.5$, $\eta_{min} = 0.1$, $\eta_{max} = 0.95$.

the relationship between the best-so-far and the two factors. As can be seen, the optimal σ and ρ were around 1.5 and 2.5, respectively. Hence, in this study $\sigma = 1.5$ and $\rho = 2.5$ were used for all the following experiments.

In the gray strategy, we employ a gray coefficient η to trade-off between the ith firefly and the best firefly. As shown in Eq. (12), it is necessary to clamp η in a fixed interval, such as $\eta_{min} \leq \eta \leq \eta_{max}$. To fix η_{min} and η_{max}, we conducted trial tests on the same six benchmark functions, and the maximum number of generations was also set to 1,000. η_{min} and η_{max} were set in $\eta_{min} \in [0, 0.5]$ and $\eta_{max} \in [0.6, 1.1]$, respectively, with the same increment of 0.05. Fig. 3 shows the relationship between η and performance of AdaFa. From Fig. 3, AdaFa will perform well when $\eta_{min} \approx 0.1$ and $\eta_{max} \approx 1.0$. Hence, in this study, we use $\eta_{min} = 0.1$ and $\eta_{max} = 0.95$ to conduct the experiments.

In strategies $\mathcal{S}_1$ $\mathcal{S}_2$ and $\mathcal{S}_3$ of α (Eq. (16)–Eq. (18)), δ can be defined by user before the optimization. How to select δ is important to α. We thus analyze the relationship between δ and α, which is illustrated in Fig. 4. As shown in Fig. 4, the larger δ is, the larger the range of α is. It is interesting to note that there is a tail in strategy $\mathcal{S}_3$ at the latter of generations, which can enhance the search abilities of the fireflies in exploitation region at latter search process.

In strategy $\mathcal{S}_4$ and strategy $\mathcal{S}_5$, the values of α are dependent on the population size N and the size D of an optimization problem, and we also analyze the relationship among them. As illustrated in Fig. 5, in strategy $\mathcal{S}_4$ the smaller P is, the more similar the trajectories of α are. As well as strategy $\mathcal{S}_1$–$\mathcal{S}_3$, the range of α is enlarged with the increment of the value of P and population size N. In strategy $\mathcal{S}_5$, as shown in Fig. 6, although the trajectories of α are different from each other varying with D, they all converge to similar points. These points are independent of the population size N.

In Fig. 7, different strategies of α are compared, where the ranges of α of all proposed strategies are larger than the strategy in standard FA. The larger ranges allow the fireflies with stronger exploration and exploitation abilities throughout the search process.

Parameters settings

Experiments were conducted to compare different algorithms on 12 benchmark functions with 30 dimensions. In these experiments, the maximum number of generation was set to 2,000, and the population of each algorithm was set to 40. According to previous analysis, the parameter settings are listed as follows:

Numerical Results

The experimental results over the twelve benchmark functions are presented in this section. Table 1 shows the mean values and the standard deviation of the results over 30 independent trials of each algorithm, and the success rate. The best are highlighted in bold, and the mean values are also illustrated in Fig. S1 in File S1 to compare the convergence of each algorithm (more details can be found in Supplementary Materials).

