Applied Mathematics and Computation 252 (2015) 155–165

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A review of chaos-based firefly algorithms: Perspectives and research challenges

Iztok Fister Jr. [a], Matjaž Perc [b,c,d,*], Salahuddin M. Kamal [c], Iztok Fister [a]

[a] Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, SI-2000 Maribor, Slovenia
[b] Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, SI-2000 Maribor, Slovenia
[c] Department of Physics, Faculty of Science, King Abdulaziz University, Jeddah, Saudi Arabia
[d] CAMTP – Center for Applied Mathematics and Theoretical Physics, University of Maribor, Krekova 2, SI-2000 Maribor, Slovenia

ARTICLE INFO

Keywords:
Firefly algorithm
Chaos
Chaotic map
Optimization
Swarm intelligence

ABSTRACT

The firefly algorithm is a member of the swarm intelligence family of algorithms, which have recently showed impressive performances in solving optimization problems. The firefly algorithm, in particular, is applied for solving continuous and discrete optimization problems. In order to tackle different optimization problems efficiently and fast, many variants of the firefly algorithm have recently been developed. Very promising firefly versions use also chaotic maps in order to improve the randomness when generating new solutions and thereby increasing the diversity of the population. The aim of this review is to present a concise but comprehensive overview of firefly algorithms that are enhanced with chaotic maps, to describe in detail the advantages and pitfalls of the many different chaotic maps, as well as to outline promising avenues and open problems for future research.

1. Introduction

In the past, facing the real-world optimization problems was in the domain of mathematicians and engineers. They developed many mathematical methods for solving the optimization problems. The first algorithms solved the problems exactly by enumerating all the possible solutions, but rapidly algorithms for approximate (heuristic) solving of these problem have been emerged because of a huge time complexities of the exact methods. One of the more promising algorithms today are swarm intelligence (SI) based algorithms inspired with a collective behavior of some simple unintelligent insects or animals, who work together in order to be capable of solving the complex problems. For instance, a foraging of insects connects social living bees, ants and termites. In nature, one individual cannot survive, but when living together in colonies, individuals are stronger and thus capable of performing very complex tasks (e.g., huge mounds by termites). These colonies of insects act as decentralized, and self-organized systems that prevents a single insect to act alone.

A firefly algorithm (FA) is one of the younger member of SI-based algorithms that was introduced by Yang in [40] at 2008. Since its introduction, many researchers began working with FA. At the beginning, some modified variants were proposed that were applied for solving the continuous optimization [43], multimodal [41], constrained optimization [30], and later also for real-world problems [47,36,16,25]. Competition of FA with other well-known meta-heuristics led to the development of more robust and sophisticated FA variants. For example, Fister et al. in [12] proposed a memetic self-adaptive FA

* Corresponding author at: Faculty of Natural Sciences and Mathematics, University of Maribor, Koroška cesta 160, SI-2000 Maribor, Slovenia.
E-mail address: matjaz.perc@um.si (M. Perc).

http://dx.doi.org/10.1016/j.amc.2014.12.006
0096-3003/

(MFA), where values of control parameters are changed during the run. Their version of the FA was developed for combinatorial optimization, while Galvez and Iglesias [17] created a new memetic self-adaptive FA for the continuous optimization. Simultaneously, also a bunch of applications using FA were created [18,19,3].

Stochastic optimization algorithms search for optimal solutions by involving randomness in some constructive way [24]. In contrast, if optimization methods provide the same results when doing the same things, these methods are said to be deterministic [11]. If the deterministic system behaves unpredictably, it arrives at a phenomenon of chaos [11]. Consequently, randomness in SI algorithms plays a huge role because this phenomenon affects the exploration and exploitation in search process [5]. These companions of stochastic global search represent the two cornerstones of problem solving, i.e., exploration refers to moves for discovering entirely new regions of a search space, while exploitation refers to moves that focus searching the vicinity of promising, known solutions to be found during the search process. Both components are also referred to as intensification and diversification in another terminology [22]. However, these refer to medium- to long-term strategies based on the usage of memory, while exploration and exploitation refer to short-term strategies tied with randomness [5].

Essentially, randomness is used in SI algorithms in order to explore new points by moving the particles towards the search space. In line with this, several random number distributions can be helpful. For example, uniform distribution generates each point of the search space using the same probability. On the other hand, Gaussian distribution is biased towards the observed solution, this means that the smaller modifications occur more often than larger ones [9]. On the other hand, the appropriate distribution depends on the problem to be solved, more precisely, on a fitness landscape that maps each position in the search space into fitness value.

Interestingly, some authors enhanced the original FA with a chaos in order to improve it. However, a comprehensive study about chaos-based FA (CFA) are still missing. In this paper, therefore, we would like to assemble papers tackling the CFA, introduce a taxonomy of these algorithms, review the most frequently used chaotic maps in CFAs and critical discuss its advantages and disadvantages. Finally, we outline the possible directions for the future research. The purpose of this review is twofold: Firstly, to show that enhancing the CFA with chaotic maps may improve the original FA significantly, and secondly, to encourage the potential developers to start using the CFA for solving the hard problems in practice.

The structure of the paper is as follows. Section 2 describes the FA basics. In Section 3, the possible ways how to enhance the original FA with chaotic maps are explained. A brief review of the chaos-based FAs is presented in Section 4. The paper concludes with Section 5 where also the possible directions for further development are outlined.

2. The firefly algorithm

After watching the environment and sky during the summer nights, professor Yang [41] got an idea for developing a new algorithm. His source of inspiration were small lighting bugs called fireflies [6]. A phenomenon presents their flashing lights which can be seen and adored on clear summer nights. The production of these lights is done by a complicated set of chemical reactions. They flashes in order to attract a mating partner or for protection against predators. The intensity of their lights I decrease when the distance r from the light source increases in terms of $I \propto r^2$. On the other hand, air absorbs the light as the distance from the source increases [16]. This firefly behavior is modeled into the Yang's firefly algorithm (FA) so that the light intensity is proportional to the fitness function of the problem to be solved.

However, because an adaptation of the natural behavior of the fireflies in an algorithm is too complex, the following idealized rules are considered by developing of the FA:

- all fireflies are unisex,
- their attractiveness is proportional to their brightness, and
- the brightness of a firefly is affected or determined by the landscape of the objective function.

A pseudo-code of the FA is illustrated in Algorithm 1, from which it can be seen that the algorithm consists of the following elements:

- a representation of a firefly,
- an initialization (line 1 in Algorithm 1),
- a moving operator (line 8 in Algorithm 1),
- an objective function (lines 2, 11 in Algorithm 1).

Note that this SI-based algorithm does not support an explicit selection. This means, a moving of the whole population of solution depends on the position of the global best solution as well as positions of the local better (brighter) solutions in the neighborhood (line 13 in Algorithm 1). Additionally, a termination condition (line 4 in Algorithm 1) is needed in order to provide the proper termination of the FA.