# Chapter 8
# Luminous Insect Inspired Algorithms

**Abstract** In this chapter, we present three algorithms that are inspired by the flashing behaviour of luminous insects, i.e., firefly algorithm (FA), glowworm swarm optimization (GlSO) algorithm, and bioluminescent swarm optimization (BiSO) algorithm. We first describe the general knowledge of the luminous insects in Sect. 8.1. Then, the fundamentals, performances and selected applications of FA, GlSO algorithm and BiSO algorithm are introduced in Sects. 8.2, 8.3 and 8.4, respectively. Finally, Sect. 8.5 summarises this chapter.

## 8.1 Introduction

The flashing light of luminous insects is an amazing sight in the summer sky. More information about firefly flash code evolution please refer to (Buck and Case 2002). In this chapter, we presented three algorithms that are inspired by the flashing behaviour of luminous insects, i.e., firefly algorithm (FA), glowworm swarm optimization (GlSO) algorithm, and bioluminescent swarm optimization (BiSO) algorithm.

## 8.2 Firefly Algorithm

### *8.2.1 Fundamentals of Firefly Algorithm*

Firefly algorithm (FA) is a nature-inspired, optimization algorithm which is based on the social (flashing) behaviour of fireflies, or lighting bugs, in the summer sky in the tropical temperature regions (Yang 2008, 2009, 2010b). In the FA, physical entities (fireflies) are randomly distributed in the search space. They carry a bioluminescence quality, called luciferin, as a signal to communicate with other fireflies, especially to prey attractions (Babu and Kannan 2002). In detail, each

B. Xing and W.-J. Gao, *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Intelligent Systems Reference Library 62,
DOI: 10.1007/978-3-319-03404-1_8, 

firefly is attracted by the brighter glow of other neighbouring fireflies. The attractiveness decreases as their distance increases. If there is no brighter one than a particular firefly, it will move randomly. Its main merit is the fact that the FA uses mainly real random numbers and is based on the global communication among the swarming particles (i.e., the fireflies), and as a result, it seems more effective in multi-objective optimization.

Normally, FA uses the following three idealized rules to simplify its search process to achieve an optimal solution (Yang 2010b):

- Fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex, that means no mutation operation will be done to alter the attractiveness fireflies have for each other;
- The sharing of information or food between the fireflies is proportional to the attractiveness that increases with a decreasing Cartesian or Euclidean distance between them due to the fact that the air absorbs light. Thus for any two flashing fireflies, the less bright one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly; and
- The brightness of a firefly is determined by the landscape of the objective function. For the maximization problems, the light intensity is proportional to the value of the objective function.

Furthermore, there are two important issues in the FA that are the variation of light intensity or brightness and formulation of attractiveness. Yang (2008) simplifies a firefly's attractiveness $\beta$ (determined by its brightness *I*) which in turn is associated with the encoded objective function. As light intensity and thus attractiveness decreases as their distance from the source increases, the variations of light intensity and attractiveness should be monotonically decreasing functions.

- Variation of light intensity: Suppose that there exists a swarm of $n$ fireflies, and $x_i$, $i = 1, 2, \ldots, n$ represents a solution for a firefly $i$ initially positioned randomly in the space, whereas $f(x_i)$ denotes its fitness value. In the simplest form, the light intensity $I(r)$ varies with the distance $r$ monotonically and exponentially. That is determined by Eq. 8.1 (Yang 2008, 2009, 2010b):

$$I = I_0 e^{-\gamma r_{ij}}, \tag{8.1}$$

where $I_0$ is the original light intensity, $\gamma$ is the light absorption coefficient, and$r$ is the distance between firefly $i$ and firefly $j$ at $x_i$ and $x_j$ as Cartesian distance $r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left( x_{i,k} - x_{j,k} \right)^2}$ or the $\ell_2$-norm, where $x_{i,k}$ is the *k*th

component of the spatial coordinate $x_i$ of the *i*th firefly and $d$ is the number of dimensions we have, for $d = 2$, we have $r_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}$.

- Movement toward attractive firefly: A firefly attractiveness is proportional to the light intensity seen by adjacent fireflies (Yang 2008). Each firefly has its distinctive attractiveness $\beta$ which implies how strong it attracts other members of the swarm. However, the attractiveness is relative; it will vary with the distance between two fireflies. The attractiveness function $\beta(r)$ of the firefly is determined via Eq. 8.2 (Yang 2008, 2009, 2010b):

$$\beta = \beta_0 e^{-\gamma r_{ij}^2}, \tag{8.2}$$

where $\beta_0$ is the attractiveness at $r = 0$, and $\gamma$ is the light absorption coefficient which controls the decrease of the light intensity.