As illustrated in Table 1 and Fig. S1 in File S1, the five versions of AdaFa achieved better results than the other four algorithms on eight benchmark functions (F_2–F_5, F_7, and F_9–F_{11}), which can be also observed from Fig. S1 in File S1. SPSO achieved the target solution on function F_1, although the five versions of AdaFa did not as well as SPSO, they obtained the comparable results to each other, and the results were better than APSO, GPSO and FA. AdaFa-$\mathcal{S}_3$ was better than AdaFa with other four different strategies on function F_2–F_4, while other algorithms were all trapped and failed to obtain good results. It is easy to solve function F_6 by the FA-based methods as shown in Table 1, as shown FA and five variants of AdaFa all found the target solutions to the problem. Due to the complex landscape of function F_8, GPSO was the sole algorithm that found a most approaching results to the target. Similarly, AdaFa-$\mathcal{S}_1$ and AdaFa-$\mathcal{S}_4$ were the only ones that achieved sharp result than the compared algorithms on functions F_{11} and F_9, respectively. Although AdaFa-$\mathcal{S}_2$, AdaFa-$\mathcal{S}_3$ and AdaFa-$\mathcal{S}_5$ were all trapped into the local regions on function F_9, they performed so well that they all highly better than the three variants of PSO and FA on functions F_{10} and F_{11}. FA was good for the function F_{12}, on which it was superior to others, but the rest possessed similar convergent characteristic as illustrated in Fig. S1 in File S1.

Comparison of CPU efficiency

To compare the computational efficiency, we use CPU time to measure the complexity of each algorithm. For each algorithm, the computational efficiency is formulated as follows,

$$p = \frac{T_e(f)}{T_{tot}(f)} \times 100\% \tag{21}$$

where T_e is the computational time of an algorithm on the benchmark function f, while the T_{tot} is the total time of all the algorithms on function f.

In Fig. 8, the computational efficiency is illustrated. As can be seen, APSO is the fastest among all the algorithms, and it is followed by FA. Their computational efficiency are quite similar to each other over each benchmark function. The costs of the proposed five versions of AdaFa are also similar to each other, and they are all comparable to those of SPSO. However, all the five

Table 1. The results achieved by different algorithms on the benchmark functions.

Algorithm	F_1	F_2	F_3	F_4
SPSO	**0.00E+00** ± 0.00E+00	1.54E+00 ± 8.56E-01	3.55E-04 ± 3.68E-04	1.98E+00 ± 1.36E+00
APSO	2.47E-37 ± 7.47E-37	3.02E-20 ± 1.56E-19	5.88E+00 ± 7.02E+00	6.62E-01 ± 3.30E-01
GPSO	2.83E-23 ± 8.63E-23	6.15E-10 ± 1.65E-09	2.15E+01 ± 1.67E+01	1.20E+00 ± 7.70E-01
FA	1.22E-03 ± 2.53E-04	4.80E-02 ± 3.53E-02	3.67E+01 ± 2.83E+01	4.38E-02 ± 1.27E-02
AdaFa-S_1	1.56E-55 ± 4.84E-55	3.45E-28 ± 1.05E-27	1.93E-54 ± 5.11E-54	1.60E-28 ± 2.95E-28
AdaFa-S_2	2.36E-87 ± 9.56E-87	5.09E-45 ± 8.84E-45	4.19E-86 ± 9.73E-86	6.85E-45 ± 1.68E-44
AdaFa-S_3	1.26E-123 ± 3.17E-123	7.44E-63 ± 6.60E-63	2.16E-122 ± 7.92E-122	**1.30E-62** ± **1.44E-62**
AdaFa-S_4	3.51E-78 ± 1.15E-77	8.13E-40 ± 2.33E-39	8.95E-77 ± 3.37E-76	1.11E-39 ± 2.06E-39
AdaFa-S_5	3.67E-74 ± 4.93E-74	4.75E-38 ± 3.59E-38	2.47E-73 ± 4.25E-73	7.16E-38 ± 6.57E-38

