International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444

International Journal of Engineering and Technology Volume 2 No. 5, May, 2012

# An Improved Fuzzy Based Glowworm Algorithm

**Behnam Iranpour[1], Mohammadreza Meybodi[2]**
[1]Department of Electronic, Computer and IT Islamic Azad University Qazvin, Iran
[2]Department of Computer engineering and IT Amirkabir University of Technology Tehran, Iran

## ABSTRACT

Glowworm swarm optimization algorithm is one of the most novel algorithms based on swarm intelligence and inspired from light emission behavior of glowworms to attract a peer or prey in nature. The main application of this algorithm is to occupy all local optima of multi modal functions. This algorithm has shown some such weaknesses in global search as low convergence rate, not so much accuracy in computations and time consuming computations. In this paper, an innovative glowworm algorithm via fuzzy logic concepts is represented which removes disadvantages of all standard glowworm algorithms to a high extent. Experimental results of well known benchmark functions show a proper efficiency of the proposed algorithm in contrast to other similar algorithms.

**Keywords:** *glowworm algorithm, fuzzy logic, Gaussian membership function, multi modal functions, optimization.*

## 1. INTRODUCTION

There are multiple optimization problems for which finding an optimal solution is so difficult because of the number of local optima. This kind of optimization problems is called multi modal optimization problems**.** To solve these problems, various algorithms, like genetic algorithm [1] and particle swarm optimization algorithms [2] were presented. However, these approaches have many flaws for optimizing these problems.

Glowworm swarm optimization (GSO) algorithm was proposed by an Indian researcher, Krishnanand, for the first time in 2005 [3], [4]. In this method, glowworms light emission property was modeled which provided them with peer or prey attraction capability. Light production of glowworms is done by a chemical named Luciferin.

Glowworm algorithm is of a great ability in solving problems, such as finding some local optima of multi modal functions simultaneously [5], [6] searching spaces of higher dimensions [7] and orienting multiple sources [3]. Nevertheless, this algorithm also like other algorithms has some flaws, such as weakness in local search, low convergence rate, low computational accuracy and time consuming computations [8], [9].

For this is an innovative algorithm, so far, limited works have been done to enhance it. The idea of using multi swarm glowworm, for the first time, was proposed by Dengxux He and Hazeng Zhu to decrease computation time and increase computation accuracy [8]. Jun Li Zhang enhanced local search by imposing chaos method on glowworm algorithm [9].

In this paper, by utilizing fuzzy logic, a new algorithm called Fuzzy Glowworm Swarm optimization (FGSO) algorithm is presented. The proposed algorithm causes increase in convergence rate of standard glowworm algorithm by enhancing algorithm's local search. Experimental results on 9 different standard functions show high efficacy of the proposed algorithm.

Fuzzy logic has a multi value viewpoint on events that is unlike Boolean logic in which everything is true or false. Indeed, fuzzy logic is the extended form of Boolean logic to apply ambiguous concepts. To state ambiguity as a number, fuzzy logic introduces a membership function in a set that maps each element to a real number between zero and one. This number shows the element's membership degree in the set.

The following of the paper is organized as below; section 2 introduces glowworm algorithm. Fuzzy logic is described briefly in section 3. The proposed algorithm is introduced in section 4. In section 5, simulation results are presented and the final section discusses the conclusion.

## 2. GlOWWORM ALGORITHM

Glowworm algorithm is started by placing randomly a set of n glowworms in different points of optimization problem's search space. At first, all glowwroms have the same amount of Luciferin, $l_0$. Each iteration of the algorithm consists of Luciferin updating phase and an glowworms' positions updating phase [3], [4].

**Luciferin updating phase:** Luciferin amount of each glowworm is determined due to fitness value of that worm's position at each of iterations. In this way that, according to fitness value and in proportion to that value, an amount of Luciferin is increased to the current glowworm. In order to model gradual drop of Luciferin by passing time, an amount of current Luciferin with a coefficient less than one is reduced from it. Hence, Luciferin updating equation is given as:

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J\big(x_i(t)\big), \qquad (1)$$

where, $l_i(t)$ is new amount of Luciferin, $l_i(t-1)$ is former amount of Luciferin and $J(x_i(t))$ is position fitness of glowworm i in iteration t of the algorithm and $\rho$ and $\gamma$ are constants for modeling Luciferin gradual drop and the effect of fitness on Luciferin.

