

# A Hybrid Firefly Algorithm for Constrained optimization and Engineering Application

Wen Long

Guizhou Key Laboratory of Economics System Simulation  
Guizhou University of Finance & Economics  
Guiyang, 550004, CHINA  
lw6822@163.com

Tiebin Wu

Department of Energy and Electrical Engineering  
Hunan University of Humanities, Science & Technology  
Loudi, 417000, CHINA  
wutiebin81@163.com

**Abstract**—Firefly algorithm (FA) has been recently proposed as a stochastic optimization method and it has so far been successfully applied in a variety of fields, especially for unconstrained optimization problems. FA as most population-based algorithm is good at identifying promising area of the search space, but less good at fine-tuning the approximation to the minimization. A novel hybrid firefly algorithm (HFA) based on Rosenbrock's local search method for constrained numerical and engineering optimization problem that relies on a feasibility-based rule for constraint-handling. Good-point-set method was used to initiate individual position, which strengthened the diversity of global searching. The comparison results with other stochastic optimization algorithms demonstrate that HFA with the embedded local search technique proves to be extremely effective and efficient at locating optimal solutions.

**Keywords**—firefly algorithm; Rosenbrock's local search; constrained optimization; engineering application

## I. INTRODUCTION

Constrained global optimization problems are frequently appear in many science and engineering disciplines, such as welded beam design, tension/ compression spring design, pressure vessel design, and so on. The general constrained global optimization problem that we are interested in can be formulated as

$$\begin{aligned} \min \quad & f(\vec{x}), \vec{x} = (x_1, x_2, \dots, x_n) \\ \text{s.t.} \quad & g_j(\vec{x}) = 0, j = 1, 2, \dots, p, \\ & g_j(\vec{x}) \leq 0, j = p+1, \dots, m. \\ & l_i \leq x_i \leq u_i, i = 1, 2, \dots, n. \end{aligned} \quad (1)$$

where  $f(\vec{x})$  is an objective function,  $x_i (i=1, 2, \dots, n)$  is the decision variables,  $g_j(\vec{x})=0$  is the equality constraints,  $g_j(\vec{x}) \leq 0$  is the inequality constraints,  $l_i$  and  $u_i$  are the lower bound and the upper bound of  $x_i$ , respectively.

During the past two decades, using meta-heuristic algorithms to solve constrained optimization problems has attracted a lot research interest, and a large number of modified meta-heuristic algorithms have been developed [1]. Firefly algorithm (FA) has been recently developed by Yang [2]. This algorithm is based on the behavior of firefly. FA has showed promising performance when applied to unconstrained

optimization problems [3]. Conversely, within the area of constrained optimization problems, only a few papers have been published to date. Therefore, aim of this research is to show that FA can be applied to this kind of optimization problems as well. Although the FA is good at exploring the search space and locating the region of global minimum, it is slow at exploiting the solutions. Thus, an effective hybrid firefly algorithm is proposed to solve constrained numerical and engineering application problems.

## II. FIREFLY ALGORITHM

FA is a bio-inspired meta-heuristic algorithm for optimization problems. It was introduced in 2009 at Cambridge University by Yang [2]. FA was based on the idealized behavior of the flashing characteristics of fireflies. FA uses the following three idealized rules:

- 1) All fireflies are unisex so that one firefly is attracted to other fireflies regardless of their sex;
- 2) The attractiveness of a firefly is proportional to its brightness and they both decrease with distance. For two flashing fireflies, the less bright one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly;
- 3) The brightness or light intensity of a firefly is determined by the value of the objective function a given problem.

In FA, each firefly has a location  $X = (x_1, x_2, \dots, x_n)$  in a  $n$ -dimensional space and a light intensity  $I(x)$  or attractiveness  $\beta(x)$  which are proportional to objective function  $f(x)$ . Attractiveness  $\beta(x)$  and light intensity  $I(x)$  are relative and these should be judged by the other fireflies. Thus, it will vary with the distance  $r_{ij}$  between firefly  $i$  and firefly  $j$ . Therefore, the attractiveness  $\beta$  of a firefly can be defined by

$$\beta = \beta_0 e^{-\kappa r^2} \quad (2)$$

where  $\beta_0$  is attractiveness at  $r=0$ ,  $\kappa$  is light absorption coefficient, and  $r$  is the distance between any two firefly  $i$  and  $j$  at  $x_i$  and  $x_j$ , respectively is the Cartesian distance  $r_{ij} = \|x_i - x_j\|$ .

