Survey Paper

A comprehensive review of firefly algorithms

Iztok Fister [a,*], Iztok Fister Jr.[a], Xin-She Yang [b], Janez Brest [a]

[a] University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia
[b] School of Science and Technology, Middlesex University, London NW4 4BT, UK

ABSTRACT

The firefly algorithm has become an increasingly important tool of Swarm Intelligence that has been applied in almost all areas of optimization, as well as engineering practice. Many problems from various areas have been successfully solved using the firefly algorithm and its variants. In order to use the algorithm to solve diverse problems, the original firefly algorithm needs to be modified or hybridized. This paper carries out a comprehensive review of this living and evolving discipline of Swarm Intelligence, in order to show that the firefly algorithm could be applied to every problem arising in practice. On the other hand, it encourages new researchers and algorithm developers to use this simple and yet very efficient algorithm for problem solving. It often guarantees that the obtained results will meet the expectations.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Swarm Intelligence (SI) belongs to an artificial intelligence discipline (AI) that became increasingly popular over the last decade [1]. It is inspired from the collective behavior of social swarms of ants, termites, bees, and worms, flock of birds, and schools of fish. Although these swarms consist of relatively unsophisticated individuals, they exhibit coordinated behavior that directs the swarms to their desired goals. This usually results in the self-organizing behavior of the whole system, and collective intelligence or swarm intelligence is in essence the self-organization of such multi-agent systems, based on simple interaction rules. This coordinated behavior is performed due to interaction between individuals, for example, termites and worms are able to build sophisticated nests, while ants and bees also use this collective behavior when searching for food. Typically, ants interact with each other via chemical pheromone trails in order to find the shortest path between their nest and the food sources. In a bee colony, the role of informer is played by so-called scouts, i.e., individual bees that are responsible for searching for new promising areas of food sources. Here, the communication among bees is realized by a so-called 'waggle dance', through which the bee colony is directed by scouts. During this discovery of the new food sources, a trade-off between exploration (the collection of new information) and exploitation (the use of existing information) must be performed by the bee colony [2]. That is, the bee colony must be aware when to exploit existing food sources and when to

look for new food sources so as to maximize the overall nectar intake while minimizing the overall foraging efforts.

The swarm of individuals shows collective behavior; for example, where to forage, when to reproduce, where to live, and how to divide the necessary tasks among the available work force [2]. In fact, these decisions are made in a decentralized manner by individuals based on local information obtained from interactions with their intermediate environments.

Swarm intelligence refers to a research field that is concerned with a collective behavior within self-organized and decentralized systems. This term was probably first used by Beni [3] in the sense of cellular robotic systems consisting of simple agents that organize themselves through neighborhood interactions. Recently, methods of swarm intelligence are used in optimization, the control of robots, and routing and load balancing in new-generation mobile telecommunication networks, demanding robustness and flexibility. Examples of notable swarm-intelligence optimization methods are ant colony optimization (ACO) [4,5], particle swarm optimization (PSO) [6], and artificial bee colony (ABC) [7,8]. Today, some of the more promising swarm-intelligence optimization techniques include the firefly algorithm (FA) [9–12], cuckoo-search [13], and the bat algorithm [14], while new algorithms such as the krill herd bio-inspired optimization algorithm [15] and algorithms for clustering [16,17] also emerged recently.

FA is one of the recent swarm intelligence methods developed by Yang [9] in 2008 and is a kind of stochastic, nature-inspired, meta-heuristic algorithm that can be applied for solving the hardest optimization problems (also NP-hard problems). This algorithm belongs to stochastic algorithms. This means that it uses a kind of randomization by searching for a set of solutions. It is inspired by the flashing lights of fireflies in nature. Heuristic means 'to find' or 'to discover solutions by trial and error' [9].

In fact, there is no guarantee that the optimal solution will be found in a reasonable amount of time. Finally, meta-heuristic means 'higher level', where the search process used in algorithms is influenced by certain trade-off between randomization and local search [9]. In the firefly algorithm, the 'lower level' (heuristic) concentrates on the generation of new solutions within a search space and thus, selects the best solution for survival. On the other hand, randomization enables the search process to avoid the solution being trapped into local optima. The local search improves a candidate solution until improvements are detected, i.e., places the solution in local optimum.

Each meta-heuristic search process depends on balancing between two major components: exploration and exploitation [18]. Both terms were defined implicitly and are affected by the algorithm's control parameters. In the sense of the natural bee colony, the actions of exploration and exploitation have yet to be explained. For the meta-heuristic algorithms [19], the exploration denotes the process of discovering the diverse solutions within the search space, while exploitation means focusing the search process within the vicinities of the best solutions, thus, exploiting the information discovered so far.

Note that FA is population-based. The population-based algorithms have the following advantages when compared with single-point search algorithms [20]:

- Building blocks are put together from different solutions through crossover.
- Focusing a search again relies on the crossover, and means that if both parents share the same value of a variable, then the offspring will also have the same value of this variable.
- Low-pass filtering ignores distractions within the landscape.
- Hedging against bad luck in the initial positions or decisions it makes.
- Parameter tuning is the algorithm's opportunity to learn good parameter values in order to balance exploration against exploitation.

The rest of this section will discuss briefly the characteristics of fireflies that have served as an inspiration for developing the firefly algorithm.

The main characteristic of fireflies is their flashing light. These lights have two fundamental functions: to attract mating partners and to warn potential predators. However, the flashing lights obey more physical rules. On the one hand, the light intensity I decreases as the distance r increases according to the term $I \propto 1/r^2$. This phenomenon inspired Yang [9] to develop the firefly algorithm. On the other hand, the firefly acts as an oscillator that charges and discharges (fires) the light at regular intervals, i.e., at $\theta = 2\pi$. When the firefly is placed within the vicinity of another firefly, a mutual coupling occurs between both fireflies. This behavior of fireflies especially inspired the solution of graph coloring problems. On this basis, a distributed graph coloring algorithm was developed by Lee [21]. Recently, the similar and greater researched behavior of Japanese tree frogs inspired Hernández and Blum [22] into developing a more useful distributed graph coloring algorithm. Therefore, the further development of the algorithm based on the oscillatory behavior of fireflies has diminished. Therefore, in this paper we are focused on Yang's firefly algorithm.

The aim of this paper is twofold: to present areas where FA has been successfully applied, and thus to broaden the range of its potential users. The structure of this paper is as follows: Section 2 discusses the biological foundations of the firefly algorithm. Main characteristics of this algorithm are then exposed, and finally, the algorithmic structure is presented. Section 3 provides an extensive review of application areas to which this algorithm has already

been applied. Let us mention only the most important areas of its application: continuous, combinatorial, constraint and multi-objective optimization, and optimization in dynamic and noisy environments. Beside optimization, it is applicable for solving classification problems arose in areas like machine learning, data mining, and neural network. Additionally, many applications cover an area of engineering applications and solve real-world problems. Section 4 brings the discussion of FA behavior and directions for further development of these algorithms are covered. This paper concludes with an overview of the work that has been performed within the discipline of swarm intelligence.

2. Firefly algorithm

2.1. Biological foundations

Fireflies (`Coleoptera: Lampyridae`) are among the most charismatic of all insects, and their spectacular courtship displays have inspired poets and scientists alike [23]. Nowadays, more that 2000 species exist worldwide. Usually, fireflies live in a variety of warm environments and they are most active in summer nights. A lot of researchers have studied firefly phenomena in nature and there exist numerous papers researching fireflies, for example, [24–28].

Fireflies are characterized by their flashing light produced by biochemical process bioluminescence. Such flashing light may serve as the primary courtship signals for mating. Besides attracting mating partners, the flashing light may also serve to warn off potential predators. Note that in some firefly species some adults are incapable of bioluminescence. These species attract their mates due to pheromone, similarly to ants.

In fireflies, bioluminescent reactions take place from light-producing organs called lanterns. The most bioluminescent organisms provide only slowly modulated flashes (also glows). In contrast, adults in many firefly species are able to control their bioluminescence in order to emit high and discrete flashes. The lanterns' light-production is initialized by signals originating within the central nervous system of firefly.

Most firefly species rely on bioluminescent courtship signals. Typically, the first signalers are flying males, who try to attract flightless females on the ground. In response to these signals, the females emit continuous or flashing lights. Both mating partners produce distinct flash signal patterns that are precisely timed in order to encode information like species identity and sex. Females are attracted according to behavioral differences in the courtship signal. Typically, females prefer brighter male flashes. It is well known that the flash intensity varies with the distance from the source. Fortunately, in some firefly species females cannot discriminate between more distant flashes produced by stronger light source and closer flashes produced by weaker light sources.

Firefly flash signals are highly conspicuous and may therefore deter a wide variety of potential predators. In the sense of natural selection [29], where only the strongest individual can survive, flash signals evolve as defense mechanisms that serve to warn potential predators.

Two features are characteristics for swarm intelligence: self-organization and decentralized decision making. Here, autonomous individuals live together in a common place as, for example, bees in hives, ants in anthills, etc. In order to live in harmony, some interaction or communication is needed among group members who live together (sociality). In fact, individuals within a group cannot behave as if they are solitary, but must adapt to the overall goals within the groups. The social life of fireflies is not just dedicated to foraging, but more to reproduction. These collective decisions are

closely connected with the flashing light behavior that served as the main biological foundation for developing the firefly algorithm.

2.2. Structure of the firefly algorithm

As mentioned in Section 1, this paper focuses on Yang's [9] implementation of the firefly algorithm. This algorithm is based on a physical formula of light intensity I that decreases with the increase in the square of the distance r^2. However, as the distance from the light source increases, the light absorption causes that light becomes weaker and weaker. These phenomena can be associated with the objective function to be optimized. As a result, the base FA can be formulated as illustrated in Algorithm 1.

Some flashing characteristics of the fireflies are idealized in order to formulate the FA, as follows:

Algorithm 1. Pseudo-code of the base Firefly algorithm.

```
 1:  t = 0; s* = ∅; γ = 1.0;      // initialize: gen.counter, best
     solution, attractiveness
 2:  P^(0) = InitializeFA();      // initialize a population
 3:  while (t < MAX_FES) do
 4:      α^(t) = AlphaNew();       // determine a new value of α
 5:      EvaluateFA(P^(t), f(s));     // evaluate s according to f(s)
 6:      OrderFA(P^(t), f(s));       // sort s according to f(s)
 7:      s* = FindTheBestFA(P^(t), f(s));   // determine the best
     solution
 8:      P^(t+1) = MoveFA(P^(t));      // vary the attractiveness
     accordingly
 9:      t = t + 1;
10:  end while
```

- All fireflies are unisex.
- Their attractiveness is proportional to their light intensity.
- The light intensity of a firefly is affected or determined by the landscape of the fitness function.