**Glowworms moving phase:** during moving step, each glowworm moves probabilisticly toward one of its neighbors with higher Luciferin. Thereafter, glowworms move toward neighbors with more luminosity. Decision radius is determining an area where glowworms of interior zone are accounted for neighbors. Sensory radius $r_s$ is determinant for upper limit of decision eradius. In fact, during GSO algorithm iterations, decision radii change due to conditions in which glowworms are, but decision radius of each glowworm would not be more than its sensory radius. Sensory radius models the maximum capability of glowworms for observing other glowworms. In Fig. 1, decision radius and sensory radius concepts are seen.

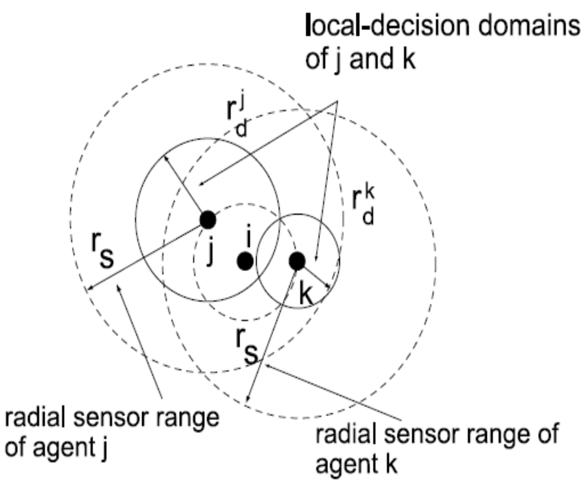


**Fig. 1. decision domain and sensory domain of glowworm [6]**

For every glowworm i, probability of moving toward more radiant neighbor j is defined as:

$$P_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \epsilon N_i(t)} l_k(t) - l_i(t)}, \qquad (2)$$

where, $N_i(t)$ is defined as :

$$j \epsilon\, N_i(t), N_i(t) = \{j : d_{ij} < r_d^i(t); l_i(t) < l_j(t)\}, \qquad (3)$$

where, in this equation j is the set of neighboring glowworms of glowworm i at instant t, $d_{ij}(t)$ is Euclidean distance between glowworm i and j at instant t and $r_d{}^i(t)$ is neighborhood domain of related variable of glowworm i at instant t. By assuming selecting glowworm j by glowworm i with probability P obtained from (2), discrete equation of motion- time can be written as:

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}\right) \qquad (4)$$

where, $x_i(t)$ is an m-dimension vector of glowworm i's position at instant t, ||.|| is Euclidean norm operator and s is moving step length.

**Updating neighborhood domain:** a neighborhood is assigned to each glowworm i which its radial domain $r_d^i$ is obviously dynamic $(0<r_d^i<r_s)$. The reason of not using a fixed neighborhood should be justified. Since glowworms only require local information in their neighborhood for movement, so the number of recognizable modals is a function of sensory domain radius. Indeed, if sensory domain of each glowworm covers whole search space, all glowworms move toward global optimum and local optima are disregarded. Therefore, GSO utilizes an adaptive neighborhood domain to recognize the existence of multiple modals in optimizing problems of multi modal functions.

With assuming $r_0$ as an initial neighborhood domain for each glowworm $(r_d^i(0) = r_0)$, at each of iterations of GSO algorithm, neighborhood domain of each glowworm is updated with respect to:

$$r_d^i(t+1) = min\left\{r_s, max\{0, r_d^i(t) + \beta(n_t - Nit, \right. \qquad (5)$$

where, $\beta$ is a constant parameter and $n_t$ is a parameter to control the number of neighborhoods. Other than two parameters $n$ and $r_s$ showing the number of and sensory radius of glowworms all glowworm algorithm's parameters are constant and their values for all experiments are specified according to table (1).

**Table 1: Parameters Values of Glowworm Algorithm that are Kept Constant for All Experiments [3]**

| $P$ | $\gamma$ | $\beta$ | $n_t$ | $S$ | $L_0$ |
|---|---|---|---|---|---|
| 0.4 | 0.6 | 0.08 | 5 | 0.03 | 5 |

## 3. FUZZY LOGIC

Fuzzy logic foundation is based on fuzzy sets theory. This theory is a generalization of classic sets theory in mathematics. In classic sets theory, an element belongs to the set or doesn't. As a matter of fact, membership of elements is following a two value pattern. But, fuzzy sets theory extends this concept and applies membership degree. So, an element can be a member of to the set –not completely- to some extent. Fuzzy set is defined as:

$$\{(x, \mu_A(x)) \mid x \epsilon X\} \qquad (6)$$

where, the membership of set's members is determined by membership function $\mu(x)$ that x is a representative of a specific element and $\mu$ is a fuzzy function which determines the membership degree of x in corresponding set and it takes a value between zero and one.