For any given two fireflies  $x_i$  and  $x_j$ , the movement of firefly  $i$  is attracted to another more attractive (brighter) firefly  $j$

is determined by

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\kappa r_{ij}^2} (x_j^t - x_i^t) + \alpha (\text{Rand} - \frac{1}{2}) \quad (3)$$

where  $\alpha$  is a significance factor of randomization parameter and Rand with uniform distribution  $U(0,1)$  is a random number obtained from uniform distribution.

### III. HYBRID FIREFLY ALGORITHM

#### A. Initialization

For population-based meta-heuristic algorithm, the good initial population plays an important role in searching the global optimum. Traditionally, FA generates initial population by uniform random sampling. Unfortunately, the random sampling technique often causes over-sampling in some areas and sparse sampling in others. As a result, the initial population generated from random sampling has low uniformity. However, the uniformity and diversity of population are critical to searching efficiency [4]. Therefore, in this research, we introduce using good point set method sampling technique to generate a good set of initial population of firefly algorithm. The basic idea of this approach is detailed description in [5].

#### B. Rosenbrock's local search method

The Rosenbrock method (RM) is a classical derivative-free local search technique with adaptive search orientation and size. The Rosenbrock local search technique is described in Fig. 1, where  $d_1, d_2, \dots, d_n$  are  $n$  vectors forming the orthonormal basis and  $\delta_1, \delta_2, \dots, \delta_n$  are  $n$  step lengths.

---

#### Algorithm 1. Rosenbrock's local search method

---

Initialize the starting point  $x^0$ , the step size  $\delta_i (i = 1, 2, \dots, n)$ , the initial orthonormal basis  $d (i = 1, 2, \dots, n)$ , the step size adjustment factor  $\alpha$  and  $\beta$ , the termination parameter  $\varepsilon > 0$ ,  $k = 0$ .

**Repeat**

$x = x^k$

**Repeat**

**For**  $i = 1, 2, \dots, n$

$y = x + d_i \delta_i$

**If**  $y$  is better than  $x$

$x = y$ ;  $\delta_i = \alpha \delta_i$

**Else**

$\delta_i = \beta \delta_i$

**End if**

**End for**

**Until** at least one successful and one unsuccessful move in each direction, then set  $k = k + 1$ ,  $x^k = x$ , update the orthonormal basis  $d_i$  and reset  $\delta_i$ .

**End** ( $\|x^{k+1}\| - \|x^k\| \leq \varepsilon$  or  $\delta_{\min} < \varepsilon$ ).

---

Fig. 1. Pseudo code of the Rosenbrock's local search method.

The initial stage of the RM local search begins with the coordinate axes as the search directions. This continues until there has been at least one successful and one unsuccessful step in each search direction. Once this occurs, the current stage terminates. If in one direction  $d_i$  an improvement is found, the step size  $\delta_i$  will be multiplied by  $\alpha > 1$ ; if no improvement is

found in the direction, the step size will be multiplied by  $-1 < \beta < 0$ . The factor  $\beta$  can not only reduce the step size in that direction, but also cause the search change to the opposite direction in the next loop. The values of the step adjustment factors recommended by Rosenbrock are  $\alpha = 3.0$  and  $\beta = -0.5$  [6].

As soon as at least one successful and one unsuccessful move in each direction is carried out, the algorithm updates its orthonormal basis to reflect the cumulative effect of all successful steps in all directions. It also resets the step sizes to their original values. It is described as follows [6]:

$$x^{k+1} - x^k = \sum_{i=1}^n \lambda_i d_i \quad (4)$$

where  $\lambda_i$  is the sum of successful step sizes along  $d_i$ .