The population of fireflies is initialized by the 'InitializeFA' function. Typically, this initialization is performed randomly. The firefly search process comprises the inside of the **while** loop (lines 3–10 in Algorithm 1) and is composed of the following steps: Firstly, the 'AlphaNew' function is dedicated to modify the initial value of parameter α. Note that this step is optional in the firefly algorithm. Secondly, the 'EvaluateFA' function evaluates the quality of the solution. The implementation of a fitness function $f(s)$ is performed inside this. Thirdly, the 'OrderFA' function sorts the population of fireflies according to their fitness values. Fourthly, the 'FindTheBestFA' function selects the best individual in population. Finally, the 'MoveFA' function performs a move of the firefly positions in the search space. Note that the fireflies are moved towards the more attractive individuals.

The firefly search process is controlled by the maximum number of fitness function evaluations MAX_FES.

2.3. Characteristics of the firefly algorithm

In order to design FA properly, two important issues need to be defined: the variation of light intensity and the formulation of attractiveness. These two issues enable developers to tailor different firefly algorithms in such a manner that they are best suited to the demands of the problems to be solved. In the standard firefly algorithm, the light intensity I of a firefly representing the solution s is proportional to the value of fitness function $I(s) \propto f(s)$, while the light intensity $I(r)$ varies according to the following equation:

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

where I_0 denotes the light intensity of the source, and the light absorption is approximated using the fixed light absorption coefficient γ. The singularity at $r=0$ in the expression I/r^2 is avoided by combining the effects of the inverse square law and an approximation of absorption in Gaussian form. The attractiveness β of fireflies is proportional to their light intensities $I(r)$. Therefore, a similar equation to Eq. (1) can be defined, in order to describe the attractiveness β

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{2}$$

where β_0 is the attractiveness at $r=0$. The light intensity I and attractiveness β are in some way synonymous. While the intensity is referred to as an absolute measure of emitted light by the firefly, the attractiveness is a relative measure of the light that should be seen in the eyes of the beholders and judged by other fireflies [9].

The distance between any two fireflies s_i and s_j is expressed as the Euclidean distance by the base firefly algorithm, as follows:

$$r_{ij} = \|s_i - s_j\| = \sqrt{\sum_{k=1}^{k=n} (s_{ik} - s_{jk})^2}, \tag{3}$$

where n denotes the dimensionality of the problem. The movement of the i-th firefly is attracted to another more attractive firefly j. In this manner, the following equation is applied:

$$s_i = s_i + \beta_0 e^{-\gamma r_{ij}^2}(s_j - s_i) + \alpha \epsilon_i, \tag{4}$$

where ϵ_i is a random number drawn from Gaussian distribution. The movements of fireflies consist of three terms: the current position of i-th firefly, attraction to another more attractive firefly, and a random walk that consists of a randomization parameter α and the random generated number from interval $[0, 1]$. When $\beta_0 = 0$ the movement depends on the random walk only. On the other hand, the parameter γ has a crucial impact on the convergence speed. Although the value of this parameter can theoretically capture any value from interval $\gamma \in [0, \infty)$, its setting depends on the problem to be optimized. Typically, it varies from 0.1 to 10.

In summary, FA is controlled by three parameters: the randomization parameter α, the attractiveness β, and the absorption coefficient γ. According to the parameter setting, FA distinguishes two asymptotic behaviors. The former appears when $\gamma \to 0$ and the latter when $\gamma \to \infty$. If $\gamma \to 0$, the attractiveness becomes $\beta = \beta_0$. That is, the attractiveness is constant anywhere within the search space. This behavior is a special case of particle swarm optimization (PSO). If $\gamma \to \infty$, the second term falls out from Eq. (4), and the firefly movement becomes a random walk, which is essentially a parallel version of simulated annealing. In fact, each implementation of FA can be between these two asymptotic behaviors.

3. Studies on firefly algorithms: classifications and analysis

Several variants of firefly algorithms exist in the literature. A certain classification scheme is necessary in order to classify them. The easiest way to achieve this purpose is to distinguish the firefly algorithms according to the settings of their algorithm parameters (also strategy parameters) [30]. The settings of these parameters are crucial for good performance and, therefore, carefully selected by developers. In general, there are two ways to seek algorithm parameters properly. On the one hand, by parameter tuning [31] the good values of parameters are found before the algorithm's run, and are fixed during iterations. On the other hand, by parameter control [30] the values of the parameters are modified during the runs.

Furthermore, the behavior of FA does not depend only on the proper values of parameters, but also on what components or features are incorporated into it. Therefore, the classification

scheme should be able to classify firefly algorithms according to these aspects being taken into account. In this review, we were interested in the following aspects:

- What is modified?
- How are the modifications made?
- What is the scope of the modifications?

From the first aspect, firefly algorithms can be classified according to the components or features of which they consists. These are

- representation of fireflies (binary, real-valued),
- population scheme (swarm, multi-swarm),
- evaluation of the fitness function,
- determination of the best solution (non-elitism, elitism),
- moving fireflies (uniform, Gaussian, Lévy flights, chaos distribution).

With regard to the second aspect, the categories of parameter control in firefly algorithms can be divided into: deterministic, adaptive, and self-adaptive. Finally, according to the last aspect, modifications in firefly algorithms may affect: an element of the firefly, the entire firefly, or the whole population.

In the early stages, FA acts as a global problem solver. That is, for several continuous optimization problems, the algorithm found the desired solutions. Difficulty arose, when the appropriate solutions could not be found for some other optimization problems. This is in accordance with the No-Free-Lunch theorem [32]. To circumvent this theorem, hybridization has been applied to optimization algorithms for solving a given set of problems. In line with this, firefly algorithms have been hybridized with other optimization algorithms, machine learning techniques, heuristics, etc. Hybridization can take place in almost every component of the firefly algorithm, for example, initialization procedure, evaluation function, moving function, etc.

In this paper, the firefly algorithms are analyzed according to Fig. 1, where the classical firefly algorithms are divided into modified and hybrid. Note that the classical firefly algorithms have been used mainly on continuous optimization problems. In order to provide the optimal results by solving the various classes of problems, they have been subject to several modifications and hybridizations. The main directions of these modifications have gone into the development of elitist and binary firefly algorithms, Gaussian, Lévy flights and chaos based firefly algorithms, and the parallelized firefly algorithms. On the other hand, the following

hybridizations have been applied to the classical firefly algorithm: Eagle strategy, genetic algorithm, differential evolution, local search, neural network, learning automata and ant colony.

In the rest of this paper, we first present a review of papers describing the classical firefly algorithm. Then, we review the studies that address the modified and hybridized versions of the firefly algorithm. Finally, an overview of papers is carried out that deal with optimization and engineering applications.

3.1. Classical firefly algorithms

Firefly algorithm inspired by the variations of light intensity was developed by Yang [9] in 2008. In the publication that introduced the classical FA, other nature inspired meta-heuristics were also described. There, it was formulated and the implementation in Matlab was discussed in detail. In order to demonstrate its performance, the four peaks two-dimensional function was used. The results of those experiments exposed the multi-modal characteristics of this algorithm. That is, the classical FA is able to discover more optimal solutions in the search space simultaneously.

In his paper [33], Yang established that the new FA was especially suitable for multi-modal optimization applications [34]. This claim was derived from the fact that fireflies can automatically subdivide themselves into a few subgroups because neighboring attraction is stronger than long-distance attraction. Experiments proving this claim were performed for finding the global optima of various multi-modal test functions taken from the literature, and compared with genetic algorithm and particle swarm optimization. FA was superior to both the mentioned algorithms in terms of efficiency and success rate. Therefore, the same author speculated that FA is potentially more powerful in solving other NP-hard problems as well.

The same author in [35] experimented with the classical FA solving the non-linear pressure vessel design optimization problem. This problem belongs to the class of continuous optimization problems. At first, he validated the algorithm using certain standard test functions. The results of FA for pressure vessel design optimization implied that this algorithm is potentially more powerful than other existing algorithms such as particle swarm optimization.

Yang's paper [36] provides an overview of the nature inspired meta-heuristic algorithms, including ant colony optimization, cuckoo search, differential evolution, firefly algorithm, harmony search, genetic algorithm, simulated annealing, and particle swarm optimization. On the base of the common characteristics

Classical firefly algorithm

Modified:
- Elitist
- Binary
- Gaussian
- Lévi fligths
- Chaos
- Parallel

Hybrid:
- Eagle strategy
- Genetic algorithm
- Differential evolution
- Local search
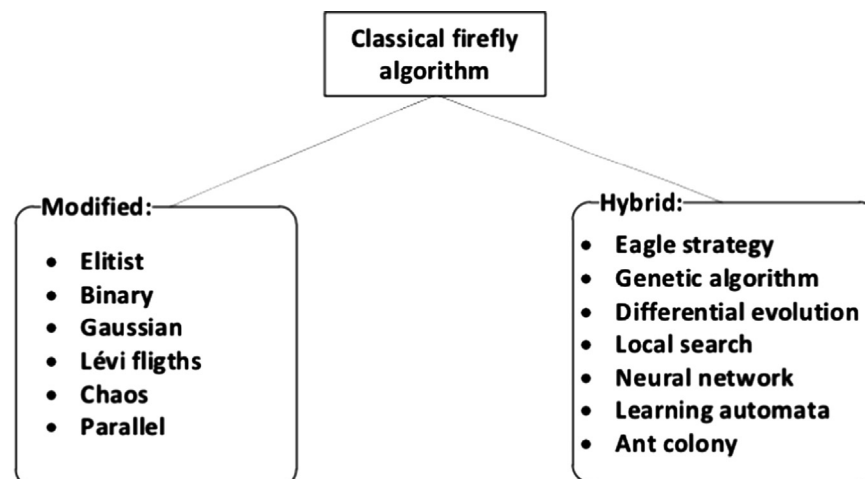- Neural network
- Learning automata
- Ant colony

Fig. 1. Taxonomy of firefly algorithms.

of these, Yang proposed a new, generic, meta-heuristic algorithm for optimization called the Generalized Evolutionary Walk Algorithm (GEWA) with the following three major components: (1) global exploration by randomization, (2) intensive local search by random walk, and (3) the selection of the best using some elitism. Interestingly, this algorithm tries to explicitly balance exploration and exploitation via a new randomization control parameter. However, the value of this parameter is problem-dependent.

Parpinelli, in his survey 'New inspirations in swarm intelligence' [37], explained the foundations that inspire developers of the new nature-inspired swarm intelligence algorithms. These foundations are bacterial foraging (BFO) [38], fireflies' bioluminescence [9], slime moulds life-cycle [39], cockroach infestation [40], mosquitoes host-seeking [41], bats echolocation [42], and various bees algorithms (BAs), i.e., inspired by bees foraging [43], and bees mating [44]. The more important applications and the main features of such meta-heuristics were also reported.

Zang et al. [45] systematically reviewed and analyzed the following nature-inspired algorithms: ant colony algorithm, bees algorithm, genetic algorithm, and the firefly algorithm. Although the paper mainly focused on the original principles behind these algorithms, their applications were also discussed.