In other words, $\mu(x)$ is a map from values x to possible numbers between zero and one. $\mu(x)$ may be a set of discrete or continuous values. The properties expressed to determine fuzzy set's members are fuzzy and are not precise. Thereafter, it is possible to use different membership functions to show a fuzzy set. Practically, those functions are used which have a simple mathematical representation and are adjustable by a limited number of parameters. Nonetheless, membership functions are categorized into point, linear and nonlinear functions. General form of linear one is inspired from polygonal shapes, like trapezoidal membership function, and general form of nonlinear case is due to bell shapes, like Gaussian membership function. In Fig. 2, 3 an example of theses membership functions is shown.

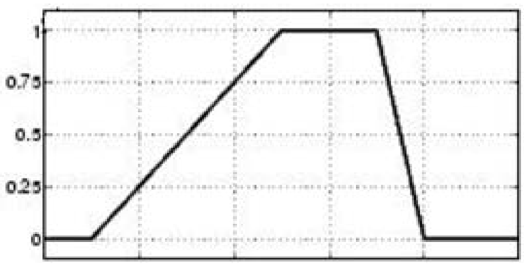


**Fig. 2: Various Membership Functions**

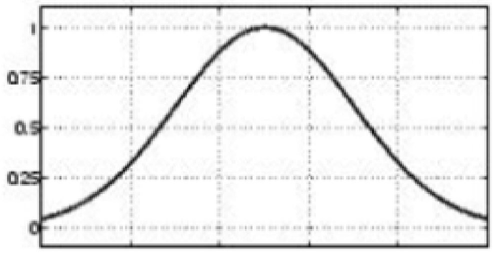


**Fig. 3: Gaussian Membership Function**

Fuzzy logic is applied extensively in solving various problems. Many researchers use fuzzy logic to improve optimization algorithms efficiency [10], [11].

## 4. PROPOSED ALGORITHM

In standard GSO algorithm, glowworms have to recognize their neighbors before moving step. In this algorithm, a glowworm which wants to change position toward other glowworms is considered as reference glowworm. Then, other glowworms which their Euclidean distance from this reference glowworm is less than reference glowworm's decision domain are considered as reference glowworm's neighbors. This approach has some obstacles. If a glowworm is just a little further than the effect domain of reference glowworm, it will not be in neighbors set. Also, with this point of view, simply, it is possible that no other glowworm be in a neighborhood of a glowworm and in moving step, the position of the glowworm with no neighbors doesn't change. It causes glowworms not to converge to a global optimum with a proper rate and to be far from each other, and even in some cases, some glowworms might not move until the end of algorithm iterations. According to mentioned cases, in order to take into account the effect of more glowworms on each glowworm, the degree of neighborhood of glowworms to each other is determined by a fuzzy membership function. So neighborhood degree is replaced with former concept of crisp neighbors set.

In the innovative proposed approach, membership degree of glowworm j in the neighborhood of glowworm i is an output of a Gaussian function with the input of Euclidean distance between glowworms i and j. center of function is its symmetry axis and determines the location of function's modal.

Gaussian membership function is defined as:

$$\mu(\vec{x}) = \exp\left(-\left(\frac{x-m}{\sigma}\right)^2\right), \qquad (7)$$

where, m is center and $\sigma$ is its standard deviation. In improved GSO algorithm by fuzzy logic called fuzzy glowworm swarm optimization (FGSO), for every glowworm i, membership degree of other glowworms, j, is calculated in the neighborhood of i and glowworms with membership degrees less than a threshold are eliminated. In next step, probabilities computed due to the differences of glowworms' Luciferin is multiplied by obtained membership degrees. Hence, by modifying these probabilities as (8), in addition to the amount of difference of glowworms' Luciferin, their distance is also effective in the probability of selecting them to move toward each other.

$$p_{ij} \rightarrow \mu_{ij} p_{ij} \qquad (8)$$

This is more natural than current situation in GSO algorithm for in reality also more further are glowworms, feel less difference of Luciferin and closer glowworms with lesser Luciferin might have more effect than glowworms with more Luciferin but in further distance. Fig. 4 shows pseudo code of FGSO algorithm.

