Yongquan ZHOU, Jiakun LIU, Guangwei ZHAO
Guangxi University for Nationalities

Leader Glowworm Swarm Optimization Algorithm for Solving Nonlinear Equations Systems

***Abstract**. This paper presents a leader glowworm swarm optimization algorithm (LGSO) for solving nonlinear equations systems. Since glowworm swarm optimization algorithm has bad optimized ability at high dimension, proposing glowworm swarm optimization algorithm with leader mechanism to strengthen the global optimization ability. Through various types nonlinear equations testing, experiment results show that the proposed algorithm has strong global searching capability and quickly finding the solutions of the equations, thus obviously improving the optimization global ability.*

***Streszczenie.** Zaprezentowano optymalizacyjny algorytm mrówkowy "świetlikowy" do rozwiązywania system równań nieliniowych. Ponieważ algorytm ten ma słabe możliwości optymalizacyjne przy dużych rozmiarach wprowadzono wspomagający mechanizm prowadzący „leader". (**Algorytm mrówkowy świetlikowy do rozwiązywania systemu równań nieliniowych**)*

Keywords: Glowworm Swarm Optimization (GSO); Leader Glowworm Swarm Optimization (LGSO); Nonlinear Equation Systems.
Słowa kluczowe: algorytm mrówkowy, równania nieliniower.

Introduction

Glowworm swarm optimization (GSO) [1-3] is a new method of swarm intelligence based algorithm for optimizing multi-modal functions raised by K. N. Krishnanad and D. Ghose in 2005. This algorithm becomes a new research hotspot of computational intelligence draw our sights on it. With the research deeper and deeper, it's been used at noisy text of sensor [4] and simulating robots [5]. Though GSO has strong universal, but there are also premature, the search accuracy is not high, optimization of high-dimensional is poor, post-iteration efficiency is not high defects. In response to these defects, many scholars proposed many improvements algorithm according to their field of study features, but the algorithms' improving ability of high-dimensional optimization is not quite obviously. In the large-scale optimization problems, having poor ability of searching optimization in high dimensional space is a major problem of constraining glowworm swarm algorithm's application.

Systems of nonlinear equations arise in many domains of practical importance such as engineering, mechanics, medicine, chemistry, and robotics. Solving such a system involves finding all the solutions (there are situations when more than one solution exists) of the polynomial equations contained in the mentioned system. The problem is nondeterministic polynomial-time hard, and it is having very high computational complexity due to several numerical issues [12]. There are several approaches for solving these types of problems. Van Hentenryck et al. [12] divided these approaches into two main categories: 1) interval methods that are generally robust but tend to be slow; 2) continuation methods that are effective for problems for which the total degree is not too high [12].

The limitations of Newton's method are pointed out in the aforementioned works. Bader [13] mentioned that standard direct methods, such as Newton's method, are impractical for large-scale problems because of their high linear algebra costs and large memory requirements. Bader proposed a tensor method using Krylov subspace methods for solving large scale systems of linear equations. There is a condition to be fulfilled--the equations must be continuously differentiable at least once. Bader's paper also provides a good review of similar research for solving systems of equations. Krylov subspace methods based on moment matching are also used by Salimbahrami and Lohmann [14]. Effati and Nazemi [15] proposed a very efficient approach for solving nonlinear systems of equations. Although there are several existing approaches for solving systems of nonlinear equations, there are still limitations of the existing techniques, and, still,
more research is to be done.

This paper introduces the leading ideas to solve this problem. Proposed a leader glowworm swarm optimization (LGSO), and applied to solving systems nonlinear equations. Experimental results show that the LGSO algorithm has good results in high-dimensional space optimization and accuracy of the solution of equations.

This paper is organized as follows. In section 2, the problem of systems nonlinear equations is described. In section 3, a basic glowworm swarm optimization is introduced. In section 4 we will introduce our leader glowworm swarm optimization, algorithm followed by the experimental results and analysis in section 5. The conclusions are given in section 6.

Description of the Problem of Nonlinear Equation

A system of nonlinear equations should include at least two or more linear or nonlinear equations. The system of equations will have a solution just when it is non-singular and has a point in the real-value continuous space.

Generally, the systems of nonlinear equations can be described as the following:

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0 \\ f_2(x_1, x_2, ..., x_n) = 0 \\ .... \\ f_n(x_1, x_2, ..., x_n) = 0 \end{cases} \tag{1}$$

where $f : R^n \to R$; $\{X = x_1, x_2, ..., x_n \,|\, x_i \in R^n\}$.

Supposed X^* is the solution of the system of equation, X^* should match each equation in the system. Then the system of equations has the same solutions as the following equation:

$$\sum_{i=1}^{n} |f_i(x_1, x_2, ..., x_n)| = 0 \tag{2}$$

defining $P(X) = \sum_{i=1}^{n} |f_i(x_1, x_2, ..., x_n)|$, X is an arbitrary point in the real-value space. Obviously, there is $P(X) \geq 0$.