The movement of a firefly $i$ at location $x_i$ attracted to another more attractive (brighter) firefly $j$ at location $x_j$ is determined by Eq. 8.3 (Yang 2008, 2009, 2010a):

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + \alpha \varepsilon_i, \tag{8.3}$$

where the first term is the current position of a firefly, the second term is used for considering a firefly's attractiveness to light intensity seen by adjacent fireflies, and the third term is randomization with the vector of random variables $\varepsilon_i$ being drawn from a Gaussian distribution, in case there are not any brighter ones. The coefficient $\alpha$ is a randomization parameter determined by the problem of interest.

- Special cases: From Eq. 8.3, it is easy to see that there exit two limit cases when $\gamma$ is small or large, respectively (Yang 2008, 2009, 2010b). When $\gamma$ tends to zero, the attractiveness and brightness are constant $\beta = \beta_0$ which means the light intensity does not decrease as the distance $r$ between two fireflies increases. Therefore, a firefly can be seen by all other fireflies, a single local or global optimum can be easily reached. This limiting case corresponds to the standard particle swarm optimization algorithm.

On the other hand, when $\gamma$ is very large, then the attractiveness (and thus brightness) decreases dramatically, and all fireflies are short-sighted or equivalently fly in a deep foggy sky. This means that all fireflies move almost randomly, which corresponds to a random search technique.

In general, the FA corresponds to the situation between these two limit cases, and it is thus possible to fine-tune these parameters, so that FA can find the global optima as well as all the local optima simultaneously in a very effective manner. A further advantage of FA is that different fireflies will work almost independently, it is thus particular suitable for parallel implementation. It is even better than genetic algorithm and particle swarm optimization because fireflies aggregate more closely around each optimum. It can be anticipated that the interactions between different sub-regions are minimal in parallel implementation.

Overall, taking into account the basic information described above, the steps of implementing FA can be summarized as follows (Yang 2009; Jones and Boizanté 2011):

- Step 1: Generate initial the population of fireflies placed at random positions within the $n$-dimensional search space.
- Step 2: Initialize the parameters, such as the light absorption coefficient ($\gamma$).

- Step 3: Define the light intensity ($I_i$) of each firefly ($x_i$) as the value of the cost function ($f(x_i)$).
- Step 4: For each firefly ($x_i$), compare its light intensity with the light intensity of every other firefly (i.e., $x_j$).
- Step 5: If ($I_j > I_i$), then move firefly $x_i$ towards $x_j$ in $n$-dimensions.
- Step 6: Calculate the new values of the cost function for each firefly and update the light intensity.
- Step 7: Rank the fireflies and determine the current best.
- Step 8: Repeat Steps 3–7 until the termination criteria is satisfied.

### *8.2.2 Performance of FA*

To test the performance of FA, a set of benchmark functions are adopted in (Yang 2009), namely, Michalewicz function, Rosenbrock function, De Jong function, Schwefel function, Ackley function, Rastrigin function, Easom function, Griewank function, Shubert function, and Yang function. Compared with other computational intelligence (CI) algorithms [such as particle swarm optimization (PSO) and genetic algorithm (GA)], computational results showed that FA is much more efficient in finding the global optima with higher success rates.

## 8.3 Glowworm Swarm Optimization Algorithm

### *8.3.1 Fundamentals of Glowworm Swarm Optimization Algorithm*

Also inspired by luminous insect, the glowworm swarm optimization (GlSO) algorithm was originally proposed by Krishnanand and Ghose (2005) to deal with multimodal problems. Just like ants, elephants, mice, and snakes, glowworms also use some chemical substances, called luciferin, as signals for indirect communication. By sensing luciferin, glowworms can be attracted by strongest luciferin concentrations. In this way, the final optimization results can be found.

Typically, each iteration of the GlSO algorithm consists of two phases, namely, a luciferin-update phase and a movement phase. In addition, for GlSO, there is a dynamic decision range update rule that is used to adjust the glowworms' adaptive neighbourhoods. The details are listed as below (Krishnanand and Ghose 2009):

- Luciferin-update phase: It is the process by which the luciferin quantities are modified. The quantities value can either increase, as glowworms deposit luciferin on the current position, or decrease, due to luciferin decay. The luciferin update rule is given via Eq. 8.4 (Krishnanand and Ghose 2009):

$$l_i(t+1) = (1-\rho) \cdot l_i(t) + \gamma \cdot J \cdot [x_i \cdot (t+1)], \tag{8.4}$$

where $l_i(t)$ denotes the luciferin level associated with the glowworm $i$ at time $t$, $\rho$ is the luciferin decay constant $(0 < \rho < 1)$, $\gamma$ is the luciferin enhancement constant, and $J(x_i(t))$ stands for the value of the objective function at glowworm $i$'s location at time $t$.