Then the Gram-Schmidt orthonormalization procedure is adopted to update the search direction as:

$$q_i = \begin{cases} p_i, & i = 1, \\ p_i - \sum_{j=1}^{i-1} \frac{q_j^T p_i}{q_j^T q_j} q_j, & i \geq 2. \end{cases} \quad (5)$$

After normalization, the updated search directions are:

$$d_i = \frac{q_i}{\|q_i\|}, i = 1, 2, \dots, n. \quad (6)$$

#### C. Constraint handling approach

It is necessary to note that FA is unconstrained optimization method that needs additional mechanisms to deal with constraints when solving constrained optimization problems. Motivated by [7], a feasibility-based rule is employed in this paper to handle constraints which consist of the following rules:

Rule 1: Any feasible solution is preferred to any infeasible solution.

Rule 2: Between two feasible solutions, the one having the better objective function value is preferred.

Rule 3: Between two infeasible solutions, the one having the smaller sum of constraint violation is preferred. This sum is calculated as:

$$\sum_{j=1}^p \max(0, |g_j(\bar{x})| - \varepsilon) + \sum_{j=p+1}^m \max(0, g_j(\bar{x})) \quad (7)$$

where  $\varepsilon$  is a small tolerant value (a very small positive value).

Based on the above rule, objective function and constraint violation information are considered separately. Consequently, penalty factors are not used at all.

#### D. Framework of proposed hybrid algorithm

After explaining the main elements of hybrid firefly algorithm, the framework of HFA is shown in Fig. 2.

---

#### Algorithm 2. Hybrid firefly algorithm

---

Objective function  $f(\bar{x})$ ,  $\bar{x} = (x_1, x_2, \dots, x_n)^T$

Set algorithm's parameters  $\kappa, \beta_0, \alpha$

Generation  $t = 1$

Initialize a population of  $N$  fireflies  $x_i (i = 1, 2, \dots, N)$  by good point set

**While** ( $t < \text{Maximum Generation}$ ) or (stop criterion) **do**

**For**  $i = 1 : N$  all  $N$  fireflies

**For**  $j = 1 : i$  all  $N$  fireflies

**If** ( $x_i$  is chosen according to feasibility-based rules when we

---

```

compare  $x_i$  and  $x_j$ ) then
  Obtain attractiveness which varies with distance  $r$  via
   $\exp[-\kappa r^2]$ 
  Move firefly  $i$  towards  $j$  in all  $n$  dimensions
  Evaluate new solutions
End if
End for
End for
Rank the fireflies and find the current best
Generate new solution by Rosenbrock's local search algorithm as
explained in section III for the current best
Update the generation number  $t = t + 1$ ;
End while

```

Fig. 2. Pseudo code of the proposed hybrid firefly algorithm.

#### IV. EXPERIMENTS AND COMPARISONS

In this section, the performance of the proposed HFA is tested by solving several constrained benchmark functions and engineering applications. There are six parameters in the proposed method, and they are: population size  $N=50$ ,  $\varepsilon = 0.0001$ ,  $\kappa = 1$ ,  $\beta_0 = 1$ , and  $\alpha = 0.2$ , we can set two fixed number of iterations 3000 and 1000 for constrained benchmark functions and engineering application problems, respectively.

##### A. Constrained benchmark functions

We apply the proposed HFA method to solve four test constrained optimization functions (denoted by g02, g03, g06, and g11) from [8]. The details expressions of four functions see [8]. For each test function, 30 independent runs are performed in MATLAB 7.0. To further verify the performance of HFA, the results of our algorithm are compared against five typical state-of-the-art algorithms from the literature, including the hybrid cuckoo search (HCS) algorithm [1], the stochastic ranking (SR) [8], the hybrid genetic algorithm with pattern search (HGA) [9], the biogeography-based optimization (BBO) algorithm [10], and the self-adaptive velocity particle swarm optimization (SVPSO) algorithm [11]. TABLE I summarizes the experimental results using the above parameters, showing the best, mean, worst objective function values and the standard deviations (st.dev) after 30 independent runs by differential approaches.