Algorithm	F_5	F_6	F_7	F_8
SPSO	3.56E+01 ± 2.84E+01	7.67E-01 ± 1.01E+00	3.38E-03 ± 1.36E-03	-7.28E+03 ± 1.04E+03
APSO	2.94E+01 ± 2.71E+01	6.67E-02 ± 2.54E-01	1.17E-02 ± 3.88E-03	-2.10E+04 ± 2.96E+03
GPSO	3.92E+01 ± 2.51E+01	6.67E-02 ± 2.54E-01	1.27E-02 ± 5.15E-03	**-1.30E+04** ± 1.13E+03
FA	8.28E+01 ± 1.31E+02	**0.00E+00** ± 0.00E+00	3.50E-02 ± 3.76E-02	-7.04E+03 ± 6.91E+02
AdaFa-S_1	**2.87E+01** ± 1.82E-02	0.00E+00 ± 0.00E+00	3.00E-05 ± 3.05E-05	-4.83E+03 ± 7.70E+02
AdaFa-S_2	**2.87E+01** ± **1.74E-02**	0.00E+00 ± 0.00E+00	3.61E-05 ± 4.76E-05	-4.87E+03 ± 5.70E+02
AdaFa-S_3	2.88E+01 ± 2.08E-02	0.00E+00 ± 0.00E+00	**2.20E-05** ± **1.78E-05**	-4.96E+03 ± 6.57E+02
AdaFa-S_4	**2.87E+01** ± 2.01E-02	0.00E+00 ± 0.00E+00	4.59E-05 ± 4.18E-05	-4.90E+03 ± **5.40E+02**
AdaFa-S_5	2.88E+01 ± 2.26E-02	0.00E+00 ± 0.00E+00	3.55E-05 ± 2.96E-05	-4.94E+03 ± 6.55E+02

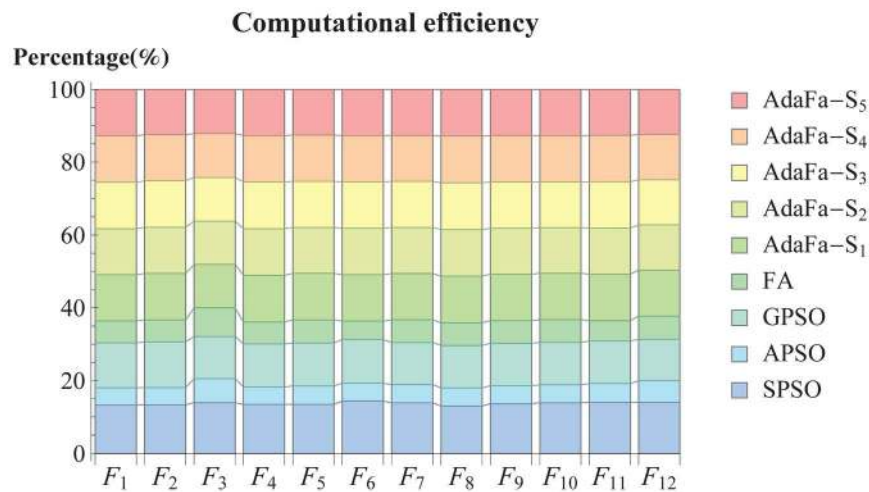| Algorithm	F_9	F_{10}	F_{11}	F_{12}
SPSO	3.48E+01 ± 3.03E+01	1.04E+00 ± 8.29E-01	6.56E-03 ± 1.12E-02	2.42E-02 ± 8.47E-02
APSO	5.00E+01 ± 1.37E+01	3.15E-01 ± 5.88E-01	1.42E-02 ± 1.73E-02	1.63E-01 ± 2.34E-01
GPSO	3.30E+01 ± 6.63E+00	8.41E-13 ± 1.06E-12	1.05E-02 ± 1.10E-02	3.11E-02 ± 8.67E-02
FA	3.49E+01 ± 1.31E+01	8.82E-03 ± 1.32E-03	2.95E-03 ± 2.24E-03	**5.87E-06** ± **1.56E-06**
AdaFa-S_1	5.55E+00 ± 3.04E+01	**4.56E-15** ± **6.49E-16**	**0.00E+00** ± 0.00E+00	8.80E-02 ± 1.69E-01
AdaFa-S_2	7.20E+00 ± 2.74E+01	8.47E-15 ± 5.00E-15	2.96E-17 ± 9.64E-17	9.33E-02 ± 2.06E-01
AdaFa-S_3	3.23E+00 ± 1.77E+01	2.78E-14 ± 2.09E-14	9.25E-16 ± 1.31E-15	6.91E-03 ± 2.63E-02
AdaFa-S_4	**3.03E-14** ± **5.33E-14**	6.10E-15 ± 1.80E-15	3.33E-17 ± 8.82E-17	1.73E-02 ± 5.50E-02
AdaFa-S_5	4.21E+00 ± 2.30E+01	1.14E-14 ± 3.55E-15	2.22E-17 ± 6.12E-17	3.80E-02 ± 1.07E-01