So, the search of X^* can be treaded as a search for the minimum of the problem $P(X)$.

Based on the above analysis, the solving of a system of equations can be transformed into the following minimization one:

$$MinP(X)=\sum_{i=1}^{n}\left|f_i(x_1,x_2,......,x_n)\right| \quad (3)$$

In the following work, GSO will be introduced to find out the solution of the above problem. With equation (3) as the computational model for the fitness function.

Basic Glowworm Swarm Optimization Algorithm (GSO)

In GSO, each glowworm distributes in the objective function definition space. These glowworms carry own luciferin respectively, and has the respective field of vision scope called local-decision range. Their brightness concerns with in the position of objective function value. The brighter the glow, the better is the position, namely has the good target value. The glow seeks for the neighbor set in the local-decision range, in the set, a brighter glow has a higher attraction to attract this glow toward this traverse, and the flight direction each time different will change along with the choice neighbor. Moreover, the local-decision range size will be influenced by the neighbor quantity, when the neighbor density will be low, glow's policy-making radius will enlarge favors seeks for more neighbors, otherwise, the policy-making radius reduces. Finally, the majority of glowworm return gathers at the multiple optima of the given objective function.

Each glowworm i encodes the object function value $J(x_i(t))$ at its current location $x_i(t)$ into a luciferin value l_i and broadcasts the same within its neighbourhood. The set of neighbours $N_i(t)$ of glowworm i consists of those glowworms that have a relatively higher luciferin value and that are located within a dynamic decision domain and updating by formula (1) at each iteration.

Local-decision range update:

$$r_d^{\,i}(t+1)=\min\{r_s,\max\{0,r_d^{\,i}(t)+\beta(n_t-|N_i(t)|)\}\}; \quad (4)$$

and $r_d^{\,i}(t+1)$ is the glowworm i's local-decision range at the $t+1$ iteration, r_s is the sensor range, n_t is the neighbourhood threshold, the parameter β affects the rate of change of the nerghbourhood range.

The number of glow in local-decision range:

$$N_i(t)=\{j:\left\|x_j(t)-x_i(t)\right\|<r_d^{\,i};l_i(t)<l_j(t)\}; \quad (5)$$

and, $x_j(t)$ is the glowworm i's position at the t iteration, $l_j(t)$ is the glowworm i's luciferin at the t iteration.; the set of neighbours of glowworm i consists of those glowworms that have a relatively higher luciferin value and that are located within a dynamic decision domain whose range r_d^i is bounded above by a circular sensor range $r_s\ (0<r_d^{\,i}<r_s)$ .Each glowworm i selects a neighbour j with a probability $p_{ij}(t)$ and moves toward it. These movements that are based only on local information, enable the glowworms to partition into disjoint subgroups, exhibit a simultaneous taxis-behaviour toward and eventually co-locate at the multiple optima of the given objective function.

Probability distribution used to select a neighbour:

$$p_{ij}(t)=\frac{l_j(t)-l_i(t)}{\sum_{k\in N_i(t)}l_k(t)-l_i(t)}; \quad (6)$$

Movement update:

$$x_i(t+1)=x_i(t)+s\left(\frac{x_j(t)-x_i(t)}{\left\|x_j(t)-x_i(t)\right\|}\right); \quad (7)$$

Luciferin-update:

$$l_i(t)=(1-\rho)l_i(t-1)+\gamma J(x_i(t)); \quad (8)$$

and $l_i(t)$ is a luciferin value of glowworm i at the t iteration, $\rho\in(0,1)$ leads to the reflection of the cumulative goodness of the path followed by the glowworms in their current luciferin values, the parameter γ only scales the function fitness values, $J(x_i(t))$ is the value of test function.

The GSO algorithm as follows [11]:

Set number of dimensions= 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_agents_randomly;
for $i=1$ to n do $l_i(0)=l_0$
$r_d^i(0)=r_o$
set maximum iteration number= $iter\_\max;$
set $t=1$;
while $(t\le iter\_\max)$ do:
{
for each glowworm i do: %Lucifeerin-update phase
$l_i(t)=(1-\rho)l_i(t-1)+\gamma J(x_i(t));$ % See (8)
for each glowworm i do: %Movement- phase
{
$N_i(t)=\{j:\left\|x_j(t)-x_i(t)\right\|<r_d^{\,i};l_i(t)<l_j(t)\};$
for each glowworm $j\in N_i(t)$ do:
$p_{ij}(t)=\frac{l_j(t)-l_i(t)}{\sum_{k\in N_i(t)}l_k(t)-l_i(t)};$ % See (6)
$j=Select\_glowworm(\vec{p});$
$x_i(t+1)=x_i(t)+s\left(\frac{x_j(t)-x_i(t)}{\left\|x_j(t)-x_i(t)\right\|}\right);$ % See (7)
$r_d^{\,i}(t+1)=\min\{r_s,\max\{0,r_d^{\,i}(t)+\beta(n_t-|N_i(t)|)\}\};$
% See (4) }
$t\leftarrow t+1;$
}