- Movement phase: During this phase, glowworm $i$ chooses the next position $j$ to move to using a bias (i.e., probabilistic decision rule) toward good-quality individual which has higher luciferin value than its own. In addition, based on their relative luciferin levels and availability of local information, the swarm of glowworms can be partitioned into subgroups that converge on multiple optima of a given multimodal function. The probability of moving toward a neighbour is given by Eq. 8.5 (Krishnanand and Ghose 2009):

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} [l_k(t) - l_i(t)]}, \tag{8.5}$$

where $j \in N_i(t)$, $N_i(t) = \{j : d_{ij}(t) < r_d^i(t);\ l_i(t) < l_{js}(t)\}$ is the set of neighbours of glowworm $i$ at time $t$, $d_{ij}(t)$ denotes the Euclidean distance between glow worms $i$ and $j$ at time $t$, and $r_d^i(t)$ stands for the variable neighbourhood range associated with glowworm $i$ at time $t$.

Based on Eq. 8.5, the discrete-time model of the glowworm movements can be stated via Eq. 8.6 (Krishnanand and Ghose 2009):

$$x_i(t+1) = x_i(t) + s\left[\frac{x_j(t) - x_i(t)}{\left\|x_j(t) - x_i(t)\right\|}\right], \tag{8.6}$$

where $x_i(t) \in \mathbf{R}^m$ is the location of glowworm $i$ at time $t$ in the $m$-dimensional real space, $\|\cdot\|$ denotes the Euclidean norm operator, and s (>0) is the step size.

- Neighbourhood range update rule: In addition to the luciferin value update rule that is illustrated in the movement phase, in GlSO the glowworms use a radial range [i.e., $\left(0 < r_d^i \le r_s\right)$] update rule to explore an adaptive neighbourhood (i.e., to detect the presence of multiple peaks in a multimodal function landscape). Let $r_0$ be the initial neighbourhood range of each glowworm [i.e., $r_d^i(0) = r_0\ \forall i$], then the updating rule is given via Eq. 8.7 (Krishnanand and Ghose 2009):

$$r_d^i(t+1) = \min\left\{r_s,\ \max\left\{0,\ r_d^i(t) + \beta(n_t - |N_i(t)|)\right\}\right\}, \tag{8.7}$$

where $\beta$ is a constant parameter, and $n_t \in N$ is a parameter used to control the number of neighbours.

Furthermore, in order to escape the dead-lock situation (i.e., all the glowworms converge to suboptimal solutions), Krishnanand and Ghose (2009) employed a local search mechanism.

The working principle is described as follows: during the movement phase, each glowworm moves a distance of step size ($s$) toward a neighbour. Hence, when

$d_{ij}(t) < s$, glowworm $i$ leapfrogs over the position of a neighbour $j$ and becomes a leader to $j$. In the next iteration, glowworm $i$ remains stationary and $j$ overtakes the position of glowworm $i$, thus regaining its leadership. In this way, the GlSO algorithm converges to a state in which all the glowworms construct the optimal solution over and over again.

Typically, by taking into account the basic rules described above, the steps of implementing the GlSO algorithm can be summarized as follows (Krishnanand and Ghose 2009):

- Step 1: Initialize the parameters.
- Step 2: Initiation population of $N$ candidate solution is randomly generated all over the search space.
- Step 3: The fitness function value corresponding to each candidate solution is calculated.
- Step 4: Perform the iteration procedures that include luciferin update phase, movement phase, and decision range update phase.
- Step 5: Check if maximum iteration is reached, go to step 3 for new beginning. If a specified termination criteria is satisfied, stop and return the best solution.

## 8.3.2 Performance of GlSO

To evaluate the performance of the GlSO algorithm, a set of multimodal test functions haven been proposed in Krishnanand and Ghose (2009), such as Peaks function, Rastrigin's function, Circles function, Plateaus function, Equal-peaks-A function, Random-peaks function, Himmelblau's function, Equal-peaks-B function, and Staircase function. Compared with niche particle swarm optimization (NichePSO), the GlSO algorithm presented a better results in terms of the number of peaks captured.

## 8.3.3 Selected GlSO Variants

Although GlSO is a new member of CI family, a number of GlSO variations have been proposed in the literature for the purpose of further improving the performance of GlSO. This section gives an overview to some of these GlSO variants which have been demonstrated to be very efficient and robust.

### 8.3.3.1 Niching GlSO with Mating Behaviour (MNGSO)

As we know, GlSO is developed to solve multimodal function optimization problem which is characterized by the existence of more than one global optimal solution. To increase the search robustness, speed up the convergence, and get more precise solutions, Huang et al. (2011) proposed a new variant of GlSO, called MNGSO, in which a niching strategy and mating behaviour are incorporated.