Figure 8. The comparison of computational efficiency of each algorithm on the test functions.
doi:10.1371/journal.pone.0112634.g008

version of AdaFa are worse than FA in terms of computational time due to several new mechanisms have been employed.

Statistical analysis

Generally, it is necessary to use the non-parametric tests to analyze the experimental results. In this section, the Friedman, Aligned Friedman and Quade tests [22] were used to validate the performances of all the algorithms.

Table 2 presents the average rankings calculated from the Friedman, Aligned Friedman, and Quade tests. Each algorithm and its score are listed in ascending order. The statistics and the corresponding p-values are shown at the bottom of the table. In terms of computed p-values, there exists significant differences among the algorithms with the 5% significance level.

Simulations on trans-membrane protein helix

Optimization approaches have been widely used in computational biology, for instance, immune algorithm (IA) was applied to discover a protein conformation with minimal energy based on lattice models [23]. Estimation of distribution algorithms (EDAs)

[24] was used to solve the PSP in simplified models. Also based on simple protein models, Islam and Chetty employed memetic algorithm (MA) with several features to accomplish the structure prediction [25]. One important step of *ab initio* PSP methods is to reconstruct the tertiary structure of a protein by some optimization algorithms. Many methods reconstruct the tertiary structure of a protein depending on the residue contact maps, which can be solved in the framework of optimizing NP-hard problem [26], for example, in [27], a stochastic method based on simulated annealing (SA) was developed to derive a three-dimensional structure from a contact map. Vassura et al. used the a heuristic method and contact map to accelerate the process [28]. As we know, the aim of PSP is to obtain the Cartesian coordinates of all the atoms, which are bonded together by inter-atomic forces called chemical bonds. It has been observed that the bond lengths subject to a Gaussian distribution with a small standard deviation in high resolution protein structural data [29], and contain the essential information to determine the backbone structure of a protein [30]. Hence, given the torsion angle and bond length constrains, conformation of the geometry of the global protein structure is also

Table 2. Average Ranks of all compared algorithms over all benchmark functions.

Average	Friedman		Aligned Friedman		Quade	
Rank	Algorithm	Score	Algorithm	Score	Algorithm	Score
1	AdaFa-S_4	3.3333	AdaFa-S_4	43.1667	AdaFa-S_4	3.2756
2	AdaFa-S_3	3.375	AdaFa-S_3	43.2083	AdaFa-S_3	3.5833
3	AdaFa-S_1	3.75	AdaFa-S_5	43.8333	AdaFa-S_5	3.891
4	AdaFa-S_5	3.8333	AdaFa-S_1	46	AdaFa-S_1	3.9167
5	AdaFa-S_2	4.3333	AdaFa-S_2	46.5833	AdaFa-S_2	4.4551
6	GPSO	6.5	APSO	59.2917	APSO	6.109
7	APSO	6.5417	GPSO	66	GPSO	6.3462
8	FA	6.625	FA	69.4583	SPSO	6.7115
9	SPSO	6.7083	SPSO	72.9583	FA	6.7115
Statistic	30.35556		10.088976		3.022701	
p-value	0.000183		0.2588364		0.004785	

doi:10.1371/journal.pone.0112634.t002

Table 3. Information of Trans-membrane protein helix.