The complete list of classical firefly algorithms is summarized in Table 1.

3.2. Modified firefly algorithms

The firefly algorithms depend primarily on the variation of light intensity and the formulation of attractiveness. Both factors allow significant scope for algorithm improvements (Table 2). For example, Luleseged et al. [46] modified the random movement of the brightest firefly that in some generations, when the current best position does not improve, may decreases its brightness. The proposed modification tries to improve the brightest firefly position generating the m-uniform random vectors and moves it in the direction of the best performance. If such a direction does not exist, the brightest firefly stays in its current position. In this case, the brightest firefly is also an elitist solution, because it is never replaced by a best-found solution in the current generation that has lower fitness. Experiments that were done optimizing seven benchmark functions showed that this modified FA outperform its classical predecessor.

Table 1
Classical firefly algorithms.

Topic	References
Introducing the firefly algorithm	[9]
Multi-modal test functions	[33]
Continuous and combinatorial optimization	[35]
Review of nature inspired meta-heuristics	[36,37,45]

Table 2
Modified firefly algorithms.

Topic	References
Elitist firefly algorithm	[46]
Binary represented firefly algorithm	[47–49,65]
Gaussian randomized firefly algorithm	[54,55]
Lévy flights randomized firefly algorithm	[55,58,59]
Chaos randomized firefly algorithm	[60,62]
Parallel firefly algorithm	[63,64]

Interestingly, a lot of binary firefly algorithms have emerged for solving different classes of problems, e.g., [47–49]. Palit in [47] proposed a binary FA for cryptanalysis in order to determine the plain text from the cipher text, using the Merkle–Hellman knapsack cipher algorithm [50]. Changes to almost all components of the binary FA need to be performed because of the new representation of fireflies. The results of this algorithm were compared with the results of the genetic algorithm. This comparison showed that the proposed binary FA was much more efficient than the genetic algorithm when used for the same purpose. Next, an implementation of the binary FA was developed by Falcon et al. [48] that uses a binary encoding of the candidate solutions, an adaptive light absorption coefficient for accelerating the search, and problem-specific knowledge to handle infeasible solutions. The empirical analysis was performed by solving the system-level fault diagnostic that is a combinatorial optimization problem. The results of the proposed algorithm when compared with an artificial immune system approach (AIS) [51] and particle swarm optimization (PSO) [52] showed that it outperforms existing techniques in terms of convergence speed and memory requirements. In the paper of Chandrasekaran et al. [49], the authors proposed a binary coded FA for solving network and reliability constrained unit commitment (UC) problem [53], without satisfying the network security constraints. The effectiveness of the proposed algorithm was demonstrated on 10 units of the IEEE-RTS system. The results of the proposed algorithm were promising when compared with the other techniques reported.

In order to stabilize fireflies' movements, Farahani in [54] formulated a new FA that increases convergence speed using Gaussian distribution to move all fireflies to global best in each iteration. Despite the fixed randomization parameter α (also step size), this parameter was modified adaptively in the proposed algorithm. This algorithm was tested on five standard functions. The experimental results showed better performance and more accuracy than the classical firefly algorithm.

Yang in his paper [55] intended to provide an overview of convergence and efficiency studies of meta-heuristics, and tried to provide a framework for analyzing meta-heuristics in terms of convergence and efficiency. Three well-known heuristics were taken into account: simulated annealing [56], particle swarm optimization [57], and the firefly algorithm. The impact of randomization methods Gaussian random walk and Lévy flight on the results of meta-heuristics was also analyzed in this paper. The conclusion was that the most important issue for the newly developed meta-heuristics was to provide a balanced trade-off between local exploitation and global exploration, in order to work better.

Yang [58] formulated a new meta-heuristic FA using the Lévy flights move strategy. Numerical studies and results suggest that the proposed Lévy-flight FA is superior to particle swarm optimization and genetic algorithms in regard to efficiency and success rate. The paper [59] was dedicated to analyze the convergence and efficiency associated with meta-heuristics like swarm intelligence, cuckoo search, firefly algorithm, random walks, and Lévy flights. Although the author Yang tried to discover some mathematical foundations for meta-heuristic behavior, he concluded that despite the fact that the newly developed nature-inspired meta-heuristics worked well on average, mathematical understanding of these partly remains a mystery.

Coelho et al. in their paper [60] proposed a combination of FA with chaotic maps [61] in order to improve the convergence of the classical firefly algorithm. Use of the chaos sequences was shown to be especially effective by easier escape from the local optima. The proposed firefly algorithms used these chaotic maps by tuning the randomized parameter α and light absorption coefficient γ in Eq. (4). A benchmark of reliability-redundancy optimization has

been considered in order to illustrate the power of the proposed FA using chaotic maps. The simulation results of the proposed FA were compared with other optimization techniques presented in literature and it was revealed that the proposed algorithm outperformed the previously best-known solutions available. On the other hand, Gandomi et al. in [62] introduced chaos into FA in order to increase its global search mobility for robust global optimization. Using chaotic maps, they tuned attractiveness β_0 and light absorption coefficient γ in Eq. (4). The authors analyzed the influence of using 12 different chaotic maps on the optimization of benchmark functions. The results showed that some chaotic FA can clearly outperform the classical FA.

Interestingly, Subotic et al. [63] developed the parallelized FA for unconstrained optimization problems tested on standard benchmark functions. Both the speed and quality of the results were placed by the authors and as a result, the parallelized FA obtained much better results over much less execution time. Unfortunately, this conclusion was valid only when more than one population was taken into account. Husselmann et al. in [64] proposed a modified FA on a parallel graphical processing unit (GPU) where the standard benchmark functions were taken for comparison with the classic firefly algorithm. They revealed that the results of this parallel algorithm were more accurate and faster than by the original firefly algorithm, but this was only valid for multi-modal functions. As a matter of fact, the classical FA is well suited to optimizing unimodal functions as very few fireflies are required and, thus, calculation times are dramatically lower.

3.3. Hybrid firefly algorithms

According to the No-Free-Lunch theorem, any two general problem solvers are equivalent when their average performance is compared across all possible problems. That is, they can obtain average results on all classes of problems. Specific heuristics are intended to solve a given set of problems, and normally improve the results of the problem solvers, i.e., heuristics exploit a specific-knowledge of the given problem domains. In fact, these heuristics can also be incorporated into a FA that is a kind of general problem solver. Such hybridized firefly algorithm, in general, improves the results when solving the given problem. In contrast, FA can be also used as a heuristic for hybridizing with other general problem solvers because of its characteristics, i.e., multi-modality and faster convergence.

As the first hybridization of the firefly algorithm, Yang in [66] formulated a new meta-heuristic search method, called Eagle Strategy (ES), which combines the Lévy flight search with the firefly algorithm. The Eagle strategy was inspired by the foraging behavior of eagles. These eagles fly freely over their territory in a random manner similar to Lévy flights [67]. When the prey is seen, the eagle tries to catch it as efficiently as possible. From the algorithmic point of view, the Eagle Strategy consists of two components: random search by Lévy flight, and intensive local search. Interestingly, FA was applied for the local search. This hybrid meta-heuristic algorithm was employed to the Ackley function with Gaussian noise. The results of simulation showed that the Eagle strategy could significantly outperform the particle swarm optimization algorithm in terms of both efficiency and success rate.

The paper of Luthra et al. [68] discussed the hybridization of FA for cryptanalysis of the mono-alphabetic substitution cipher with the operators of mutation and crossover commonly used in Genetic Algorithms. Dominant gene crossover was used as the crossover operator, while the permutation mutation was taken into account for mutation. From the experiments, it was observed that the algorithm worked better for large input cipher text lengths. For smaller input cipher lengths, a larger total number of generations would need to be used.

In [69] Abdullah et al. proposed a Hybrid Evolutionary Firefly Algorithm (HEFA) that combined the classical Firefly Algorithm with the evolutionary operations of the Differential Evolution (DE) method in order to improve searching accuracy and information sharing among the fireflies. This algorithm divided the population of fireflies into two sub-populations according to fitness. In the first, the classical firefly operators were applied, while in the other, the evolutionary operators were adopted from the Differential Evolution [70–72]. The proposed method was used to estimate the parameters in a biological model. The experimental results showed that the accuracy and speed performance of HEFA had significantly outperformed the results produced by the genetic algorithms, particle swarm optimization, evolutionary programming, and the classical firefly algorithm.

In the paper of Fister Jr. et al. [73], the classical FA was hybridized using local search heuristic and applied to graph 3-coloring that is a well-known combinatorial optimization problem [74]. The results of the proposed memetic FA (MFA) were compared with the results of the Hybrid Evolutionary Algorithm (HEA) [75], Tabucol [76], and the evolutionary algorithm with SAW method (EA-SAW) [77] by coloring a suite of medium-scaled random graphs (graphs with 500 vertices) generated using the Culberson random graph generator. The results of FA were very promising and showed the potential that FA could successfully be applied to the other combinatorial optimization problems as well.

Hassanzadeh et al. [78] used FA for training the parameters of the Structure Equivalent Fuzzy Neural Network (SEFNN) in order to recognize the speech. FA improved the ability of generalizing fuzzy neural networks. The results showed that this hybridized algorithm for speech recognition had a higher recognition rate than the classical fuzzy neural network trained by the particle swarm optimization method. On the other hand, Nandy et al. [79] applied a firefly meta-heuristic with back-propagation method to train a feed-forward neural network. Here, the firefly algorithm was incorporated into a back-propagation algorithm in order to achieve a faster and improved convergence rate when training feed-forward neural network. The proposed hybrid algorithm was tested over some standard data sets. It was revealed that the proposed algorithm converged to local optima within a few iterations. These results were compared with the results of the genetic algorithm optimized the same problems. It was observed that the proposed algorithm consumed less time to converge and improved the convergence rate with minimum feed-forward neural network design.

In the paper of Hassanzadeh et al. [80], cellular learning automata were hybridized with the firefly algorithm. In this meta-heuristic, the cellular learning automata were responsible for making diverse solutions in the firefly population, while FA improved these solutions in the sense of local search. The performance of the proposed algorithm was evaluated on five well-known benchmark functions. The experimental results showed that it was able to find the global optima and improve the exploration rate of the standard firefly algorithm.

Farahani [65] proposed three classes of algorithms for improving the performance of the classical firefly algorithm. In the first class, learning automata were used for adapting the absorption and randomization parameters in the firefly algorithm. The second class hybridized the genetic algorithm with FA in order to balance the exploration and exploitation properties of this proposed meta-heuristic by time. The last class used random walk based on a Gaussian distribution in order to move the fireflies over the search space. Experimental results on five benchmark functions showed that the proposed algorithms were highly competitive with the classical firefly and the particle swarm optimization algorithm.