Generally speaking, niching is a concept developed in the genetic algorithm (GA) community (Angus 2008). Some of the better known niching methods include crowding (Mahfoud 1995), fitness sharing (Goldberg and Richardson 1987), and clearing (Petrowski 1996). Nowadays, Niching strategy has been used extensively in the filed of CI to find multiple solutions at the same time, such as niching for ant colony optimization (ACO) (Angus 2008, 2009), and NichePSO (Engelbrecht 2007).

The basic operating principle of MNGSO is using restricted competition selection (RCS) dynamic niching strategy (Lee et al. 1999), which is a variation of crowding to search several local optimal synchronously. The detail procedures of RCS are as follows (Huang et al. 2011):

- Initialize $N$ subpopulations and mark the best individuals of every subpopulation with $p_{nbest}$.
- When the distance ($d_{ij}$) between $p_{ibest}$ and $p_{jbest}$ (where $p_{ibest}$ and $p_{jbest}$ are best individuals of two different subpopulations) is shorter than $R_{niche}$ (where $R_{niche}$ is the radius of niche), then compare their fitness, set 0 to the lower one and keep the value of the other. The $R_{niche}$ can be updated via Eq. 8.8 (Huang et al. 2011):

$$R_{niche}^{(t+1)} = R_{niche}^{t} - R_{niche}^{t} \times c, \tag{8.8}$$

  where $c$ is a constant used for adjusting the decay rate.
- Randomly initialize the best individuals who are set to 0, and reset its local-decision range $r_d$ to $r_s$. In addition, reselect the best one in its niche, then return to Step 1 until the distance ($d_{ij}$) of any two best individuals respectively belongs to two different niches is lesser than the radius of niche.

In addition, Huang et al. (2011) added a mating behaviour to the MNGSO algorithm in order to get more precise solutions. The formula of updating mate-decision range ($mate\_rs$) is via Eq. 8.9 (Huang et al. 2011):

$$mate\_rs = (1 - constrap)mate\_rs, \tag{8.9}$$

where $constrap$ denotes the contractibility rate.

The steps of implementing the MNGSO algorithm can be summarized as follows (Huang et al. 2011):

- Step 1: Initialize the parameters.
- Step 2: Update luciferin of all the glowworm.
- Step 3: Calculate the neighbours of each glowworm.
- Step 4: Select $j(j \in N_i(t))$ as the movement direction of glowworm $i$ by roulette, and update the position of $i$.
- Step 5: Implement the RCS niching strategy, determine the best individuals of every niching subgroups.

- Step 6: Implement mating behaviour to the best individual of each niche.
- Step 7: When the predetermined iterations for eliminating reached, the worst niching subgroup is eliminated and updated.
- Step 8: Check if maximum iteration is reached, go to step 2 for new beginning. If a specified termination criteria is satisfied stop and return the best solution.

#### 8.3.3.2 Performance of MNGSO

To verify the availability and feasibility of MNGSO, a set of standard functions are tested in Huang et al. (2011). Compared with PSO, PSO with chaos (CPSO), artificial fish swarm algorithm (AFSA), and AFSA with chaos (CAFSA), the experimental results showed that MNGSO is an effective global algorithm for finding optimal results.

### *8.3.4 Representative GlSO Applications*

The applications of GlSO can be found in many areas, in this section, wireless sensor networks (WSNs) is selected as an example and summarized in the following section. Recently, WSNs are becoming a rapidly developing area in both research community and civilian applications, such as target acquisition, forest fire prevention, structural health measurement, and surveillance. In general, a WSN includes a large number of small wireless devices (i.e., sensor nodes) in which each one has high precision to acquire some physical data (Benini et al. 2006). Among others, one of the key features in a WSN is the coverage issue including energy saving (Anastasi et al. 2009), connectivity (Raghavan and Kumara 2007), and deployment of wireless sensor nodes (Pradhan and Panda 2012).

#### 8.3.4.1 Sensor Deployment Approach Using GlSO

To ensure that the area of targets of interest can be covered, an optimized sensor deployment scheme is an essential guide for anyone interested in wireless communications. Recently, Liao et al. (2011) proposed a GlSO-based deployment approach to enhance the coverage after an initial random placement of sensors. In details, each sensor node is mimicked as a glowworm and emitted by luciferin. The intensity of luciferin is based on the distance between a sensor and its neighbours. By using the probabilistic mechanism, each sensor node selects its neighbours which has lower intensity of lucifein and decides to move towards one of them. In this way, the coverage of sparsely covered areas can be minimized.

To validate the performance of the GlSO algorithm, a comparison with the virtual force algorithm (VFA) has been illustrated. Computational results showed that the GlSO algorithm can improve the coverage rate with limited senor movement.

## 8.4 Emerging Luminous Insect Inspired Algorithms

In addition to the aforementioned FA and GlSO algorithms, the characteristics of this interesting insect also motivate researchers to develop another luminous insect inspired innovative CI algorithm.