Leader Glowworm Swarm Optimization Algorithm (LGSO)

Thought of LGSO Algorithm

In basic GSO algorithm foundation, introduce leader mechanisms. Before each generation of the algorithm, set the best glowworm's position as the leader in the current generation. After each generation, all glowworms are moved to the location of the leader, so that the glowworm swarm has high ability of searching global optimization, and improving the algorithm's ability in high dimensional space optimization. Improved algorithm updates the location as the following formula:

$$x_i(t) = x_i(t) + rand * (x_{leader}(t) - x_i(t)); \quad (9)$$

where, $x_{leader}(t)$ is the position of the t th leader, $x_i(t)$ is the position of the glowworm.

LGSO algorithm

The basic step of LGSO as below:

Step 1 Initialization parameters of $\rho, \gamma, \beta, s, l_0, m, n$, initialization the position of each glowworm.

Step 2 To each glow, updating luciferin value according to equation (8).

Step 3 Selects conforms to the condition glowworm according to equation (5).

Step 4 Select the optimal location of the current generation of glowworm swarm and set it to the leader

Step 5 Using (6) to select the distribution $j(j \in N_i(t))$, and updating with equation (7).

Step 6 Revision search radius by equation (4).

Step 7 Each glowworms move to the leader position according to (9).

Step 8 One iteration complete, enter the next iteration, judges whether to satisfy the termination condition, satisfied the withdrawal circulation, the record result, otherwise transferred Step 2.

The flow chart of LGSO as below:

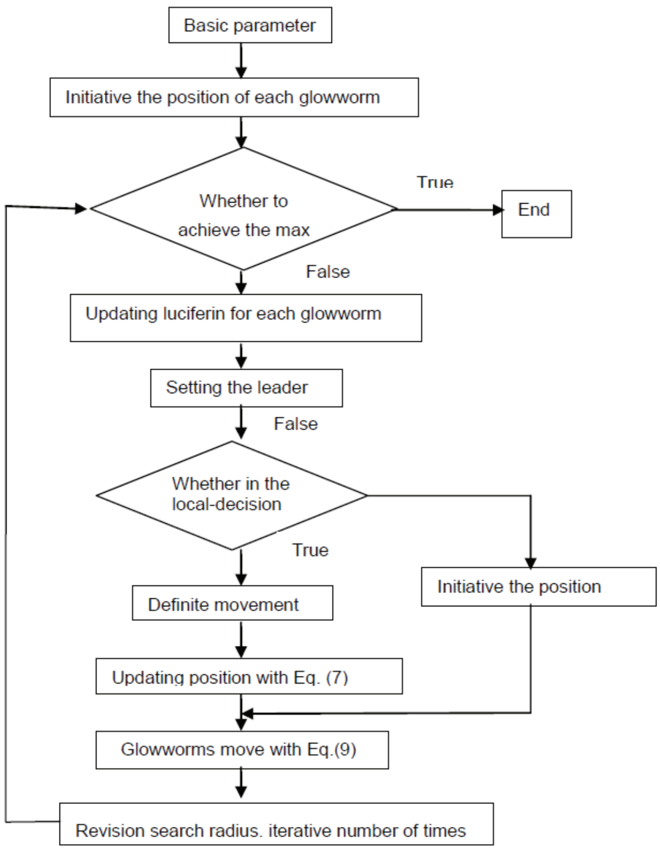

Figure 1. GSO-D Algorithm Flow Chart

Experimental Results Comparison

Test functions

In order to test the effective of our algorithm, we take there high-dimension test functions [9] to verify and compare with Particle swarm optimization (PSO), Artificial fish-swarm algorithm (ASFA) [7-8], GSO. The three test functions are as below:

F_1: $f_1(x) = \sum_{i=1}^{n} x_i^2$, $-100 \le x_i \le 100$ $(i = 1, 2, \ldots, n; n = 20)$, the objective function value is 0 at $(0, 0, \ldots, 0)$.

F_2 : $f_2(x)=\sum_{i=1}^{n}(x_i+0.5)^2$, $-10 \le x_i \le 10, (i=1,2,\ldots,n; n=20)$, the objective function value is 0 at $(0,0,\ldots,0)$.

F_3 : $f_3(x)=\sum_{i=1}^{n} i x_i^2, -5.12 < x_i < 5.12$, the objective function value is 0 at $(0,0)$.