Aruchamy et al. [81] developed the Flexible Neural Tree (FNT) model for micro-array data to predict cancer using the Ant Colony Optimization (ACO). The parameters encoded in the neural tree

were tuned using a firefly algorithm. This proposed model helped to find the optimal solutions at a faster convergence to lower error. In extensive experiments, a comparison was performed between FA and exponential particle swarm optimization (EPSO). The results showed that FA was superior to the EPSO in terms of both efficiency and success.

A list of hybrid firefly algorithms is presented in Table 3.

3.4. Why firefly algorithms are so efficient

As the literature of firefly algorithms is rapidly expanding, a natural question is 'why FA is so efficient?'. There are many reasons for its success. By analyzing the main characteristics of the standard/classical FA, we can highlight the following three points:

- FA can automatically subdivide its population into subgroups, due to the fact that local attraction is stronger than long-distance attraction. As a result, FA can deal with highly non-linear, multi-modal optimization problems naturally and efficiently.
- FA does not use historical individual best s_i^*, and there is no explicit global best g^* either. This avoids any potential drawbacks of premature convergence as those in PSO. In addition, FA does not use velocities, and there is no problem as that associated with velocity in PSO.
- FA has an ability to control its modality and adapt to problem landscape by controlling its scaling parameter such as γ. In fact, FA is a generalization of SA, PSO and DE, as seen clearly in the next paragraph.

In addition, the standard firefly algorithm can be considered as a generalization to particle swarm optimization (PSO), differential evolution (DE), and simulated annealing (SA). From Eq. (4), we can

Table 3
Hybrid firefly algorithms.

Topic	References
Eagle strategy using Lévy walk	[66]
Genetic Algorithms	[68,65]
Differential Evolution	[69]
Memetic algorithm	[73]
Neural network	[78,79]
Cellular learning automata	[80,65]
Ant colony	[81]

see that when β_0 is zero, the updating formula becomes essentially a version of parallel simulated annealing, and the annealing schedule is controlled by α. On the other hand, if we set $\gamma = 0$ in Eq. (4) and set $\beta_0 = 1$ (or more generally, $\beta_0 \in$ Unif(0,1)), FA becomes a simplified version of differential evolution without mutation, and the crossover rate is controlled by β_0. Furthermore, if we set $\gamma = 0$ and replace s_j by the current global best solution g^*, then Eq. (4) becomes a variant of PSO, or accelerated particle swarm optimization, to be more specific. Therefore, the standard firefly algorithm includes DE, PSO and SA as its special cases. As a result, FA can have all the advantages of these three algorithms. Consequently, it is no surprise that FA can perform very efficiently.

4. Applications of firefly algorithms

Nowadays, FA and its variants have been applied for solving many optimization and classification problems, as well as several engineering problems in practice. The taxonomy of the developed firefly algorithm applications is illustrated in Fig. 2. As can be seen from this figure, FA has been applied to the following classes of optimization problems: continuous, combinatorial, constrained, multi-objective, dynamic and noisy optimization. In addition, it has been used for classification problems in machine learning, data mining, and neural networks. Finally, the firefly algorithms are used in almost all branches of engineering. In this review, we focused on the following engineering areas: image processing, industrial optimization, wireless sensor networks, antenna design, business optimization, robotics, semantic web, chemistry, and civil engineering.

4.1. Optimization

Continuous optimization: Most of classical publications about the firefly algorithm, like [35,55,36,58,59] apply to continuous optimization problems. In most cases, the benchmarks of well-known optimization functions were taken into account. In order to complete the comprehensive picture of this area, the paper of Gandomi et al. in [82] was reviewed, where FA was used for solving mixed continuous/discrete structural optimization problems taken from the literature regarding welded beam design, pressure vessel design, helical compression spring design, reinforced concrete beam designs, stepped cantilever beam design, and car side impact design. The optimization results indicated that FA is more efficient than other meta-heuristic algorithms such as particle swarm optimization, genetic algorithms, simulated
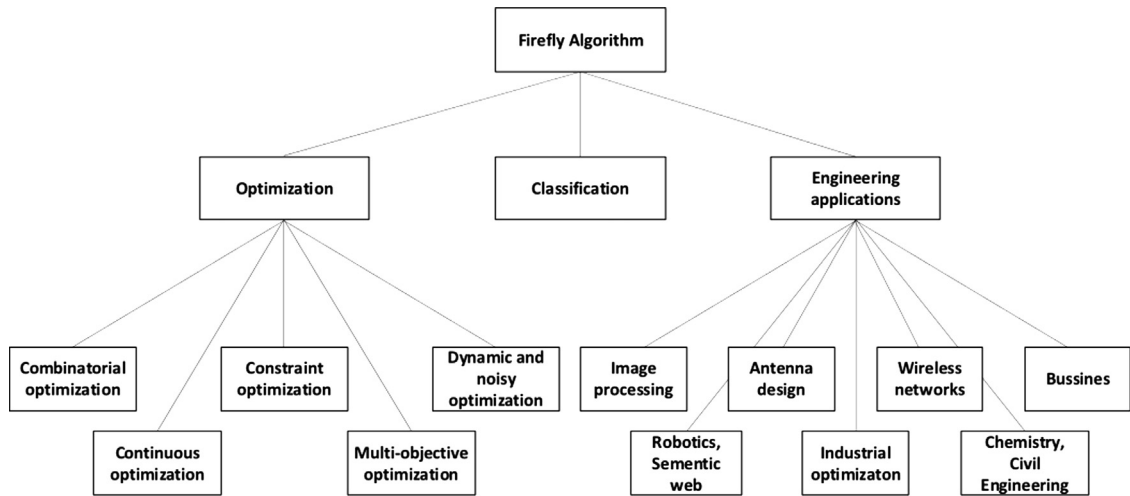

Fig. 2. Taxonomy of firefly applications.

annealing and differential evolution. Although FA was very efficient, oscillatory behavior was observed as the search process approached the optimum design. The overall behavior of FA could be improved by gradually reducing the randomization parameter as the optimization progressed.

Combinatorial optimization: FA is also employed for solving combinatorial optimization problems. Durkota in his BSc thesis [83] adjusted FA to solve the class of discrete problems named Quadratic Assignment Problem (QAP), where the solutions are represented as permutations of integers. In this algorithm, the continuous functions, like attractiveness, distance and movement, were mapped into newly developed discrete functions. The experimental results were obtained on 11 different QAP problems chosen from the public QAPLIB Library. The author reported fast convergence and success rate on the simplest problems, while the algorithm often falls into the local optima by solving the hard problems.

The paper of Sayadi et al. [84] presented a new discrete firefly meta-heuristic for minimizing the makespan for the permutation flow shop scheduling problem that is classified as a NP-hard problem. The results of the proposed algorithm were compared with other existing ant colony optimization technique and indicated that it performs better than the ant colony for some well-known benchmark problems.

The first approach to apply the firefly meta-heuristic to the task graph scheduling problem (NP-hard problem) was performed by Hönig [85]. The results on an extensive experimental benchmark with 36,000 task graph scheduling problems showed that the presented algorithm required less computing time than all the other meta-heuristics used in the test, but these were not as promising when the other performance measures were taken into account.

Jati et al. [86] applied FA to the symmetric traveling salesman problem that is a well-known NP-hard problem. In this evolutionary discrete FA (EDFA), a permutation representation was used, where an element of array represents a city and the index represents the order of a tour. *m*-moves were generated for each firefly using inversion mutation. The simulation results indicated that the EDFA performed very well for some TSPLIB instances when compared to the memetic algorithm. Unfortunately, it can often be trapped into local optimum.

Khadwilard et al. [87] developed FA for solving the Job shop scheduling problem (JSSP). A computational experiment was conducted using five benchmark datasets of the JSSP instance from the well-known OR-Library [88], for finding the lowest makespan. The authors reported that although this FA could found the best known solution in some cases, it was also trapped into the local optima several times.

In [89] the authors Liu et al. proposed a path planning adaptive firefly algorithm. Here, the random and absorption parameters were designed to be adaptive in order to improve the solution quality and convergence speed of the classical firefly algorithm. The simulation tests verified the effectiveness of the improved algorithm and the feasibility of the path-planning method based on the firefly algorithm. In addition, Wang et al. [90] developed a new modified FA (MFA) for solving the path planning problem for uninhabited combat air vehicle (UCAV), where a modification is applied for exchanging information between top fireflies during the process of light intensity updating. This modification accelerated the global convergence speed, while preserving the strong robustness of the classical firefly algorithm. In order to prove the performance of this algorithm, MFA was compared with classical FA and other population-based optimization algorithms like ant colony optimization (ACO) [4], biogeography-based optimization (BBO) [91], differential evolution (DE) [70], evolutionary strategy (ES) [92], genetic algorithm (GA) [93], probability-based

incremental learning (PBIL) [94], particle swarm optimization (PSO) [6], and the stud genetic algorithm (SGA) [95]. The experiment showed that the proposed algorithm was more effective and feasible in UCAV path planning than the other algorithms.

The paper of Kwiecien et al. [96] experimented with FA for optimization of the queuing system that also belongs to a class of NP-hard problems. The queuing theory provides methods for analyzing complex service systems in computer systems, communications, transportation networks and manufacturing. This algorithm was tested by solving two problems found in literature. The results of the experiments performed for selected queuing systems were promising when comparing with the results of genetic algorithms.

Constraint optimization: The first contribution to the domain of constraint optimization was made by Lukasik et al. in [97] who experimented with the classical FA for constrained continuous optimization. The proposed algorithm was tested by solving a set of 14 benchmark constraint problems and the obtained results were compared with the existing particle swarm optimization algorithm [6]. Here, FA was outperformed by particle swarm optimization in their tests. Later, Gandomi et al. [82] studied a set of non-linear constraint optimization problems in engineering and showed that FA can provided better results than other methods such as particle swarm optimization.

Gomes in [98] employed FA for optimizing a structural mass optimization on shape and size by taking dynamic constraints into account. This optimization problem had extreme non-linear behavior with regard to the frequency constraints especially for shape optimization, since eigenvalues are very sensitive to shape modifications. In this algorithm, the constraints were treated as penalty functions that affected the fitness function. It was tested by solving three examples of increasing difficulty, which were compared with the results in the literature. From an engineering point of view, FA performed well in all three cases.

Multi-objective optimization: The contribution of FA to multi-objective optimization [99] is as follows. Yang in [100] formulated a new multi-objective FA for multi-objective optimization that extended FA for producing Pareto optimal front directly. This algorithm was tested on a subset of multi-objective functions with convex, non-convex, and discontinuous Pareto fronts taken from the literature and was then applied for solving design optimization benchmarks in industrial engineering. The results were compared with other algorithms for multi-objective optimization like the vector evaluated genetic algorithm (VEGA) [101], non-dominated sorting genetic algorithm-II (NSGA-II) [102], multi-objective differential evolution (MODE) [103], differential evolution for multi-objective optimization (DEMO) [104], multi-objective bees algorithms (Bees) [105] and strength Pareto evolutionary algorithm (SPEA) [106], and suggested that the proposed algorithm is an efficient multi-objective optimizer.