### *8.4.1 Fundamentals of Bioluminescent Swarm Optimization Algorithm*

Bioluminescent swarm optimization (BiSO) algorithm was proposed by Oliveira et al. (2011). Although BiSO can be loosely regarded as a hybridization of PSO and GlSO, several characteristics have made it unique. For example, apart from the basic characteristics of GlSO (such as luciferin update rule and stochastic neighbour movement rule), Oliveira et al. (2011) proposed a set of new features, namely stochastic adaptive step sizing, global optimum attraction, leader movement, and mass extinction. In addition, the BiSO algorithm is incorporated with two local search techniques, i.e., local unimodal sampling (LUS) and single-dimension perturbation search (SDPS). The following subsections give us a detailed description about some of these unique features.

#### 8.4.1.1 Luciferin-Update Phase

Instead of using fitness-based function $(J(x_i(t)))$ to evaluate the luciferin value between the glowworms as proposed by the GlSO, BiSO uses luciferin-based attraction which is controlled by luciferin decay constant $(\rho)$ and the luciferin enhancement constant $(\gamma)$, respectively. The luciferin update rule is given by Eq. 8.10 (Oliveira et al. 2011):

$$l_i(t+1) = (1-\rho) \cdot l_i(t) + \gamma \cdot f(x_i(t)), \tag{8.10}$$

where $l_i(t)$ denotes the luciferin level associated with the glowworm $i$ at time $t$, $\rho$ is the luciferin decay constant $(0 < \rho < 1)$, $\gamma$ is the luciferin enhancement constant, and $f(x_i(t))$ stands for the value of the objective function at glowworm $i$'s location at time $t$.

#### 8.4.1.2 Stochastic Adaptive Step Sizing

- In BiSO, the following equation is employed to calculated the next location of a given artificial luminous insect via Eq. 8.11 (Oliveira et al. 2011).

$$x_i(t+1) = x_i(t) + rand \cdot s \cdot \left[\frac{x_j(t) - x_i(t)}{\left\|x_j(t) - x_i(t)\right\|}\right] + c_g \cdot rand \cdot s \cdot \left[\frac{g(t) - x_i(t)}{\|g(t) - x_i(t)\|}\right], \tag{8.11}$$

where the artificial luminous insect's current position is denoted by $x_i(t)$, *rand* represents a random number which falls within $[0, 1]$, the artificial luminous insect's current step size is indicated by $s$, $c_g$ is a constant which is used to express the global best attraction, and $g(t)$ stands for the global best location.

- In GlSO, a fixed step size is normally used, whereas BiSO alters the step size in a random manner which is similar to PSO. Apart from this, the maximum step in BiSO is adaptive governed by Eq. 8.12 (Oliveira et al. 2011):

$$s = s_0 \cdot \frac{1}{1 + c_s \cdot l_i(t)}, \tag{8.12}$$

where $s_0$ stands for the maximum step, $l_i(t)$ denotes the amount of luciferin of an artificial luminous insect, and $c_s$ represents a slowing constant.

#### 8.4.1.3 Global Optimum Attraction

Like PSO, BiSO employed a global optimum factor ($c_g$) to enhance the neighbour selection. In other words, the selecting of next location is governed by two factors: the current step size and an attractive force. By using a combination of these two factors, every node tries to maximize its value while maintaining the required number of neighbours.

#### 8.4.1.4 Mass Extinction

To prevent early stagnation, Oliveira et al. (2011) proposed a mechanism called mass extinction to counteract this effect. It works by reinitializing all or part of the particles, but keeping the best-so-far value (i.e., global optima). That means, in BiSO, the Luciferin value is reinitialized each time when the system approaches stagnation or no improved solution has been generated for a certain number of iterations, except the global best location ($g(t)$). The parameter $eT$ is used to control this procedure.

#### 8.4.1.5 Local Search Procedures

Local search is usually used to find high-quality solutions to combinatorial optimization problems in reasonable time. In BiSO, Oliveira et al. (2011) applied two local search method, i.e., LUS and SDPS. The former one is embedded at each iteration meaning the default movement for the best particle, called weak one, while the latter one is embedded at each $lR$ iterations for searching an improved solution within the neighbourhood of the current solution, called strong one.

The steps of implementing the BiSO algorithm can be summarized as follows (Oliveira et al. 2011):

- Step 1: Initialize the parameters.
- Step 2: Randomly generate the bioluminescent particle population.
- Step 3: Perform the iteration procedures that include luciferin update phase, movement phase, step size update phase, and local search phase.
- Step 4: Check if maximum iteration is reached, go to Step 3 for new beginning. If a specified termination criteria is satisfied stop and return the best solution.

### 8.4.2 Performance of BiSO

The BiSO algorithm has been tested by four well-known benchmark functions, namely, Rastrigin function, Griewank function, Schaffer function, and Rosenbrock function in (Oliveira et al. 2011). Compared with PSO, the BiSO algorithm presented a better results of finding the global best solution.