TABLE I. COMPARISON OF OUR ALGORITHM WITH OTHER FOUR APPROACHES FOR FOUR FUNCTIONS

| Problem/<br>Optimal | Method | Best      | Mean      | Worst     | St.dev   |
|---------------------|--------|-----------|-----------|-----------|----------|
| g02/<br>-0.803619   | HCS    | -0.803619 | -0.762088 | -0.640249 | 4.10E-02 |
|                     | SR     | -0.803515 | -0.781975 | -0.726288 | 2.0E-02  |
|                     | HGA    | -0.611330 | -0.556323 | -0.526660 | 2.50E-02 |
|                     | BBO    | -0.803557 | -0.802774 | -0.792576 | 2.72E-03 |
|                     | SVPSO  | -0.803443 | -0.740577 | -0.631598 | 4.20E-02 |
|                     | HFA    | -0.803582 | -0.797061 | -0.782472 | 7.70E-03 |
| g03/<br>-1.000500   | HCS    | -1.000000 | -1.000000 | -1.000000 | 1.00E-10 |
|                     | SR     | -1.000    | -1.000    | -1.000    | 1.90E-04 |
|                     | HGA    | -1.000000 | -1.000000 | -1.000000 | 0.00E+00 |
|                     | BBO    | -1.000    | -1.000    | -1.000    | 6.04E-16 |
|                     | SVPSO  | -1.0048   | -1.0034   | -0.9976   | 1.70E-02 |
|                     | HFA    | -1.000388 | -1.000207 | -1.000052 | 1.16E-04 |

|                   |       |           |           |           |          |
|-------------------|-------|-----------|-----------|-----------|----------|
| g06/<br>-6961.814 | HCS   | -6961.814 | -6961.814 | -6961.814 | 8.36E-06 |
|                   | SR    | -6961.814 | -6875.940 | -6350.262 | 1.60E+02 |
|                   | HGA   | -6961.814 | -6961.814 | -6961.809 | 1.27E-03 |
|                   | BBO   | -6961.814 | -6961.814 | -6961.814 | 4.55E-12 |
|                   | SVPSO | -6961.814 | -6961.814 | -6961.814 | 0.00E+00 |
|                   | HFA   | -6961.814 | -6961.814 | -6961.814 | 3.88E-07 |
| g11/<br>0.749900  | HCS   | 0.749999  | 0.750000  | 0.750000  | 3.70E-09 |
|                   | SR    | 0.750     | 0.750     | 0.750     | 8.00E-05 |
|                   | HGA   | 0.750000  | 0.750000  | 0.750000  | 0.00E+00 |
|                   | BBO   | 0.750     | 0.750     | 0.750     | 0.00E+00 |
|                   | SVPSO | 0.749     | 0.749     | 0.749     | 3.90E-07 |
|                   | HFA   | 0.749999  | 0.750000  | 0.750000  | 2.38E-06 |

As shown in TABLE I, for problem g02, compare with HCS, HFA provided better “Mean”, “Worst” and “St.dev” values. The better “Best” value is obtained by HCS. With respect to SR, HGA and SVPSO, HFA found better results. Compare to the BBO, HFA reached similar results. For problem g03, with respect to HCS, SR, HGA and BBO, HFA obtained better results. For problem g06, compare with HCS, BBO and SVPSO, HFA obtained similar results. With respect to SR and HGA, HFA found better results. For problem g11, in compare with HCS, SR, HGA and BBO, HFA provided similar experimental results. With respect to SVPSO, HFA obtained worse results.

##### B. Engineering application

Weld beam design problem (as shown in Fig. 3), which has been often used as benchmark problem, was proposed by Eskandar [12]. In this problem, the aim is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress ( $\tau$ ), bending stress ( $\sigma$ ) in the beam, buckling load on the bar ( $P_b$ ), end deflection of the beam ( $\delta$ ). There are four design variables:  $x_1(h)$ ,  $x_2(l)$ ,  $x_3(t)$  and  $x_4(b)$ .