Abedinia et al. in [107] developed a multi-objective FA for an Environmental/Economic Power Dispatch (EED) problem. This problem was formulated as a non-linear constrained multi-objective problem with the conflict objectives of fuel cost, emission, and system loss. The proposed algorithm run on the IEEE 30- and 118-bus test systems and the results were compared with other known multi-objective algorithms from the literature. The achieved numerical results of the proposed FA demonstrated the feasibility of this for solving the multi-objective EED problem.

A slightly different problem was solved by Niknam et al. in [108], where the authors proposed a multi-objective FA in order to achieve a set of non-dominated (Pareto-optimal) solutions. A tuning of randomization and absorption coefficients based on the usage of chaotic maps and self-adaptive probabilistic mutation strategies were used to improve the overall performance of the algorithm. This algorithm solved the problem of dynamic

economic emission dispatch (DEED) that incorporated combined heat and power units in so-called CHP (Combined Heat and Power) systems. The problem was defined as multi-objective and the performance of the proposed algorithm was successfully validated using numerical simulations.

Dynamic environment: Many real-world problems are mostly time varying optimization problems. Therefore, these require special mechanisms for detecting changes in the environment and then respond to them. FA can be used also for this dynamic environment. Here, a short review is made of those papers that deal with this class of optimization.

Abshouri et al. in [109] hybridized FA with learning automata that were responsible for tuning the firefly parameters, i.e., randomized α, attractiveness β_0 and absorption coefficient γ. The main idea in this algorithm was to split the population of fireflies into a set of interacting swarms that were able to respond quicker to the changing environment. This algorithm was tested on a variety of instances of the multi-modal dynamic moving peaks benchmark [110] and the obtained results were compared with other algorithms for dynamic environments like multi-quantum swarm optimization (mQSO) [111], fast multi-swarm optimization (FMSO) [112], and cellular particle swarm optimization [113]. They showed that this FA significantly outperformed other mentioned algorithms.

In [114], Chai et al. adopted two meta-heuristics: bees and firefly algorithms, in order to find optimal solutions to various types of noisy continuous mathematical functions, using two variables. The results were measured with regard to mean and standard deviations of execution time needed to find an optimal solution. According to the obtained results, the authors concluded that FA performed better than the bees algorithm when the noise levels increased.

Farahani et al. in [115] developed a multi-swarm based FA for dynamic environments that splits the population of particles into sets of interacting swarms. Each swarm interacts locally by an exclusion parameter and globally through an anti-convergence operator. This algorithm was tested solving the moving peaks benchmark [110]. The results were compared according to the performance and accuracy with other particle swarm optimization (PSO) and evolutionary algorithms from the literature and showed that the proposed algorithm significantly outperformed other algorithms in experiments.

FA for dynamic environments that was proposed by Nasiri et al. in [116] had the following characteristics: The best solution found so far was preserved. The randomization parameter α was adapted. The algorithm maintained only one swarm of fireflies. It was applied to a moving peaks benchmark problem [110], which is the most famous benchmark for assessment in dynamic environments, and compared with other algorithms for dynamic optimization, like mQSO [111], AmQSO [117], mCPSO [111], SPSO [118], rSPSO [119] and PSO-CP [120]. The obtained results showed the proper accuracy and convergence rate for the proposed algorithm in comparison with the other mentioned algorithms.

Finally, Sulaiman in [121] presented an application of FA for determining the optimal location and size of distributed generation (DG) in distribution power networks. In this paper, an IEEE 69-bus distribution test system was used to show the effectiveness of the firefly algorithm. The comparison with the genetic algorithm was also conducted to see the performance of FA as to whether it was as good as the genetic algorithm in solving the optimal allocation problem.

The detailed list of optimization algorithms can be seen in Table 4.

4.2. Classification

Classification algorithm is a procedure for selecting a hypothesis from a set of alternatives that best fits a set of observations or

Table 4
Optimization applications.

Topic	References
Continuous optimization	[35,55,36,58,59,82]
Combinatorial optimization	[83–87,89,90,96]
Constrained Optimization	[97,82,98]
Multi-objective optimization	[100,107,108]
Dynamic and noisy environment	[109,114–116,121]

data. Usually, this kind of algorithm appears in machine learning, data mining, and neural networks. Although classifications can be considered as optimization, Holland [122] wrote that a learning (as component part of classification) is viewed as a process of adaptation to a particularly unknown environment, not as an optimization problem. In the rest of this subsection, contributed papers from this area are reviewed.

Banati et al. in [123] hybridized FA with the Rough Set Theory (RST) to find a subset of features and is a valuable preprocessing techniques in machine learning. FA simulated the attraction system of the real fireflies that guides the feature selection procedure. Four different medical datasets from the UCI machine learning data repository [124] were evaluated in order to obtain the performance of the proposed algorithm. The experimental results showed that the proposed algorithm outperformed the other feature selection method like Genetic algorithm (GenRSAR) [125], Ant colony Optimization (AntRSAR) [126], particle swarm optimization (PSO-RSAR) [127] and artificial bee colony (BeeRSAR) [128], in terms of time and optimality.

Horng et al. in [129] proposed FA for training of the radial basis function (RBF) network that is a type of neural network using a radial basis function as its activation function [130]. In these algorithms, each of the fireflies represented a specific RBF network for classification. The performance of the proposed FA was measured according to the percent of correct classification, and the mean square error and complexity on five data sets was taken from the UCI machine repository [124]. The obtained result for the firefly algorithms was compared with the gradient descent algorithm (GD) [131], the genetic algorithm (GA), the particle swarm optimization (PSO), and the artificial bee colony (ABC) algorithm. The experimental results showed that usage of FA obtained satisfactory results over the GD and GA algorithms, but it was not apparently superior to the PSO and ABC algorithms.

In paper [132], Senthilnath et al. applied FA for clustering data objects into groups according to the values of their attributes. The performance of FA for clustering was compared with the results of other nature-inspired algorithms like artificial bees colony (ABC) [7] and particle swarm intelligence (PSO) [6], and other nine methods used in the literature on the 13 test data sets from the literature [133,134]. The performance measure used in the comparison was the classification error percentage (CEP) that is defined as a ratio of the number of misclassified samples in the test data set and total number of samples in the test data set. The authors concluded from the obtained results that FA was the efficient method for clustering.

4.3. Engineering applications

Firefly algorithms have become a crucial technology for solving problems in engineering practice. Nowadays, there are applications for almost every engineering areas. Table 5 shows those engineering areas in which FA was applied with reference to the paper in which the algorithm is described, and the summarized number of papers from a given area. These papers were not analyzed explicitly because of this reviews' scope.

Table 5
Engineering applications.

Engineering area	References	Total
Industrial Optimization	[135–148]	14
Image processing	[149–156]	8
Antenna design	[157–160]	4
Business optimization	[161,162]	2
Civil engineering	[163,164]	2
Robotics	[165,166]	2
Semantic web	[167]	1
Chemistry	[168]	1
Meteorology	[169]	1
Wireless sensor networks	[170]	1

The greatest number of papers in Table 5 belongs to industrial optimization. Image processing is also well covered by firefly algorithms. Antenna optimization consists of four papers. In business optimization, robotics, and civil engineering, there are two papers for each area. Other engineering areas in the table are represented by one paper.

In summary, every day new papers emerge using FA within always new engineering areas. This fact proves that the development of engineering applications using firefly algorithms is very diverse and rapidly expanding.

5. Discussion and further work

This section analyses the main characteristics of the firefly algorithm. That is, the firefly algorithms are suitable for multi-modal optimization, they have fast convergence, obtain good results on function optimization, and are appropriate for combinatorial optimization. The reason for the fast convergence is loosing the diversity of population. In order to keep using this algorithm for large-scale optimization problems, a balancing of the exploration and exploitation, as explained by Yang in [36], should be established.

The results of firefly algorithms depend on the best-found solution within a swarm. Therefore, improving the best solution can improve the search power of the swarm. Experiments using binary firefly algorithms show that this algorithm could successfully be applied to those problems where binary representation is necessary. In some cases, the convergence speed can be improved, when Gaussian or Lévy flight searches are taken in the move function. Using chaotic maps is appropriate for tuning parameters. Parallel firefly algorithms with one swarm improve the results for solving multi-modal functions, while the results are deteriorated on the unimodal functions. In general, parallelism is worthwhile, when the multi-swarm population scheme is employed.

Analyses of papers hybridizing the firefly algorithms show that these are more used as a local search for hybridizing other algorithms, like Eagle Strategy, neural networks, learning automata and genetic algorithms. Amongst other approaches, it is worth mentioning the hybridization of firefly algorithms with neural network that acts as meta-heuristic for setting parameters of firefly algorithm, and the hybridization of FA with local search heuristics. In the latter case, FA acts as a global problem solver. Interestingly, there also exists hybridizations that supports the co-evolution of genetic algorithms and firefly algorithms, and operators borrowed from genetic algorithms. However, there are a lot of possibilities for future developments.

Solving the combinatorial optimization problems illustrates the oscillatory behavior of the firefly algorithm. That is, FA can find the solution faster in some cases, while in other cases the solution cannot be found. Here, the main problem to be solved is how to balance the exploration and exploitation during firefly search. This question also opens up new directions in the further development of this algorithm.

Using firefly algorithms for combinatorial optimization problems demands a mapping of the discrete variables (usually permutation of integer) in the problem space to continuous variables in the search space, where the firefly operators (like moving fireflies) act. Normally, firefly algorithms for solving the constrained optimization problems consider the constraints as a penalty function that punishes the infeasible solutions. This violation affects the fitness function, this means that the more the solution violates the feasibility condition the higher the value of the penalty function.

Dynamic environment is usually solved by FA using the multi-swarm population scheme, which can respond quicker to the changing environment. Additionally, the adaptation for changing environment is faster, when the control parameters of FA are taken into account. Normally, the learning automata was employed to hybridize FA for dynamic environments. The adaptation of control parameters is also useful for solving the multi-objective problems, while for classification problems, despite the adaptation of control parameters, the hybridization of FA is applicable with other classification heuristics like neural networks.

Despite the huge success of FA in practice, mathematical analysis of the algorithm has a very limited literature. In real applications, firefly algorithms often converge quickly, however, there is no theoretical analysis how quickly it can indeed converge. It will be very useful to use theories of dynamical systems, and/or Markov chains as well as complexity to analyze the convergence and stability of all major variants of firefly algorithm. In fact, there are a few important open problems concerning all meta-heuristic algorithms [55].