## 8.5 Conclusions

In this chapter, three CI methods are introduced, namely, FA, GlSO algorithm, and BiSO algorithm. The general idea behind those algorithms is similar, such as all algorithms are inspired by the luminous insects, and the updating rule is proportional to the higher value of objective function. However, the actual procedures is very different. For example, FA is proposed as a general optimization algorithm, GlSO algorithm is designed to capture multiple peaks in mulitmodal functions (i.e., without the aim of finding the global best), and BiSO can be loosely regarded as a hybridization of PSO algorithm and GlSO algorithm. The main difference between GlSO algorithm and BiSO algorithm lies in the finding of global optimum. Although FA, GlSO algorithm, and BiSO algorithm are newly introduced CI methods, we have witnessed the following rapid spreading of these luminous insect inspired algorithms:

First, in addition to the selected variants detailed in this chapter, several enhanced version of FA and GlSO algorithm can also be found in the literature are outlined below:

- Chaos enhanced FA (Yang 2011).
- Discrete FA (Sayadi et al. 2010).
- Enhanced FA (Niknam et al. 2012).
- Lévy-flight FA (Yang 2010a).
- Multiobjective FA (Yang 2013).

- Definite updating search domains based GlSO (Liu et al. 2011).
- Hierarchical multi-subgroups based GlSO (He et al. 2013).
- Hybrid GlSO (Zhou et al. 2013; Gong et al. 2011).
- Improved GlSO (Wu et al. 2012; He and Zhu 2011).
- Local search based GlSO (Zhao et al. 2012b).
- MapReduce based GlSO (Aljarah and Ludwig 2013a).
- Metropolis criterion based GlSO (Zhao et al. 2012a).
- Modified GlSO (Oramus 2010; Zhang et al. 2011).

Second, the FA has also been successfully applied to a variety of optimization problems as listed below:

- Artificial neural network training (Horng et al. 2012).
- Continuous constrained optimization (Łukasik and Żak 2009).
- Data clustering (Senthilnath et al. 2011a).
- Image processing (Horng and Liou 2011; Horng 2012).
- Linear array antenna design optimizaiton (Basu and Mahanti 2011).
- Multimodal optimization (Yang 2009).
- Multivariable proportional-integral-derivative control (Coelho and Mariani 2012).
- Power system (Apostolopoulos and Vlachos 2011; Niknam et al. 2012; Yang et al. 2012).
- Scheuling optimization (Sayadi et al. 2010).
- Sematic Web service composition optimization (Pop et al. 2011a, 2011b).
- Stock market price forecasting (Kazem et al. 2013).
- Structure design optimization (Gomes 2011; Gandomi et al. 2011; Talatahari et al. 2012; Miguel and Miguel 2012).
- Structure design optimization (Talatahari et al. 2012).

Third, apart from the representative GlSO applications, it has also been successfully applied to a variety of optimization problems as arrayed below:

- Data clustering (Aljarah and Ludwig 2013b; Huang and Zhou 2011; Tseng 2008).
- Image processing (Senthilnath et al. 2011b).
- Injection mould water channel location optimization (Chiang 2012).
- Multi-dimensional knapsack problem (Gong et al. 2011).
- Robotics control (Krishnanand and Ghose 2005; Krishnanand et al. 2006).
- Wireless sensor networks (Krishnanand and Ghose 2005).

Interested readers please refer to them together with several excellent reviews [e.g., (Fister et al. 2013)] as a starting point for a further exploration and exploitation of luminous insect inspired algorithms.

## References


Aljarah, I., Ludwig, S. A. (2013a, April 15–19). A MapReduce based glowworm swarm optimization approach for multimodal functions. In *IEEE Symposium Series on Computational Intelligence (SSCI 2013)*, Singapore (pp. 22–31). *IEEE*.

Aljarah, I., Ludwig, S. A. (2013b, June 20–23) A new clustering approach based on glowworm swarm optimization. In *IEEE Congress on Evolutionary Computation*, Cancún, México (pp. 2642–2649). IEEE.

Anastasi, G., Conti, M., Francesco, M. D., & Passarella, A. (2009). Energy conservation in wireless sensor networks: a survey. *Ad Hoc Networks, 7*, 537–568.

Angus, D. (2009). Niching for ant colony optimisation. In A. Lewis (Ed.), *Biologically-inspired optimisation methods, SCI 210.* Berlin Heidelberg: Springer.

Angus, D. J. (2008). *Niching ant colony optimisation*. Doctor of Philosophy, Swinburne University of Technology.

Apostolopoulos, T., & Vlachos, A. (2011). Application of the firefly algorithm for solving the economic emissions load dispatch problem. *International Journal of Combinatorics, 523806*, 1–23.