Fuzzy glowworm swarm optimizatio (FGSO)algorithm

Set number of dimension = *m*
Set number of glowworm = *n*
Let *s* be the step size
Let $x_i(t)$ be the location of glowworm *i* at time *t*
*deploy_agent_randomly*,
For i=1 to *n* do $l_i(0) = l_0$

set maximum iteration number = *iter_max*;
set $t = 1$;
while ( $t \leq iter\_max$) do:
{
for each glowworm *i* do : % Luciferin-update phase
$l_i(t) = (1-\rho) l_i(t-1) + \gamma J(x_i(t))$,
for each glowworm *i* do : %Movement –phase
{
$N_i(t) = \{j : \|x_j(t) - x_i(t)\| < r_d^i(t); l_i(t) l_j(t)\}$,
Where $\|\vec{x}\|$ is the norm of $\vec{x}$
$\mu(\vec{x}) = \exp\left(-\left(\frac{x-m}{\sigma}\right)^2\right)$,
for each glowworm $j \in N_i(t)$ do:
$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}$,
$p_{ij} \rightarrow \mu_{ij} p_{ij}$
*j=select_glowworm* ($\vec{p}$)
$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}\right)$
$r_d^i(t+1) = min\{r_s, max\{0, r_d^i(t) +$
$\beta nt - Nit$
}
$t \leftarrow t + 1$;
}

**Fig. 4: FGSO Algorithm's Pseudo Code**

This increases local search property and using fuzzy neighborhood increases convergence rate algorithm speed generally.

## 5. EXPERIMENTAL RESULTS

To study this algorithm, optimization problem of Rastrigin function as the sample problem is considered to compare proposed improved algorithm and standard glowworm swarm optimization algorithm. Rastrigin function has multiple modals and close to each other. Close modals increase the possibility of being tangled in local traps. The threshold of eliminating from neighborhood is considered 0.3 and Gaussian function's variance is considered equal to effect domain of each glowworm at each time. It is clear that the center of Gaussian function is also zero. To show the advantage of FGSO to GSO, comparison of convergence graphs, that is the changes of the best fitness in different iterations is more useful. For this reason, convergence graphs of both algorithms are represented for comparison in Fig. 5 and 6.

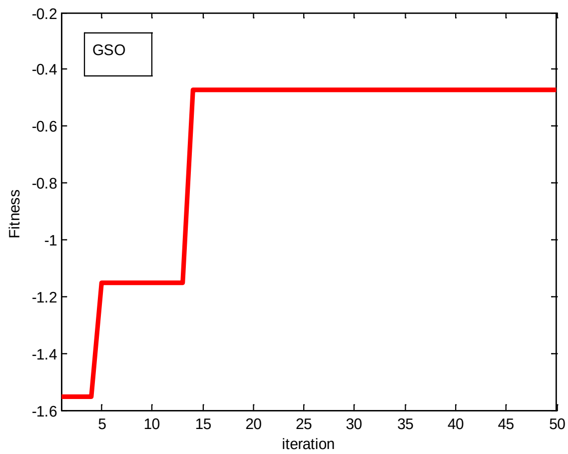


**Fig. 5: Convergence Graph of GSO Algorithm**

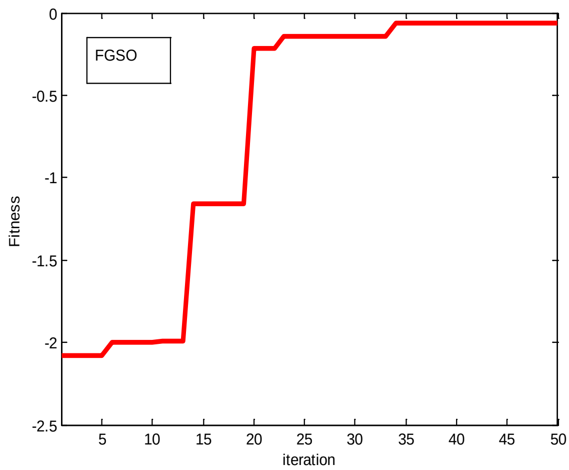


**Fig. 6: Convergence Graph of FGSO Algorithm**

Comparison of two graphs shows that FGSO's convergence graph illustrates more changes which is expressing better search.

In next step, algorithm efficiency was studied on various functions. For each of these functions, GSO and FGSO algorithms were performed 20 times with different conditions and parameters and final result, that is the difference between obtained optimal amount and real optimal value was computed averagely to eliminate the effect of selecting parameters and the difference resulted from randomness of algorithms. Final result of above analysis along with GSO's error average is presented in table (2).

