Journal of Computational Science xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Journal of Computational Science

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

A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm

Xuewen Xia [a,b,*], Ling Gui [c], Guoliang He [d], Chengwang Xie [a,b], Bo Wei [a,b], Ying Xing [a,b], Ruifeng Wu [a,b], Yichao Tang [a,b]

[a] School of Software, East China Jiaotong University, Jiangxi 330013, China
[b] Intelligent Optimization and Information Processing Lab., East China Jiaotong University, Jiangxi 330013, China
[c] School of Economics and Management, East China Jiaotong University, Jiangxi 330013, China
[d] School of Computer, Wuhan University, Hubei 430079, China

ARTICLE INFO

Article history:
Received 31 March 2017
Received in revised form 16 May 2017
Accepted 11 July 2017
Available online xxx

Keywords:
Firefly algorithm
Particle swarm optimization
Knowledge-based detecting
Local search operator

ABSTRACT

As two widely used evolutionary algorithms, particle swarm optimization (PSO) and firefly algorithm (FA) have been successfully applied to diverse difficult applications. And extensive experiments verify their own merits and characteristics. To efficiently utilize different advantages of PSO and FA, three novel operators are proposed in a hybrid optimizer based on the two algorithms, named as FAPSO in this paper. Firstly, the population of FAPSO is divided into two sub-populations selecting FA and PSO as their basic algorithm to carry out the optimization process, respectively. To exchange the information of the two sub-populations and then efficiently utilize the merits of PSO and FA, the sub-populations share their own optimal solutions while they have stagnated more than a predefined threshold. Secondly, each dimension of the search space is divided into many small-sized sub-regions, based on which much historical knowledge is recorded to help the current best solution to carry out a detecting operator. The purposeful detecting operator enables the population to find a more promising sub-region, and then jumps out of a possible local optimum. Lastly, a classical local search strategy, i.e., BFGS Quasi-Newton method, is introduced to improve the exploitative capability of FAPSO. Extensive simulations upon different functions demonstrate that FAPSO is not only outperforms the two basic algorithm, i.e., FA and PSO, but also surpasses some state-of-the-art variants of FA and PSO, as well as two hybrid algorithms.

1. Introduction

In recent years, many real-world problems become extremely complex and are difficult solved by conventional algorithms. Thus, non-deterministic algorithms and heuristic algorithms play more and more important roles in various applications [10,29,30,44]. As a type of heuristic algorithm, evolutionary algorithms (EAs) have shown very favorable performance on non-convex and non-differentiable problems, and various EAs have been developed and applied to diverse difficult real-life problems during the last few decades.

Firefly algorithm (FA) [40] and particle swarm optimization (PSO) [22] are two widely used evolutionary algorithms inspired by some social behaviors of eusocial organisms. Through simple interaction among individuals, the entire population can manifest very high intelligence when optimizing a problem. Aming to further improve the performance of them and broaden their application fields, many strategies are proposed during the last decades, such as adjusting parameters [2,3,25,38,43,45] and enriching learning models [14,20,21,39]. However, considering that different optimizers have their own merits and characteristics which are suitable for different problems, many researchers pay much attention on hybridizing of different EAs to deal with real-world problems involving complexity, noise, imprecision, uncertainty, and vagueness [11,33,36].

In the research field of EAs, hybridization refers to merging different optimization techniques into a single framework. Through the synergistic mechanism, a hybrid algorithm could take advantage of various merits within different algorithms, and then yields more favorable performance than a single algorithm. Some

* Corresponding author at: School of Software, East China Jiaotong University, Jiangxi 330013, China.
E-mail address: xwxia@whu.edu.cn (X. Xia).

http://dx.doi.org/10.1016/j.jocs.2017.07.009
1877-7503/

preliminary research manifests that hybrid optimizers are effective and competent for global optimization [6,7,13].

Inspired by these researches, we proposed a hybrid evolutionary algorithm based on FA and PSO. In the hybrid optimizer, which is called FAPSO in this paper, there are three modules proposed to enhance its comprehensive performance. The first module is parallel-evolving module, in which an entire population is divided into two sub-populations parallel evolved by FA and PSO, respectively. To take advantage of different merits of PSO and FA, the two sub-populations share their own optimal solutions while they have ceased improve more than a predefined threshold. The second one is detecting module in which a purposeful detecting operator is adopted to help the best individual of the population to jump out of local optimum solutions. The last module is local search module, in which the BFGS Quasi-Newton method is applied to improve solutions' accuracy.

The rest of this paper is organized as follows. In Section 2, a brief introduction on FA and PSO is provided. The details of FAPSO are demonstrated in Section 3. Experimental setups, including details of benchmark functions and peer algorithms, are introduced in Section 4. Section 5 experimentally compares FAPSO with other 12 peer algorithms using 30 benchmark functions. Moreover, the efficiency and effectiveness of the modules involved in FAPSO are also discussed in this section. Finally, Section 6 concludes this paper.