Babu, B. G., & Kannan, M. (2002). Lightning bugs. *Resonance, 7*, 49–55.

Basu, B., & Mahanti, G. K. (2011). Fire fly and artificial bees colony algorithm for synthesis of scanned and broad-side linear array antenna. *Progress In Electromagnetics Research B, 32*, 169–190.

Benini, L., Farella, E., & Guiducci, C. (2006). Wireless sensor networks: enabling technology for ambient intelligence. *Microelectronics Journal, 37*, 1639–1649.

Buck, J., & Case, J. (2002). Physiological links in firefly flash code evolution. *Journal of Insect Behavior, 15*, 51–68.

Chiang, Y.-S. (2012). *Water channel location optimization of injection molding using glowworm swarm algorithm with variable step.* Unpublished Master Thesis (in Chinese), Tatung University.

Coelho, L. D. S., & Mariani, V. C. (2012). Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Computers and Mathematics with Applications, 64*, 2371–2382.

Engelbrecht, A. P. (2007). *Computational intelligence: An introduction,* West Sussex, England: Wiley, ISBN 978-0-470-03561-0.

Fister, I., Jr Fister, I., Yang, X.-S. & Brest, J. (2013). A comprehensive review of firefly algorithm. *Swarm and Evolutionary Computation*. http://dx.doi.org/10.1016/j.swevo.2013.06.001i

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers and Structures, 89*, 2325–2336.

Goldberg, D. E. & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *2nd International Conference on Genetic Algorithm* (pp. 41–49).

Gomes, H. M. (2011). A firely metaheuristic algorithm for structural size and shape optimization with dynamic constraints. *Mecánica Computacional,* 30, 2059–2074.

Gong, Q., Zhou, Y., & Luo, Q. (2011). Hybrid artificial glowworm swarm optimization algorithm for solving multi-dimensional knapsack problem. *Procedia Engineering, 15*, 2880–2884.

He, D.-X., & Zhu, H.-Z. (2011). An improved glowworm swarm optimization algorithm for high-dimensional function optimization. *Energy Procedia, 13*, 5657–5664.

He, L., Tong, X., & Huang, S. (2013). Glowworm swarm optimization algorithm based on hierarchical multi-subgroups. *Journal of Information and Computational Science, 10*, 1245–1251.

Horng, M.-H. (2012). Vector quantization using the firefly algorithm for image compression. *Expert Systems with Applications, 39*, 1078–1091.

Horng, M.-H., Lee, Y.-X., Lee, M.-C. & Liou, R.-J. (2012). Firefly meta-heuristic algorithm for training the radia basis function network for data classification and disease diagnosis. In: R.

Parpinelli (Ed.), *Theory and new applications of swarm intelligence,* Chap. 7 (pp. 115–132). Rijeka, Croatia: In-Tech. ISBN 978-953-51-0364-6.

Horng, M.-H., & Liou, R.-J. (2011). Multilevel minimum cross entropy threshold selection based on the firefly algorithm. *Expert Systems with Applications, 38*, 14805–14811.

Huang, K., Zhou, Y., & Wang, Y. (2011). Niching glowworm swarm optimization algorithm with mating behavior. *Journal of Information and Computational Science, 8*, 4175–4184.

Huang, Z., & Zhou, Y. (2011). Using glowworm swarm optimization algorithm for clustering analysis. *Journal of Convergence Information Technology, 6*, 78–85.

Jones, K. O., & Boizanté, G. (2011, June 16–17). Comparison of firefly algorithm optimisation, particle swarm optimisation and differential evolution. *International Conference on Computer Systems and Technologies (CompSysTech)*, (pp. 191–197). Vienna, Austria.

Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M. & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing, 13*, 947–958. http://dx.doi.org/10.1016/j.asoc.2012.09.024.

Krishnanand, K. N., Amruth, P., Guruprasad, M. H., Bidargaddi, S. V. & Ghose, D. (2006, May). Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources. In *IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, USA, (pp. 958–963). IEEE.

Krishnanand, K. N. & Ghose, D. (2005). Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In *IEEE Swarm Intelligence Symposium (SIS)* (pp. 84–91). *IEEE.*

Krishnanand, K. N., & Ghose, D. (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intelligence, 3*, 87–124.

Lee, C. G., Cho, D. H., & Jung, H. K. (1999). Niche genetic algorithm with restricted competition selection for multimodal function optimization. *IEEE transaction on Magnetics, 35*, 1122–1125.

Liao, W.-H., Kao, Y., & Li, Y.-S. (2011). A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. *Expert Systems with Applications, 38*, 12180–12188.

Liu, J., Zhou, Y., Huang, K., Ouyang, Z., & Wang, Y. (2011). A glowworm swarm optimization algorithm based on definite updating search domains. *Journal of Computational Information Systems, 7*, 3698–3705.