Another important area is parameter tuning, which is important for all meta-heuristic algorithms. In firefly algorithms, more studies are needed to identify optimal setting for algorithm-dependent parameters so that they can solve a wider range of problems with minimal adjustments of parameter values. This itself is a tough optimization problem. It is possible to even design automatic schemes to tune parameters in an adaptive and intelligent manner, but how to achieve such goals is still an open question.

Applications of firefly algorithms are very diverse, as we have seen from this review. It would be more fruitful to apply to new areas such as bioinformatics, data mining, telecommunications, and large-scale real-world applications [171]. There is no doubt that more applications of firefly algorithms will emerge in the near future.

6. Conclusions

FA algorithm has widely expanded its application domains since its establishment in 2008. Nowadays, there is practically no domain where FA had not been applied. Moreover, the development areas of this algorithm are very dynamic, because new applications appear almost every day.

This paper performs a comprehensive review of the firefly algorithm. It has shown that this algorithm:

- possesses multi-modal characteristics,
- can handle multi-modal problems efficiently,
- has a fast convergence rate,
- can be used as a general, global problem solver as well as a local search heuristic,
- is applicable to every problem domain.

FA solves problems from the following domains: continuous optimization, combinatorial optimization, constraint optimization, multi-objective optimization, as well as dynamic and noisy environments, and even classification. The important domain of its applicability is represented by engineering application.

The objective of this review was twofold. Firstly, it has summarized the status of applications within various application areas. This review has proved that FA can be practically applied within every problem domain. Secondly, this review has shown that FA is simple, flexible and versatile, which is very efficient in solving a wide range of diverse real-world problems. At the same, we have provided some open questions and challenges that can inspire further research in these areas in the near future, like ensemble of parameters in [172].

References

[1] C. Blum, X. Li, Swarm intelligence in optimization, in: C. Blum, D. Merkle (Eds.), Swarm Intelligence: Introduction and Applications, Springer Verlag, Berlin, 2008, pp. 43–86.

[2] M. Beekman, G. Sword, S. Simpson, Biological foundations of swarm intelligence, in: C. Blum, D. Merkle (Eds.), Swarm Intelligence: Introduction and Applications, Springer Verlag, Berlin, 2008, pp. 3–41.

[3] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in: Proceedings of NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, 1989, pp. 26–30.

[4] M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 11–32.

[5] P. Korošec, J. Šilc, B. Filipič, The differential ant-stigmergy algorithm, Information Sciences 192 (2012) 82–97, http://dx.doi.org/10.1016/j.ins.2010.05.002.

[6] J. Kennedy, R. Eberhart, The particle swarm optimization: social adaptation in information processing, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 379–387.

[7] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization 39 (3) (2007) 459–471.

[8] I. Fister, I. Fister Jr., J. Brest, V. Žumer, Memetic artificial bee colony algorithm for large-scale global optimization, in: IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.

[9] X.S. Yang, Firefly algorithm, Nature-Inspired Metaheuristic Algorithms 20 (2008) 79–90.

[10] A. Gandomi, X.-S. Yang, S. Talatahari, A. Alavi, Metaheuristic in modeling and optimization, in: A. Gandomi, X.-S. Yang, S. Talatahari, A. Alavi (Eds.), Metaheuristic Application in Structures and Infrastructures, Elsevier, Waltham, 2013, pp. 1–24.

[11] X.-S. Yang, Optimization and metaheuristic algorithms in engineering, in: X.-S. Yang et al. (Ed.), Metaheuristic in Water Geotechnical and Transport Engineering, Elsevier, Waltham, 2013, pp. 1–23.

[12] M. Sahab, V. Toropov, A. Gandomi, Traditional and modern optimization techniques – theory and application, in: A.H. Gandomi et al. (Ed.), Metaheuristic Applications in Structures and Infrastructures, Elsevier, Waltham, 2013, pp. 26–47.

[13] X.S. Yang, S. Deb, Cuckoo search via levy flights, in: World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, 2009, pp. 210–214.

[14] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: C. Cruz, J. Gonzlez, G.T.N. Krasnogor, D.A. Pelta (Eds.), Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, vol. 284, Springer Verlag, Berlin, 2010, pp. 65–74.

[15] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Communications in Nonlinear Science and Numerical Simulation 17 (12) (2012) 4831–4845, http://dx.doi.org/10.1016/j.cnsns.2012.05.010.

[16] A. Hatamlou, S. Abdullah, H. Nezamabadi-pour, A combined approach for clustering based on k-means and gravitational search algorithms, Swarm and Evolutionary Computation 6 (2012) 47–52.

[17] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Information Sciences 222 (2013) 175–184.

[18] M. Črepinšek, M. Mernik, S. Liu, Analysis of exploration and exploitation in evolutionary algorithms by ancestry trees, International Journal of Innovative Computing and Applications 3 (1) (2011) 11–19.

[19] K. Tashkova, J. Šilc, N. Atanasova, S. Džeroski, Parameter estimation in a nonlinear dynamic model of an aquatic ecosystem with meta-heuristic optimization, Ecological Modelling 226 (2012) 36–61, http://dx.doi.org/10.1016/j.ecolmodel.2011.11.029.

[20] A. Prügel-Bennett, Benefits of a population: five mechanisms that advantage population-based algorithms, IEEE Transactions on Evolutionary Computation 14 (4) (2010) 500–517.

[21] S.A. Lee, k-phase oscillator synchronization for graph coloring, Mathematics in Computer Science 3 (1) (2010) 61–72.

[22] H. Hernández, C. Blum, Distributed graph coloring: an approach based on the calling behavior of japanese tree frogs, Swarm Intelligence 6 (2) (2012) 117–150.

[23] S.M. Lewis, C.K. Cratsley, Flash signal evolution, mate choice, and predation in fireflies, Annual Revision in Entomology 53 (2008) 293–321.

[24] J.R. De Wet, K. Wood, M. DeLuca, D. Helinski, S. Subramani, Firefly luciferase gene: structure and expression in mammalian cells, Molecular and Cellular Biology 7 (2) (1987) 725–737.

[25] A. Brasier, J. Tate, J. Habener, et al., Optimized use of the firefly luciferase assay as a reporter gene in mammalian cell lines, BioTechniques 7 (10) (1989) 11–16.

[26] B.L. Strehler, J.R. Totter, Firefly luminescence in the study of energy transfer mechanisms. I. Substrate and enzyme determination, Archives of Biochemistry and Biophysics 40 (1) (1952) 28–41.

[27] M. Deluca, Firefly luciferase, Advances in Enzymology and Related Areas of Molecular Biology 40 (2006) 37–68.

[28] H. Seliger, W. McElroy, Spectral emission and quantum yield of firefly bioluminescence, Archives of Biochemistry and Biophysics 88 (1) (1960) 136–141.

[29] C. Darwin, The Origin of Species, P.F. Collier, New York, 1859.

[30] A. Eiben, J. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, 2003.

[31] A. Eiben, S. Smit, Parameter tuning for configuring and analysing evolutionary algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 19–31.

[32] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.

[33] X.S. Yang, Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, Springer, 2009, pp. 169–178.

[34] S. Das, S. Maity, B.-Y. Qu, P.N. Suganthan, Real-parameter evolutionary multimodal optimization – a survey of the state-of-the-art, Swarm and Evolutionary Computation 1 (2) (2011) 71–88.

[35] X.S. Yang, Firefly algorithm, stochastic test functions and design optimisation, International Journal of Bio-Inspired Computation 2 (2) (2010) 78–84.

[36] X.S. Yang, Review of meta-heuristics and generalised evolutionary walk algorithm, International Journal of Bio-Inspired Computation 3 (2) (2011) 77–84.

[37] R. Parpinelli, H. Lopes, New inspirations in swarm intelligence: a survey, International Journal of Bio-Inspired Computation 3 (1) (2011) 1–16.

[38] K. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine (2002) 52–67.

[39] D. Monismith, B. Mayfield, Slime mold as a model for numerical optimization, in: IEEE Swarm Intelligence Symposium, 2008, pp. 1–8.

[40] T. Havens, G. Alexander, C. Abbott, J. Keller, M. Skubic, M. Rantz, Contour tracking of human exercises, in: IEEE Workshop on Computational Intelligence for Visual Intelligence, 2009, pp. 22–28.

[41] X. Feng, F. Lau, D. Gao, A new bio-inspired approach to the traveling salesman problem, in: Complex Sciences, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 5, Springer Verlag, Berlin, 2009, pp. 1310–1321.

[42] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, vol. 284, Springer Verlag, Berlin, 2010, pp. 65–74.

[43] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[44] O. Haddad, A. Afshar, MBO algorithm, a new heuristic approach in hydrosystems design and operation, in: The First International Conference on Managing Rivers in the 21st Century, 2004, pp. 499–504.

[45] H. Zang, S. Zhang, K. Hapeshi, A review of nature-inspired algorithms, Journal of Bionic Engineering 7 (2010) 232–237.

[46] Surafel Luleseged Tilahun, Hong Choon Ong, Modified firefly algorithm, Journal of Applied Mathematics 2012 (2012), doi: http://dx.doi.org/10.1155/2012/467631.

[47] S. Palit, S. Sinha, M. Molla, A. Khanra, M. Kule, A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm, in: The Second International Conference on Computer and Communication Technology (ICCCT-2011), IEEE, 2011, pp. 428–432.

[48] R. Falcon, M. Almeida, A. Nayak, Fault identification with binary adaptive fireflies in parallel and distributed systems, in: IEEE Congress on Evolutionary Computation (CEC-2011), IEEE, 2011, pp. 1359–1366.

[49] K. Chandrasekaran, S. Simon, Network and reliability constrained unit commitment problem using binary real coded firefly algorithm, International Journal of Electrical Power & Energy Systems 43 (1) (2012) 921–932.

[50] A. Behrouz, Cryptography and Network Security, Tata McGraw Hill, New York, 2007.

[51] H. Yang, M. Elhadef, A. Nayak, X. Yang, Network fault diagnosis: an artificial immune system approach, in: Proceedings of the 14th IEEE International Conference on Parallel and Distributed Systems, Melbourne, Australia, 2008, pp. 463–469.

[52] R. Falcon, M. Almeida, A. Nayak, A binary particle swarm optimization approach to fault diagnosis in parallel and distributed systems, in: Proceedings of IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, 2010, pp. 41–48.

[53] M. Carrion, J.M. Arroyo, A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem, IEEE Transactions on Power Systems 21 (2006) 1371–1378.

[54] S.M. Farahani, A. Abshouri, B. Nasiri, M. Meybodi, A gaussian firefly algorithm, International Journal of Machine Learning and Computing 1 (5) (2011) 448–454.

[55] X.S. Yang, Metaheuristic optimization: algorithm analysis and open problems, in: P. Pardalos, S. Rebennack (Eds.), Experimental Algorithms, Lecture notes in Computer Science, vol. 6630, Springer Verlag, Berlin, 2011, pp. 21–32.