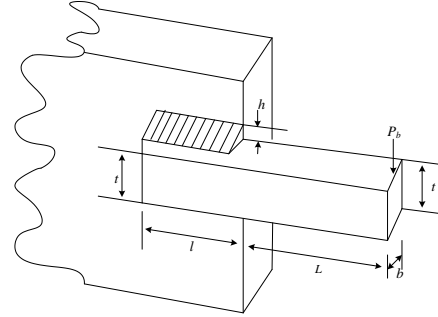


Fig. 3. Welded beam design problem.

The mathematical formulation of welded beam design problem is stated as:

$$\begin{aligned}
 \min \quad & f(\bar{x}) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \\
 \text{s.t.} \quad & g_1(\bar{x}) = \tau(\bar{x}) - \tau_{\max} \leq 0 \\
 & g_2(\bar{x}) = \sigma(\bar{x}) - \sigma_{\max} \leq 0 \\
 & g_3(\bar{x}) = x_1 - x_4 \leq 0 \\
 & g_4(\bar{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \leq 0 \\
 & g_5(\bar{x}) = 0.125 - x_1 \leq 0 \\
 & g_6(\bar{x}) = \delta(\bar{x}) - \delta_{\max} \leq 0 \\
 & g_7(\bar{x}) = P - P_c(\bar{x}) \leq 0
 \end{aligned} \tag{8}$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}; \tau' = \frac{P}{\sqrt{2x_1x_2}}; \tau'' = MR/J; M = P(L + x_2/2);$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}; J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}; \sigma(x) = \frac{6PL}{x_4x_3^2};$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3x_4}; P_c(x) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right).$$

$$P = 6000, L = 14, E = 30 \times 10^6, G = 12 \times 10^6, \tau_{\max} = 13600, \sigma_{\max} = 30000,$$

$$\delta_{\max} = 0.25. \text{ where } 0.1 \leq x_1, x_4 \leq 2, 0.1 \leq x_2, x_3 \leq 10.$$

This problem has been used as a benchmark problem for testing the efficiency of numerous optimization methods [12], such as GA2, GA3, CAEP, CPSO, WCA, and FA [3]. The obtained statistical results using the reported optimizers and the proposed HFA are given in TABLE II. The comparisons of the best solutions among several reported algorithms are given in TABLE III.

TABLE II. COMPARISON OF STATISTICAL RESULTS FOR VARIOUS METHODS FOR WELDED BEAM DESIGN PROBLEM

| Methods  | Best      | Mean      | Worst     | St.dev   |
|----------|-----------|-----------|-----------|----------|
| GA2[12]  | 1.748309  | 1.771973  | 1.785835  | 1.12E-02 |
| GA3[12]  | 1.728226  | 1.792654  | 1.993408  | 7.47E-02 |
| CAEP[12] | 1.724852  | 1.971809  | 3.179709  | 4.43E-01 |
| CPSO[12] | 1.728024  | 1.748831  | 1.782143  | 1.29E-02 |
| WCA[12]  | 1.724856  | 1.726427  | 1.744697  | 4.29E-03 |
| FA[3]    | 1.7312065 | 1.8786560 | 2.3455793 | 2.68E-01 |
| HFA      | 1.724893  | 1.724946  | 1.725086  | 5.46E-05 |

TABLE III. COMPARISON OF THE BEST SOLUTION GIVEN BY DIFFERENTIAL OPTIMIZERS FOR WELDED BEAM DESIGN PROBLEM

| Variables | CPSO[12]   | FA[3]      | GA3[12]    | HFA        |
|-----------|------------|------------|------------|------------|
| $x_1$     | 0.202369   | 0.2015     | 0.205986   | 0.20573556 |
| $x_2$     | 3.544214   | 3.5620     | 3.471328   | 3.47037213 |
| $x_3$     | 9.048210   | 9.0414     | 9.020224   | 9.03661649 |
| $x_4$     | 0.205723   | 0.2057     | 0.206480   | 0.20573598 |
| $g_1(x)$  | -13.655547 | -1.700E-04 | -0.103049  | -7.198E-04 |
| $g_2(x)$  | -78.814077 | -27.368227 | -0.231747  | -0.8760122 |
| $g_3(x)$  | -3.35E-03  | -0.0042000 | -5E-04     | -4.230E-07 |
| $g_4(x)$  | -3.424572  | -3.4243755 | -3.430044  | -3.4329470 |
| $g_5(x)$  | -0.077369  | -0.0765000 | -0.080986  | -0.0807356 |
| $g_6(x)$  | -0.235595  | -0.2355611 | -0.235514  | -0.2355407 |
| $g_7(x)$  | -4.472858  | 0.50949507 | -55.646888 | -0.3346258 |
| $f(x)$    | 1.728024   | 1.73121    | 1.728226   | 1.724893   |