PDB ID	Res. No.	PDB ID	Res. No.	PDB ID	Res. No.
1afoA	40	2c0xA	50	2kdrA	28
1bzkA	42	2gofA	19	2l0eA	31
1eq8A	23	2h95A	18	2l6wA	39
1javA	19	2hacA	33	2lk9A	24
1lb0A	13	2htgA	27	2lx0A	32
1lcxA	13	2jtwA	25	2xkmA	46
1pjdA	15	2jwaA	44	3c9jA	25
1wazA	46	2k9jA	42	3e86A	70
1z65A	30	2k9yA	41	3hroA	37
2beqA	36	2ka1A	35	3mraA	25

an optimization problem, where the final structure resolution is significantly dependent on the algorithms.

In this study, we applied the developed AdaFa variants on the protein structure prediction problem with focus on the transmembrane helix proteins (TMHP). Membrane proteins account for 30% of the whole genome and more than 70% known drug targets. However, because of the hydrophobic environment of TMHP, they are the extreme difficult targets for the experimental structure biology studies [31,32].

In high resolution protein structural data, it has been investigated that the bond lengths and angles subject to Gaussian distribution with a small standard deviation [33]. Here, we try to rebuild the tertiary structure directly from the bond lengths and angles with AdaFa. The data set of TMHP is filtered from the RCSB protein data bank (PDB) benchmark with 30% sequence identity cutoff, and it is released after Jan. 1st, 2010. We use 30 TMHPs for the tests, where the number of the residues (Res. No.) is less than or equal to 100, and the atomic resolutions of all the proteins are better than 2.5Å, without missed internal residues and sequence redundancies. These benchmark proteins are presented in Table 3.
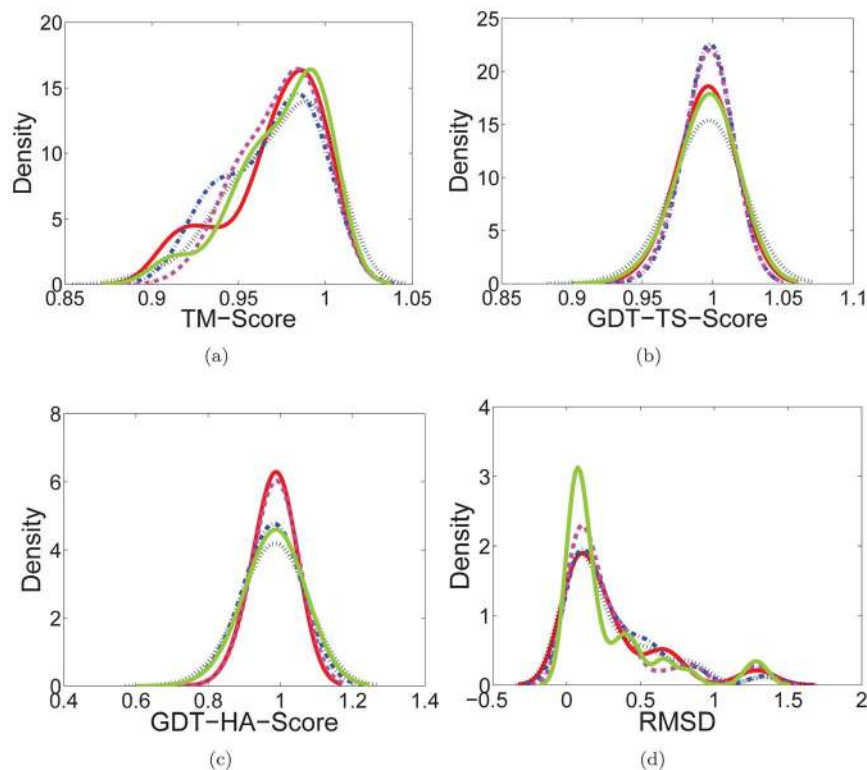