2. A brief introduction on FA and PSO

2.1. Firefly algorithm (FA)

Firefly algorithm (FA) inspired by the social behavior of fireflies flying in the tropical and temperate summer sky was proposed by Yang in 2009 [40]. In FA, a firefly's brightness I depends on its position $\mathbf{X}$ which is regarded as a potential solution. And the trajectory of the swarm can be characterized as a search process. During the optimization process of FA, a firefly moves towards a brighter one not only depending on I of the brighter firefly but also relying on the distance r between the two fireflies.

In the canonical FA, I of a firefly is determined by $\mathbf{X}$ which is proportional to the value of objective function $I(\mathbf{X}) \propto f(\mathbf{X})$. In addition, inspired by the phenomenon that the brightness is always absorbed in the light propagation media, I in FA decreases with the distance r from its source. A widely accepted update form of I is defined as (1).

$$I(r) = I_0 \cdot e^{-\gamma r^2} \tag{1}$$

where I_0 denotes the light intensity of the light source, and γ is the light absorption coefficient of the propagation media.

Accordingly, a firefly's attractiveness β, which is proportional to I, can be described as (2).

$$\beta(r) = (\beta_0 - \beta_{\min}) \cdot e^{-\gamma r^2} + \beta_{\min} \tag{2}$$

where β_0 is the attractiveness at $r = 0$, generally takes $\beta_0 = 1$; β_{min} is the minimum attractiveness.

The distance between any pair of fireflies, whose positions are denoted as $\mathbf{X}_i$ and $\mathbf{X}_j$, respectively, can be represented by the Euclidean distance as (3).

$$r_{ij} = ||\mathbf{X}_i - \mathbf{X}_j|| = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{3}$$

where x_{ik} and x_{jk} are the kth component of the spatial coordinate $\mathbf{X}_i$ and $\mathbf{X}_j$, respectively.

Based on the definition introduced above, the movement of firefly $\mathbf{X}_i$ attracted by anther brighter firefly $\mathbf{X}_j$ can be described as (4).

$$\mathbf{X}_i = \mathbf{X}_i + ((\beta_0 - \beta_{\min}) \cdot e^{-\gamma r_{ij}^2} + \beta_{\min}) \cdot (\mathbf{X}_j - \mathbf{X}_i) + \alpha \cdot (rnd - 0.5) \tag{4}$$

where α is the parameter deciding the size of the random walk, and rnd is a random number uniformly distributed in [0, 1].

The pseudo code of FA is detailed in Algorithm 1.

Algorithm 1. FA

```
Begin
Generate initial population of fireflies X_i (i = 1, . . ., N);
Initialize parameters: α, γ, β_min, t = 0, and fes = 0;
Brightness I_i at X_i is determined by f(X_i);
Define light absorption coefficient γ;
While (not meet the stop conditions)
    For i=1: N all N fireflies
        For j=1: N all N fireflies
            If I_j > I_i Then
                Move firefly i towards j in all dimension according to Eq. (4);
            End If
            Attractiveness varies with distance according to Eq. (2);
            Evaluate the new solution and update its brightness; fes = fes + 1;
        End For
    End For
Rank the fireflies and find the current best;
t = t + 1;
End While
Post process results.
End
```

2.2. Particle swarm optimization (PSO)

Particle swarm optimization algorithm (PSO) is a widely known swarm intelligence algorithm proposed by Kennedy and Eberhart in 1995 [22,31]. During the optimizing process for a specific problem with D dimension variables, the ith particle has a velocity vector and a position vector represented as $\mathbf{V}_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ and $\mathbf{X}_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, respectively. The vector $\mathbf{X}_i$ is regarded as a candidate solution of the problem while the vector $\mathbf{V}_i$ is treated as the particle's search direction and step size. During the process of optimization, each particle decides its trajectory according to its personal historical best position $\mathbf{Pb}_i = [pb_{i1}, pb_{i2}, \ldots, pb_{iD}]$ and the global best-so-far position $\mathbf{Gb} = [gb_1, gb_2, \ldots, gb_D]$. In the canonical PSO, the update rules of $\mathbf{V}_i$ and $\mathbf{X}_i$ are defined as (5) and (6), respectively.

$$v_{ij}^{t+1} = \omega \cdot v_{ij}^t + c_1 \cdot rnd_1 \cdot (pb_{ij}^t - x_{ij}^t) + c_2 \cdot rnd_2 \cdot (gb_j^t - x_{ij}^t) \tag{5}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \tag{6}$$

where ω represents an inertia weight indicating how much the previous velocity is preserved; c_1 and c_2 are known as two acceleration coefficients determining relative learning weights for $\mathbf{Pb}_i$ and $\mathbf{Gb}$, which called "self-cognitive" and "social-learning", respectively; rnd_1 and rnd_2 are two random numbers uniformly distributed over [0, 1].

The pseudo-code of PSO is detailed as Algorithm 2.

Algorithm 2. PSO