In terms of statistical results, from TABLE II, compared with GA2, GA3, CPSO and FA, HFA obtained better “Best”, “Mean”, “Worst”, and “St.dev” results for welded beam design problem. With respect to CAEP and WCA, HFA provided better “Mean”, “Worst”, and “St.dev” results. The better “Best” values are obtained by CAEP and WCA. From

TABLE III, the proposed HFA method detected the best solution with considerable improvement compared with other optimizers.

## V. CONCLUSION

An effective hybrid firefly algorithm is proposed. Good point set method is adopted to obtain the sample points with better diversity of population. A Rosenbrock’s local search technique is introduced to improve the best approximation found by the firefly algorithm. Numerical and engineering application examples are used to validate the performance of the proposed algorithm. The results demonstrate that for most problems the global optima are effectively obtained and comparison with other optimizers proves that the proposed algorithm can efficiently identify the global or near-global design optimum.

## ACKNOWLEDGMENT

This work was financially supported by the National Natural Science Foundation of China (No. 61463009) and the Scientific Research Fund of Hunan Provincial Education Department (No. 14B097).

## REFERENCES

- [1] W. Long, X.M. Liang, Y.F. Huang and Y.X. Chen, “An effective hybrid cuckoo search algorithm for constrained global optimization,” *Neural Comput. Appl.* vol. 25, pp. 911-926, 2014.
- [2] X.S. Yang, “Firefly algorithms for multimodal optimization,” in *Proceedings of International Conference on Stochastic algorithms: Foundation and Applications*, pp. 169-178, 2009.
- [3] A.H. Gandomi, X.S. Yang and A.H. Alavi, “Mixed variable structural optimization using firefly algorithm,” *Comput. Struct.* Vol. 89, pp. 2325-2336, 2011.
- [4] R.L. Haupt and S.E. Haupt, “Practical genetic algorithms,” USA: John Wiley & Sons, 2004.
- [5] H. Liu, Z.X. Cai and Y. Wang, “A new constrained optimization evolutionary algorithm by using good point set,” in *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1247-1254, 2007.
- [6] R.M. Lewis, V. Torczona and M.W. Trossetc, “Direct search methods: then and now,” *J. Comput. Appl. Math.* Vol. 124, pp. 191-207, 2000.
- [7] K. Deb, “An efficient constraint handling method for genetic algorithm,” *Comput. Methods Appl. Mech. Eng.* vol. 186, pp. 311-338, 2000.
- [8] T.P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” *IEEE Trans. Evol. Comput.* Vol. 4, pp. 284-294, 2000.
- [9] L. Costal, I. Santo and E. Fernandes, “A hybrid genetic pattern search augmented Lagrangian method for constrained global optimization,” *Appl. Math. Comput.* Vol. 218, pp. 9415-9426, 2012.
- [10] I. Boussaid, A. Chatterjee, P. Siarry and M. Ahmed-Nacer, “Biogeography-based optimization for constrained optimization problems,” *Comput. Oper. Res.* Vol. 39, pp. 3293-3304, 2012.
- [11] H.Y. Lu and W.Q. Chen, “Self-adaptive velocity particle swarm optimization for solving constrained optimization problems,” *J. Glob. Optim.* Vol. 41, pp. 427-445, 2008.
- [12] H. Eskandar, A. Sadollah, A. Bahreininejad and M. Hamdi, “Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems,” *Comput. Struct.* Vol. 110-111, pp. 151-166, 2012.