Figure 9. The kernel smoothing density estimate of (a) TM-Score, (b) GDT-TS-Score, (c) GDT-HA-Score, and (d) RMSD achieved over the native constrains. AdaFa-S_1–AdaFa-S_5 were represented by red solid line, black dotted line, blue dotted dashed line, magenta dashed line, and green solid line, respectively.
doi:10.1371/journal.pone.0112634.g009

From the 30 PDB structural data set, we extract only the angles of protein backbone including bond angles and torsion angles, which will be used as the constrains of the construction by the proposed AdaFa. In the experiments, the coordinate of the first backbone C_α atom was set randomly, the second one was determined by the standard bond length, the third one was fixed by two standard bond lengths and a bond angle, and the fourth one was calculated by the former three bond lengths, two bond angles and a dihedral angle. From the fifth C_α atom, the coordinates were determined by the same constrains involving four bond lengths, three bond angles and two dihedral angles. When considering the whole process as an optimization problem, we define the objective function as an energy, which is to calculate the distance between the undetermined C_α atom and its former four fixed C_α atom. For example, there are four determined coordinates of C_α atoms ($C_i, i \geq 4$), and we need to determine the coordinate of the $(i+4)$th C_α atom using the bond lengths, bond angles, and dihedral angles among the C_α atom C_i, C_{i+1}, C_{i+2} and C_{i+3} in the backbone of a chain. We transform the obtained knowledge into distance-based constrains in geometrical respect, and the distances can be denoted as $d_{i,j}$, $d_{i+1,j}$, $d_{i+2,j}$ and $d_{i+3,j}$ ($j = 1, 2, 3, 4$). Accordingly, the energy can be formulated as $E = \sum_{j=1}^{4} (d_{i,j} - d_{i,j}^o)$, where $d_{i,j}^o$ is transformed from the native constrains. The energy is used as the criterion of the proposed AdaFa in the optimization process. Hence, the backbone of a protein can be achieved by AdaFa iteratively. We then use the entire backbone of a protein chain as input of the PULCHRA [34] to determine the other atoms, such as hydrogen atoms and nitrogen atoms.

As illustrated in Fig. S2 in File S1, the tertiary structures of the proteins listed in Table 3 were predicted by AdaFa with high accuracy from the native constrains of each protein. In the experiments over 30 proteins, the averaged RMSDs of AdaFa-S_1–AdaFa-S_5 are 0.3074Å, 0.2960Å, 0.3043Å, 0.3079Å, and 0.2838Å, respectively. To validate the robustness of AdaFa, the native constrains with 10% Gaussian white noise were used to predict the tertiary structures. According to the results in Fig. 9 and Fig. S3 in File S1, it can be seen that AdaFa variants (AdaFa-S_1–AdaFa-S_5) are able to construct the tertiary structure of the protein with high accuracy and good robustness, where the averaged RMSDs of AdaFa-S_1–AdaFa-S_5 are 1.4425Å, 1.164Å, 1.2946Å, 1.0354Å, and 1.3508Å, respectively.

The different scores (TM-Score [35], GDT-TS-Score [36] and GDT-HA-Score [37]) and RMSD are illustrated by kernel smoothing density estimate [38] in Fig. 9 (More details can be found in Supporting Information). From Fig. 9 (a), the proposed AdaFa with five various strategies achieved different TM-Scores, in which AdaFa-S_1, AdaFa-S_4 and AdaFa-S_5 were better than the rest two variants of AdaFa. AdaFa-S_4 was superior to the other one in terms of GDT-TS-Score and GDT-HA-Score as shown in Fig. 9 (b) and Fig. 9 (c), while the AdaFa-S_2 was the worst one among the variants of AdaFa over the protein benchmark dataset. On the other hand, AdaFa-S_1 and AdaFa-S_3 exhibited a little different and competed with each other in the two score items. In respect of RMSD as shown in Fig. 9 (d), AdaFa-S_5 occupied an

overwhelming position, which was far better than the other four AdaFa variants, among which AdaFa-S_1–AdaFa-S_3 possessed similar density estimation.