**Table 2: The Difference Between Real Optimal Value and Obtained Optimal Value by GSO and FGSO Algorithms**

| Function | Error Average GSO (standard deviation) | Error Average FGSO (standard deviation) |
|---|---|---|
| Ackley | 0.1498(0.021) | *0.0849(0.011)* |
| Rastrigin | 0.1537(0.036) | *0.0558(0.003)* |
| Penholder | 0.0627(0.023) | *0.0575(0.065)* |
| Himmelbau | 0.0959(0.004) | *0.0521(0.006)* |
| Crossintray | 0.1904(0.006) | *0.0560(0.001)* |
| Bukin4 | 0.1603(0.001) | *0.0774(0.002)* |
| Beale | 0.2060(0.003) | *0.1112(0.011)* |
| Holdertable | 0.1393(0.005) | *0.0587(0.009)* |
| Griewank | 0.2157(0.018) | *0.0638(0.012)* |

Experimental results on aforementioned functions in table (2) show that fuzzy improved glowworm algorithm, in all being studied functions, its error average in finding real optimum is less than error average of standard glowworm algorithm. Therefore, applying neighborhood degree instead of neighborhood radius has considerable effect in enhancing the algorithm. So, in general, FGSO algorithm is more successful than GSO algorithm.

## 6. CONCLUSION

In this paper, standard glowworm algorithm was improved by means of a fuzzy Gaussian membership function and a new algorithm called fuzzy glowworm swarm optimization (FGSO) was introduced. Experimental results on studied functions showed that introduced algorithm was of better error average than standard glowworm algorithm's error average for all functions. In another word, presented algorithm could improve algorithm convergence to a high extent and increase glowworm algorithm speed by improving local search. The new glowworm algorithm, even in some functions, acts much better than PSO algorithm. In future works, to enrich the search behavior of presented algorithm and prevent from being tangled in local traps, chaotic dynamic could be utilized.

## REFERENCES

[1] Miller, B. L., Shaw, M. J., *"Genetic algorithm with dynamic niche sharing for multiple function*", in Proceeding of the IEEE International Conference on Evolutionary Computing, pp. 786-791, 1996.

[2] Brits, R., Engel Bregh, A.P., van denBerge, F., "*A niching sharing for multimodal function optimization*", in Proceeding of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, pp. 692-696, 2002.

[3] Krishnanand, K. N., Ghose, D., *"Detection of multiple source locations using a glowworm metaphor with applications to collective robotics*", In Proceedings of IEEE swarm intelligence symposium , Piscataway IEEE Press, pp. 84–89, 2005.

[4] Krishnanand, K. N., Ghose, D., *"Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications"*, in Multiagent and Grid Systems, pp. 209–222, 2006.

[5] Krishnanand, K. N., Ghose D., *"Glowworm swarm optimisation: a new method for optimising multi-modal functions"*, Int. J. Computational Intelligence Studies, Vol. 1, No. 1, 2009.

[6] Krishnanand, K.N., Ghose D., *"Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions"*, Swarm Intell 3: pp. 87–124, 2009.

[7] Krishnanand, K.N., Ghose, D., *"Glowworm Swarm Optimization for Searching Higher Dimensional Spaces*", C.P. Lim et al. (Eds.): Innovations in Swarm Intelligence, SCI 248, Springer-Verlag Berlin Heidelberg, pp. 61–75, 2009.

[8] He, D., Zhu, H., "*Glowworm Swarm Optimization Algorithm Based on Multi population*", Sixth

 

International Conference on Natural Computation, 2010.

[9] Zhang J., Zhou G., Zhou, Y., "*A New Artificial Glowworm Swarm Optimization Algorithm Based on Chaos Method*", B.-Y. Cao et al. (Eds.): Quantitative Logic and Soft Computing 2010, AISC 82, Springer-Verlag Berlin Heidelberg, pp.683–693, 2010.

[10] R. R. Yagar and L. A. Zadeh, "*An Introduction to Fuzzy Logic Applications in Intelligent Systems*" , Kluwer Academic Publishers, pp. 69-70, 1992.

[11] Abdelbar, A., Abdelshahid, S., "*Fuzzy Pso: A Generalization of Particle Swarm Optimization"*, Proceeding of International Joint Conference on NeuralNetwork, Canada, 2005.

 