[56] D. Bertsimas, J. Tsitsiklis, Simulated annealing, Statistical Science 8 (1993) 10–15.

[57] J. Kennedy, R.C. Eberhart, Swarm intelligence in cellular robotic systems, in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.

[58] X.S. Yang, Firefly algorithm, levy flights and global optimization, in: M. Bramer, R. Ellis, M. Petridis (Eds.), Research and Development in Intelligent Systems XXVI, Springer, 2010, pp. 209–218.

[59] X.S. Yang, Efficiency analysis of swarm intelligence and randomization techniques, Journal of Computational and Theoretical Nanoscience 9 (2) (2012) 189–198.

[60] L. dos Santos Coelho, D.L. de Andrade Bernert, V.C. Mariani, A chaotic firefly algorithm applied to reliability-redundancy optimization, in: IEEE Congress on Evolutionary Computation (CEC 2011), vol. 18, IEEE, 2013, pp. 89–98.

[61] S. Strogatz, Nonlinear Dynamics and Chaos, Perseus Publishing, Massachusett, 2000.

[62] A. Gandomi, X.-S. Yang, S. Talatahari, A. Alavi, Firefly algorithm with chaos, Communications in Nonlinear Science and Numerical Simulation 18 (1) (2013) 89–98, http://dx.doi.org/10.1016/j.cnsns.2012.06.009.

[63] M. Subutic, M. Tuba, N. Stanarevic, Parallelization of the firefly algorithm for unconstrained optimization problems, in: Latest Advances in Information Science and Applications, 2012, pp. 264–269.

[64] A. Husselmann, K. Hawick, Parallel Parametric Optimisation with Firefly Algorithms on Graphical Processing Units, Technical Report CSTN-141, 2012.

[65] S. Farahani, A. Abshouri, B. Nasiri, M. Meybodi, Some hybrid models to improve firefly algorithm performance, International Journal of Artificial Intelligence 8 (12) (2012) 97–117.

[66] X.S. Yang, S. Deb, Eagle strategy using levy walk and firefly algorithms for stochastic optimization, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), 2010, pp. 101–111.

[67] C. Brown, L.S. Liebovitch, R. Glendon, Lévy flights in dobe ju/hoansi foraging patterns, Human Ecology 35 (2007) 129–138.

[68] J. Luthra, S. Pal, A hybrid firefly algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution cipher, in: World Congress on Information and Communication Technologies (WICT 2011), IEEE, 2011, pp. 202–206.

[69] A. Abdullah, S. Deris, M. Mohamad, S. Hashim, A new hybrid firefly algorithm for complex and nonlinear problem, Distributed Computing and Artificial Intelligence (2012) 673–680.

[70] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[71] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657.

[72] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Transactions on Evolutionary Computation 15 (1) (2011) 4–31.

[73] I. FisterJr., X.S. Yang, I. Fister, J. Brest, Memetic firefly algorithm for combinatorial optimization, in: B. Filipič, J. Šilc (Eds.), Bioinspired Optimization Methods and Their Applications: Proceedings of the Fifth International Conference on Bioinspired Optimization Methods and their Applications—BIOMA 2012, Jozef Stefan Institute, 2012, pp. 75–86.

[74] I. Fister, M. Mernik, B. Filipič, Graph 3-coloring with a hybrid self-adaptive evolutionary algorithm, Computational Optimization and Applications (2012) 1–30.

[75] P. Galinier, J.K. Hao, Hybrid evolutionary algorithms for graph coloring, Journal of Combinatorial Optimization 3 (1999) 379–397.

[76] A. Hertz, D. de Werra, Using tabu search techniques for graph coloring, Computing 39 (1987) 345–351.

[77] A.E. Eiben, K. Hauw, J.I. Hemert, Using tabu search techniques for graph coloring, Journal of Heuristics 4 (1998) 25–46.

[78] T. Hassanzadeh, K. Faez, G. Seyfi, A speech recognition system based on structure equivalent fuzzy neural network trained by firefly algorithm, in: 2012 International Conference on Biomedical Engineering (ICoBE), IEEE, 2012, pp. 63–67.

[79] S. Nandy, P. Sarkar, A. Das, Analysis of a nature inspired firefly algorithm based back-propagation neural network training, arXiv preprint arXiv:1206.5360.

[80] T. Hassanzadeh, M. Meybodi, A new hybrid algorithm based on firefly algorithm and cellular learning automata, in: The 20th Iranian Conference on Electrical Engineering, IEEE, 2012, pp. 628–633.

[81] R. Aruchamy, K.D.D. Vasantha, A comparative performance study on hybrid swarm model for micro array data, International Journal of Computer Applications 30 (6) (2011) 10–14.

[82] A.H. Gandomi, X.S. Yang, A.H. Alavi, Mixed variable structural optimization using firefly algorithm, Computers & Structures 89 (23–24) (2011) 2325–2336.

[83] K. Durkota, Implementation of a Discrete Firefly Algorithm for the QAP Problem within the Sage Framework, Master's Thesis, Czech Technical University, Prague, 2009.

[84] M. Sayadi, R. Ramezanian, N. Ghaffari-Nasab, A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems, International Journal of Industrial Engineering Computations 1 (1) (2010) 1–10.

[85] U. Hönig, A firefly algorithm-based approach for scheduling task graphs in homogeneous systems, in: Informatics, ACTA Press, 2010, pp. 24–33.

[86] G. Jati, Evolutionary discrete firefly algorithm for travelling salesman problem, Adaptive and Intelligent Systems (2011) 393–403.

[87] A. Khadwilard, S. Chansombat, T. Thepphakorn, P. Thapatsuwan, W. Chainate, P. Pongcharoen, Application of firefly algorithm and its parameter setting for job shop scheduling, in: The First Symposium on Hands-On Research and Development, 2011, pp. 1–10.

[88] J. Beasley, Or-library: distributing test problems by electronic mail, Journal of the Operational Research Society 41 (1990) 1069–1072.

[89] C. Liu, Z. Gao, W. Zhao, A new path planning method based on firefly algorithm, in: The Fifth International Joint Conference on Computational Sciences and Optimization (CSO-2012), IEEE, 2012, pp. 775–778.

[90] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, A modified firefly algorithm for UCAV path planning, International Journal of Hybrid Information Technology 5 (3) (2012) 123–144.

[91] D. Simon, Biogeography-based optimization, IEEE Transactions on Evolutionary Computation 12 (2008) 702–713.

[92] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, Oxford, 1996.

[93] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, MA, 1996.

[94] B. Shumeet, Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Technical Report, Carnegie Mellon University, Pittsburgh, 1994.

[95] W. Khatib, P. Fleming, The stud GA: a mini revolution?, in: A. Eiben, T. Back, M. Schoenauer, H. Schwefel (Eds.), Parallel Problem Solving from Nature, Springer Verlag, New York, 1998, pp. 683–691.

[96] J. Kwiecień, B. Filipowicz, Firefly algorithm in optimization of queueing systems, Technical Sciences 60 (2) (2012) 363–368.

[97] S. Łukasik, S. Żak, Firefly algorithm for continuous constrained optimization tasks, in: N.T. Nguyen, R. Kowalczyk, S.-M. Chen (Eds.), Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems, Springer, 2009, pp. 97–106.

[98] H. Gomes, A firefly metaheuristic structural size and shape optimisation with natural frequency constraints, International Journal of Metaheuristics 2 (1) (2012) 38–55.

[99] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.

[100] X.S. Yang, Multiobjective firefly algorithm for continuous optimization, Engineering with Computers 29 (2013) 175–184.

[101] R. Falcon, M. Almeida, A. Nayak, Multiple objective optimization with vector evaluated genetic algorithms, in: Proceedings of the First International Conference, Genetic Algorithms, 1985, pp. 93–100.

[102] K. Deb, A. Pratap, S. Agarwal, T. Mayarivan, A fast and elitist multiobjective algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[103] B. Babu, A. Gujarathi, Multi-objective differential evolution (mode) for optimization of supply chain planning and management, in: IEEE Congress on Evolutionary Computation (CEC 2007), IEEE, 2007, pp. 2732–2739.

[104] T. Robič, B. Filipič, Demo: differential evolution for multi-objective optimization, in: C. Coello, A.A.E. Zitzler (Eds.), Lecture Notes on Computer Science, EMO 2005, Springer Verlag, New York, 2005, pp. 520–533.

[105] R. Falcon, M. Almeida, A. Nayak, Multi-objective optimisation using the bees algorithm, in: The Third International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007), Whittles, Dunbeath, Scotland, 2007, pp. 1–5.

[106] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computattation 3 (4) (1999) 257–271.

[107] O. Abedinia, N. Amjady, M. Naderi, Multi-objective environmental/economic dispatch using firefly technique, in: The 11th International Conference on Environment and Electrical Engineering (EEEIC 2012), IEEE, 2012, pp. 461–466.

[108] T. Niknam, R. Azizipanah-Abarghooee, A. Roosta, B. Amiri, A new multi-objective reserve constrained combined heat and power dynamic economic emission dispatch, Energy 42 (2012) 530–545.

[109] A. Abshouri, M. Meybodi, A. Bakhtiary, New firefly algorithm based on multi swarm & learning automata in dynamic environments, in: IEEE Proceedings, 2011, pp. 73–77.

[110] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, in: Proceedings of the Congress on Evolutionary Computation 99, vol. 3, IEEE, 1999, pp. 1–8.

[111] T. Blackwell, J. Branke, Multiswarms, exclusion, and anticonvergence in dynamic environments, IEEE Transactions on Evolutionary Computation 10 (2006) 459–472.

[112] C. Li, S. Yang, Fast multi-swarm optimization for dynamic optimization problems, in: The Fourth International Conference on Natural Computation, vol. 7, Jinan, Shandong, China, 2008, pp. 624–628.

[113] A. Hashemi, M. Meybodi, Cellular PSO: a PSO for dynamic environments, Advances in Computation and Intelligence (2009) 422–433.

[114] N. Chai-ead, P. Aungkulanon, P. Luangpaiboon, Bees and firefly algorithms for noisy non-linear optimization problems, in: Proceedings of the International Multi Conference of Engineering and Computer Scientists, vol. 2, 2011, pp. 1449–1454.

[115] S. Farahani, B. Nasiri, M. Meybodi, A multiswarm based firefly algorithm in dynamic environments, in: The Third International Conference on Signal Processing Systems (ICSPS 2011), vol. 3, 2011, pp. 68–72.

[116] B. Nasiri, M. Meybodi, Speciation based firefly algorithm for optimization in dynamic environments, International Journal of Artificial Intelligence 8 (12) (2012) 118–132.

[117] T. Blackwell, J. Branke, X. Li, Particle swarms for dynamic optimization problems, Swarm Intelligence 20 (2008) 193–217.

[118] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, Information Sciences 178 (15) (2008) 3096–3109.