Łukasik, S., & Żak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. *Computational collective intelligence. semantic web, social networks and multiagent systems LNCS 5796,* (pp. 97–106). Berlin: Spinger.

Mahfoud, S. W. (1995). *Niching methods for genetic algorithms.* Doctor of Philosophy, University of Illinois.

Miguel, L. F. F., & Miguel, L. F. F. (2012). Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. *Expert Systems with Applications, 39*, 9458–9467.

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

Oliveira, D. R. D., Parpinelli, R. S. & Lopes, H. S. (2011). Bioluminescent swarm optimization algorithm. *Evolutionary Algorithms,* Chap. 5 (pp. 71–84). Eisuke Kita: InTech.

Oramus, P. (2010). Improvements to glowworm swarm optimization algorithm. *Computer Science, 11*, 7–20.

Petrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In *IEEE International Conference on Evolutionary Computation*, (pp. 798–803).

Pop, C. B., Chifu, V. R., Salomie, I., Baico, R. B., Dinsoreanu, M., & Copil, G. (2011a). A hybrid firefly-inspired approach for optimal semantic Web service composition. *Scalable Computing Practice and Experience, 12*, 363–369.

Pop, C. B., Chifu, V. R., Salomie, I., Baico, R. B., Dinsoreanu, M. & Copil, G. (2011b, 19–21 September). A hybrid firefly-inspired approach for optimal semantic Web service composition. *3rd Workshop on Software Services: Semantic-based software services*, Szczecin, Poland, (pp. 1–6).

Pradhan, P. M., & Panda, G. (2012). Connectivity constrained wireless sensor deployment using multi objective evolutionary algorithms and fuzzy decision making. *Ad Hoc Networks, 10*, 1134–1145.

Raghavan, U. N., & Kumara, S. R. T. (2007). Decentralised topology control algorithms for connectivity of distributed wireless sensor networks. *International Journal of Sensor Networks, 2*, 201–210.

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

Senthilnath, J., Omkar, S. N., & Mani, V. (2011a). Clustering using firefly algorithm: Performance study. *Swarm and Evolutionary Computation, 1*, 164–171.

Senthilnath, J., Omkar, S. N., Mani, V., Tejovanth, N., Diwakar, P. G., & Archana, S. B. (2011b). Multi-spectral satellite image classification using glowworm swarm optimization. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 47–50). IEEE.

Talatahari, S., Gandomi, A. H. & Yun, G. J. (2012). Optimum design of tower structures using firefly algorithm. *The Structural Design of Tall and Special Buildings.* (DOI:10.1002/tal.1043).

Tseng, K.-T. (2008). *A glowworm algorithm for solving data clustering problems (in Chinese).* Unpublished Master Thesis, Tatung University.

Wu, B., Qian, C., Ni, W., & Fan, S. (2012). The improvement of glowworm swarm optimization for continuous optimization problems. *Expert Systems with Applications, 39*, 6335–6342.

Yang, X.-S. (2008). *Nature-inspired metaheuristic algorithms*. UK: Luniver Press. ISBN 978-1-905986-28-6.

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In O. Watanabe, & T. Zeugmann, (Eds.), *SAGA 2009, LNCS 5792, (*pp. 169–178). Berlin Heidelberg: Springer.

Yang, X.-S. (2010a). Firefly algorithm, Lévy flights and global optimization. In M. Bramer, (Ed.) *Research and development in intelligent systems.* 26, 209–218. London, UK: Springer-Verlag.

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

Yang, X.-S. (2011). Chaos-enhanced firefly algorithm with automatic parameter tuning. *International Journal of Swarm Intelligence Research, 2*, 1–11.

Yang, X.-S. (2013). Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers, 29*, 175–184. (DOI 10.1007/s00366-012-0254-1).

Yang, X.-S., Hosseini, S. S. S., & Gandomi, A. H. (2012). Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Applied Soft Computing, 12*, 1180–1186.

Zhang, Y.-L., Ma, X.-P., Gu, Y., & Miao, Y.-Z. (2011) A modified glowworm swarm optimization for multimodal functions. In *Chinese Control and Decision Conference (CCDC),* (pp. 2070–2075). IEEE.

Zhao, G., Zhou, Y., Luo, Q., & Wang, Y. (2012a). A glowworm swarm optimization algorithm based on metropolis criterion. *International Journal of Advancements in Computing Technology*, *4*, 149–155.

Zhao, G., Zhou, Y., & Wang, Y. (2012b). The glowworm swarm optimization algorithm with local search operator. *Journal of Information & Computational Science*, *9*, 1299–1308.

Zhou, Y., Zhou, G., & Zhang, J. (2013). A hybrid glowworm swarm optimization algorithm for constrained engineering design problems. *Applied Mathematics and Information Sciences, 7*, 379–388.