Conclusion

In this paper, we develop an adaptive firefly algorithm (AdaFa) and its five variants to enhance the search ability of the original FA. In AdaFa, we propose a distance-based technique to overcome the two main drawbacks in using a constant light absorption coefficient, which tunes the light among the fireflies dynamically to control the sharing distance information leading to the variation of the attractiveness. Simultaneously, the differences among the fireflies can be adequately used to enhance the local search ability of each firefly, hence we employ the gray relational analysis to design a gray coefficient as another self-adaptively altering parameter. According to the designed parameters, AdaFa uses heterogeneous update laws to accomplish the balance between the exploitation and the exploration throughout the search process.

In the numerical experiments, we compared the performances of all five proposed AdaFa variants with FA and other three PSO-based algorithms, and the statistical results demonstrated the five AdaFa variants were significantly better than the other four algorithms with the 5% significance level. The experiments on the reconstruction of the tertiary structure of the protein showed that AdaFa had the potential ability in predicting the helix structures with high accuracy and good robustness. Although AdaFa exhibited good performance on either the numerical experiments or real-world application on the prediction of the proteins' tertiary structures, it is still a challenging problem to deal with the cooperation among the fireflies for further improving the performance of FA, which is our future efforts. The codes of AdaFa is available upon request at http://www.csbio.sjtu.edu.cn/bioinf/AdaFa-PAA/.

Supporting Information

Algorithm S1　The FA algorithm.
(TXT)

Algorithm S2　The AdaFa algorithm.
(TXT)

File S1　Combined file of supporting figures and tables.
Figure S1: The mean value over the benchmark functions with 30-dimensions. Figure S2: Simulation results over thirty proteins. Figure S3: The kernel smoothing density estimates of different measurement metrics. Table S1: Benchmark Functions.
(ZIP)

Author Contributions

Conceived and designed the experiments: NJC XMD HBS. Performed the experiments: NJC. Analyzed the data: NJC XMD. Wrote the paper: NJC XMD HBS. Method development and implementation: NJC HBS.

References

1. Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Computers & Structures 89: 2325–2336.
2. Kazem A, Sharifi E, Hussain FK, Saberi M, Hussain OK (2013) Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Applied Soft Computing 13: 947–958.
3. dos Santos Coelho L, Mariani VC (2013) Improved firefly algorithm approach applied to chiller loading for energy conservation. Energy and Buildings 59: 273–278.
4. Miguel LFF, Lopez RH, Miguel LFF (2013) Multimodal size, shape, and topology optimisation of truss structures using the firefly algorithm. Advances in Engineering Software 56: 23–37.
5. Amiri B, Hossain L, Crawford JW, Wigand RT (2013) Community detection in complex networks: Multiobjective enhanced firefly algorithm. Knowledge-Based Systems.
6. Su Z, Li L, Peng H, Kurths J, Xiao J, et al. (2014) Robustness of interrelated traffic networks to cascading failures. Sci Rep 4.

7. Yang XS (2008) Nature-Inspired Metaheuristic Algorithms. Luniver Press.

8. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Stochastic Algorithms: Foundations and Applications, SAGA 2009. volume 5792, pp. 169–178.

9. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation 10: 281–295.

10. Cheung NJ, Ding XM, Shen HB (2014) OptiFel: A convergent heterogeneous particle swarm optimization algorithm for Takagi-Sugeno fuzzy modeling. Fuzzy Systems, IEEE Transactions on 22: 919–933.

11. Li L, Peng H, Kurths J, Yang Y, Schellnhuber HJ (2014) Chaos-order transition in foraging behavior of ants. Proceedings of the National Academy of Sciences 111: 8392–8397.

12. Gandomi A, Yang XS, Talatahari S, Alavi A (2013) Firefly algorithm with chaos. Communications in Nonlinear Science and Numerical Simulation 18: 89–98.

13. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation 10: 646–657.

14. Yang XS (2010) Firefly algorithms for multimodal optimization. ArXiv e-prints.

15. Yang XS, Deb S (2010) Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. In: Gonzlez J, Pelta D, Cruz C, Terrazas G, Krasnogor N, editors, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer Berlin Heidelberg, volume 284 of Studies in Computational Intelligence. pp. 101–111.

16. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-inspired Computation 2: 78–84.

17. Deng JL (1989) Introduction to grey system theory. J Grey Syst 1: 1–24.

18. Engelbrecht AP (2010) Heterogeneous particle swarm optimization. In: Swarm Intelligence, Springer Berlin Heidelberg, volume 6234 of Lecture Notes in Computer Science. pp. 191–202.

19. Zhan ZH, Zhang J, Li Y, Chung HH (2009) Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39: 1362–1381.

20. Leu MS, Yeh MF (2012) Grey particle swafrm optimization. Applied Soft Computing 12: 2985–2996.

21. Clerc M (2012). Standard particle swarm optimisation. http://clerc.maurice.free.fr/pso/SPSO_descriptions.pdf.

22. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1: 3–18.

23. Cutello V, Nicosia G, Pavone M, Timmis J (2007) An immune algorithm for protein structure prediction on lattice models. IEEE Transactions on Evolutionary Computation 11: 101–117.

24. Santana R, Larranaga P, Lozano J (2008) Protein folding in simplified models with estimation of distribution algorithms. IEEE Transactions on Evolutionary Computation 12: 418–438.

25. Islam M, Chetty M (2013) Clustered memetic algorithm with local heuristics for ab initio protein structure prediction. IEEE Transactions on Evolutionary Computation 17: 558–576.

26. Breu H, Kirkpatrick DG (1998) Unit disk graph recognition is NP-hard. Computational Geometry 9: 3–24.

27. Vendruscolo M, Kussell E, Domany E (1997) Recovery of protein structure from contact maps. Folding and Design 2: 295–306.

28. Vassura M, Margara L, Lena PD, Medri F, Fariselli P, et al. (2008) Reconstruction of 3D structures from protein contact maps. IEEE/ACM Transactions on Computational Biology and Bioinformatics 5: 357–367.

29. Engh RA, Huber R (1991) Accurate bond and angle parameters for X-ray protein structure refinement. Acta Crystallographica Section A 47: 392–400.

30. Wu S, Zhang Y (2008) ANGLOR: A composite machine-learning algorithm for protein backbone torsion angle prediction. PLoS ONE 3: e3400.

31. Shen HB, Chou JJ (2008) MemBrain: Improving the accuracy of predicting transmembrane helices. PLoS ONE 3: e2399.

32. Yang J, Jang R, Zhang Y, Shen HB (2013) High-accuracy prediction of transmembrane inter-helix contacts and application to GPCR 3D structure modeling. Bioinformatics 29: 2579–2587.

33. Engh RA, Huber R (2006) Structure quality and target parameters, John Wiley & Sons, Ltd. doi:10.1107/97809553602060000695.

34. Rotkiewicz P, Skolnick J (2008) Fast procedure for reconstruction of full-atom protein models from reduced representations. Journal of Computational Chemistry 29: 1460–1465.

35. Zhang Y, Skolnick J (2004) Scoring function for automated assessment of protein structure template quality. Proteins: Structure, Function, and Bioinformatics 57: 702–710.

36. Zemla A, MJFK, Venclovas Č (1991) Processing and analysis of casp3 protein structure predictions. Proteins S3: 22–29.

37. Zemla A (2003) LGA: a method for finding 3D similarities in protein structures. Nucleic Acids Research 31: 3370–3374.

38. Bowman AW, Azzalini A (1997) Applied Smoothing Techniques for Data Analysis. Oxford University Press, USA.