[119] S. Bird, X. Li, Using regression to improve local convergence, in: IEEE Congress on Evolutionary Computation (CEC 2007), IEEE, 2007, pp. 592–599.

[120] L. Liu, S. Yang, D. Wang, Particle swarm optimization with composite particles in dynamic environments, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40 (6) (2010) 1634–1648.

[121] M. Sulaiman, M. Mustafa, A. Azmi, O. Aliman, S. Abdul Rahim, Optimal allocation and sizing of distributed generation in distribution system via firefly algorithm, in: IEEE International Power Engineering and Optimization Conference (PEOCO 2012), IEEE, 2012, pp. 84–89.

[122] J. Holland, Adaptation in Natural and Artificial Systems, 1st edition, MIT Press, Cambridge, USA, 1992.

[123] H. Banati, M. Bajaj, Firefly based feature selection approach, International Journal of Computer Science Issues (IJCSI) 8 (4) (2011) 473–480.

[124] A. Asuncion, D. Newman, UCI Machine Learning Repository, 2012. url: ⟨http://www.ics.uci.edu/mlearn/MLRepository.html⟩ (accessed 1.12.12).

[125] H. Liu, A. Abraham, Y. Li, Nature inspired population-based heuristics for rough set reduction, Rough Set Theory: A True Landmark in Data Analysis (2009) 261–278.

[126] R. Jensen, Q. Shen, Finding rough set reducts with ant colony optimization, in: Proceedings of the 2003 UK Workshop on Computational Intelligence, vol. 1, 2003, pp. 15–22.

[127] B. Yue, W. Yao, A. Abraham, H. Liu, A new rough set reduct algorithm based on particle swarm optimization, Bio-Inspired Modeling of Cognitive Tasks 20 (2007) 397–406.

[128] N. Suguna, K. Thanushkodi, A novel rough set reduct algorithm for medical domain based on bee colony optimization, Journal of Computing 2 (6) (2010) 49–54.

[129] M. Horng, Y. Lee, M. Lee, R. Liou, Firefly meta-heuristic algorithm for training the radial basis function network for data classification and disease diagnosis, in: R. Parpinelli, H. Lopes (Eds.), Theory and New Applications of Swarm Intelligence, InTech, 2012, pp. 1–19.

[130] T. Ou, Y. Oyang, C. Chen, A novel radial basis function network classifier with centers set by hierarchical clustering, in: Proceedings of the International Joint Conference on Neural Network, vol. 3, 2005, pp. 1383–1388.

[131] N. Karayiannis, Reformulated radial basis function neural network trained by gradient descent, IEEE Transactions on Neural Networks 3 (1999) 657–671.

[132] J. Senthilnath, S. Omkar, V. Mani, Clustering using firefly algorithm: performance study, Swarm and Evolutionary Computation 1 (3) (2011) 164–171.

[133] D. Karaboga, C. Ozturk, A novel cluster approach: artificial bee colony (ABC) algorithm, Applied Soft Computing 11 (2010) 652–657.

[134] I. Falco, A. Cioppa, E. Tarantino, Facing classification problems with particle swarm optimization, Applied Soft Computing 7 (2007) 652–658.

[135] T. Apostolopoulos, A. Vlachos, Application of the firefly algorithm for solving the economic emissions load dispatch problem, International Journal of Combinatorics 2011 (2011) 23 pp.

[136] X.S. Yang, S.S.S. Hosseini, A.H. Gandomi, Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, Applied Soft Computing 12 (3) (2011) 1180–1186.

[137] T. Mauder, C. Sandera, J. Stetina, M. Seda, Optimization of the quality of continuously cast steel slabs using the firefly algorithm, Materiali in Tehnologije 45 (4) (2011) 347–350.

[138] A. Chatterjee, G. Mahanti, A. Chatterjee, Design of a fully digital controlled reconfigurable switched beam concentric ring array antenna using firefly and particle swarm optimization algorithm, Progress in Electromagnetics Research B 36 (2012) 113–131.

[139] A. Kazemzadeh, Optimum design of structures using an improved firefly algorithm, International Journal of Optimization in Civil Engineering 2 (1) (2011) 327–340.

[140] P. Aungkulanon, N. Chai-Ead, P. Luangpaiboon, Simulated manufacturing process improvement via particle swarm optimisation and firefly algorithms, in: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. 2, 2011, pp. 1–6.

[141] B. Rampriya, K. Mahadevan, S. Kannan, Unit commitment in deregulated power system using lagrangian firefly algorithm, in: IEEE International Conference on Communication Control and Computing Technologies (ICCCCT 2010), IEEE, 2010, pp. 389–393.

[142] K. Chandrasekaran, S. Simon, Demand response scheduling in SCUC problem for solar integrated thermal system using firefly algorithm, in: IET Conference on Renewable Power Generation (RPG 2011), IET, 2011, pp. 1–8.

[143] H. Hu, Fa-based optimal strategy of trains energy saving with energy materials, Advanced Materials Research 485 (2012) 93–96.

[144] L. Dekhici, P. Borne, B. Khaled, et al., Firefly algorithm for economic power dispatching with pollutants emission, Informatica Economică 16 (2) (2012) 45–57.

[145] O. Roeva, T. Slavov, Firefly algorithm tuning of PID controller for glucose concentration control during E. coli fed-batch cultivation process, in:

Proceedings of the Federated Conference on Computer Science and Information Systems, IEEE, 2012, pp. 455–462.

[146] O. Roeva, Optimization of E. coli cultivation model parameters using firefly algorithm, International Journal of Bioautomation 16 (2012) 23–32.

[147] O. Abedinia, N. Amjady, K. Kiani, H. Shayanfar, Fuzzy PID based on firefly algorithm: load frequency control in deregulated environment, in: The 2012 International Conference on Bioinformatics and Computational Biology, 2012, pp. 1–7.

[148] R. Dutta, R. Ganguli, V. Mani, Exploring isospectral spring–mass systems with firefly algorithm, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 467 (2011) 3222–3240.

[149] Y. Zhang, L. Wu, A novel method for rigid image registration based on firefly algorithm, International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC) 2 (2) (2012) 141–146.

[150] M. Horng, T. Jiang, The codebook design of image vector quantization based on the firefly algorithm, Computational Collective Intelligence. Technologies and Applications (2010) 438–447.

[151] M. Horng, Vector quantization using the firefly algorithm for image compression, Expert Systems with Applications 39 (1) (2012) 1078–1091.

[152] T. Hassanzadeh, H. Vojodi, F. Mahmoudi, Non-linear grayscale image enhancement based on firefly algorithm, in: B.K. Panigrahi, P.N. Suganthan, S. Das, S.C. Satapathy (Eds.), Swarm, Evolutionary, and Memetic Computing, Springer, 2011, pp. 174–181.

[153] T. Hassanzadeh, H. Vojodi, A. Moghadam, An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm, in: The Seventh International Conference on Natural Computation (ICNC 2011), vol. 3, IEEE, 2011, pp. 1817–1821.

[154] M. Mohd Noor, A. Ahmad, Z. Hussain, K. Ahmad, A. Ainihayati, Multilevel thresholding of gel electrophoresis images using firefly algorithm, in: IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2011), IEEE, 2011, pp. 18–21.

[155] M. Horng, T. Jiang, Multilevel image thresholding selection based on the firefly algorithm, in: Ubiquitous Intelligence & Computing and Seventh International Conference on Autonomic & Trusted Computing (UIC/ATC), IEEE, 2010, pp. 58–63.

[156] M. Horng, R. Liou, Multilevel minimum cross entropy threshold selection based on the firefly algorithm, Expert Systems with Applications 38 (12) (2011) 14805–14811.

[157] B. Basu, G. Mahanti, Fire fly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna, Progress in Electromagnetics Research B 32 (2011) 169–190.

[158] M. Zaman, A. Matin, et al., Nonuniformly spaced linear antenna array design using firefly algorithm, International Journal of Microwave Science and Technology 2012 (2012) 256759 (8 pp).

[159] A. Chatterjee, G. Mahanti, Minimization in variations of different parameters in different φ planes of a small-size concentric ring array antenna using firefly algorithm, Annals of Telecommunications 20 (2012) 1–8.

[160] B. Basu, G. Mahanti, Thinning of concentric two-ring circular array antenna using fire fly algorithm, Scientia Iranica 19 (6) (2012) 1802–1809.

[161] X.S. Yang, S. Deb, S. Fong, Accelerated particle swarm optimization and support vector machine for business optimization and applications, Networked Digital Technologies (2011) 53–66.

[162] G. Giannakouris, V. Vassiliadis, G. Dounias, Experimental study on a hybrid nature-inspired algorithm for financial portfolio optimization, Artificial Intelligence: Theories, Models and Applications (2010) 101–111.

[163] S. Talatahari, A. Gandomi, G. Yun, Optimum design of tower structures using firefly algorithm, The Structural Design of Tall and Special Buildings doi: http://dx.doi.org/10.1002/tal.1043.

[164] S. Gholizadeh, H. Barati, A comparative study of three metaheuristics for optimum design of trusses, International Journal of Optimization in Civil Engineering 3 (2012) 423–441.

[165] B. Jakimovski, B. Meyer, E. Maehle, Firefly flashing synchronization as inspiration for self-synchronization of walking robot gait patterns using a decentralized robot control architecture, Architecture of Computing Systems —ARCS 2010 (2010) 61–72.

[166] S. Severin, J. Rossmann, A comparison of different metaheuristic algorithms for optimizing blended PTP movements for industrial robots, Intelligent Robotics and Applications (2012) 321–330.

[167] C. Pop, V. Chifu, I. Salomie, R. Baico, M. Dinsoreanu, G. Copil, A hybrid firefly-inspired approach for optimal semantic web service composition, Scalable Computing: Practice and Experience 12 (2011) 363–369.

[168] S.E. Fateen, A. Bonilla-Petriciolet, G.P. Rangaiah, Evaluation of covariance matrix adaptation evolution strategy, shuffled complex evolution and firefly algorithms for phase stability, phase equilibrium and chemical equilibrium problems, Chemical Engineering Research and Design 90 (12) (2012) 2051–2071.

[169] A.F.d. Santos, H.F.d. Campos Velho, E.F. Luz, S.R. Freitas, G. Grell, M.A. Gan, Firefly optimization to determine the precipitation field on South America, Inverse Problems in Science and Engineering (2013) 1–16.

[170] M. Breza, J. McCann, Lessons in implementing bio-inspired algorithms on wireless sensor networks, in: NASA/ESA Conference on Adaptive Hardware and Systems (AHS'08), IEEE, 2008, pp. 271–276.

[171] A. Zamuda, J. Brest, Population reduction differential evolution with multiple mutation strategies in real world industry challenges, in: ICAISC (SIDE-EC), 2012, pp. 154–161.

[172] R. Mallipeddi, P. Suganthan, Q. Pan, M. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Applied Soft Computing 11 (2) (2011) 1679–1696.