Test nonlinear equations system

E_1 : $$\begin{cases} f_1(x_1,x_2)=\cos(2x_1)-\cos(2x_2)-0.4=0 \\ f_1(x_1,x_2)=2(x_2-x_1)+\sin(2x_2)-\sin(2x_1)-1.2=0 \end{cases}$$

E_2 : $$\begin{cases} f_1(x_1,x_2)=\exp(x_1)+x_1x_2-1=0 \\ f_1(x_1,x_2)=\sin(x_1x_2)+x_1+x_2-1=0 \end{cases}$$

E_3 : $$\begin{cases} 0 = x_1 - 0.25428722 - 0.18324857 x_4 x_3 x_9 \\ 0 = x_2 - 0.37842197 - 0.16275449 x_1 x_{10} x_6 \\ 0 = x_3 - 0.27162577 - 0.16955071 x_1 x_2 x_{10} \\ 0 = x_4 - 0.19807914 - 0.15585316 x_7 x_1 x_6 \\ 0 = x_5 - 0.44166728 - 0.19950920 x_7 x_6 x_3 \\ 0 = x_6 - 0.14654133 - 0.18922793 x_8 x_5 x_{10} \\ 0 = x_7 - 0.42937161 - 0.21180486 x_2 x_5 x_8 \\ 0 = x_8 - 0.07056438 - 0.17081208 x_1 x_7 x_6 \\ 0 = x_9 - 0.34504906 - 0.19612740 x_{10} x_6 x_8 \\ 0 = x_{10} - 0.42651102 - 0.21466544 x_4 x_8 x_1 \end{cases}$$

E_4 : $$\begin{cases} x_1^2+x_3^2=1 \\ x_2^2+x_4^2=1 \\ x_5x_3^3+x_6x_4^3=c_1 \\ x_5x_1^3+x_6x_2^3=c_2 \\ x_5x_1x_3^2+x_6x_4^2x_2=c_3 \\ x_5x_1^2x_3+x_6x_2^2x_4=c_4 \end{cases}$$, where $C_i, i=1,\ldots,4$ is random, we set them as 0;

E_5 : $$\begin{cases} (x_k + \sum_{i=1}^{n-k-1} x_i x_{i+k}) x_n - c_k = 0, 1 \le k \le n-1 \\ \sum_{l=1}^{n-1} x_l + 1 = 0 \end{cases}$$, where $C_i, i=1,\ldots,4$ is random, we set them as 0, $n=20$.

Test platform and parameter

The PSO, ASFA, GSO and LGSO are code in MATLAB7.0 and implemented on Intel Core2 T5870 2.00GHz machine with 2G RAM under windows 7 platform. The set of PSO, ASFA, GSO and LGSO's parameters are as below: $n=500$, max of iteration $\max t=300$, $\rho=0.4$, $\gamma=0.6$, $\beta=0.08$, moving step $s=0.03$, $n_t=5$ and initialization of luciferin $l_0=5$. In the test part of the equation, the literature compared with the literature [10].

Analyses of results

To test the function $F_1 \sim F_3$, do 10 times independent experiments, Find the best value, worst value, average, running times and compared with the PSO, ASFA, GSO. Result as table 1. PSO is fast, but has poor accuracy. Considering the run time and accuracy, this paper presents the LGSO for high-dimensional and the domain of a wider range of large-scale optimization problems has better optimization algorithm superior. Table 2~4 is the results of equations problems of Classical algorithm, we can see that the proposed algorithm LGSO the highest accuracy in the solution, computation time is also good.

Figure 2 to Figure 9 are LGSO、PSO、ASFA and GSO algorithm convergence curves, visually see the LGSO calculation of high accuracy, fast convergence and high-dimensional functions for greater processing power.

Table.1 Experimental comparison between PSO, ASFA, GSO and LGSO

Function	Algorithm	Best value	Worst value	Average value	Running time
F_1	PSO	22.3447889201	32.34743982752	27.1827846718	3.556s
	ASFA	4.572934e+002	8.234511e+002	6.469821e+002	308.759s
	LGSO	1.9087733e-004	4.4534521e-004	2.8392271e-004	142.146s
	GSO	2.2345496e+004	2.2234592e+004	1.7784242e+004	134.996s
F_2	PSO	0.1134592343	0.4234892034	0.2172830646	3.900s
	ASFA	)011232347	0.0045879232	0.0023437976	387.721s
	LGSO	)03452e-004	1.3423418e-004	1.1628304e-004	147.251s
	GSO	1.848132e+002	3.231412e+002	2.4531444e+002	145.605s
F_3	PSO	36.2312349179	65.2128934751	56.4081229098	4.228s
	ASFA	0.1323452662	0.8234521934	0.3863685092	526.283s
	LGSO	0.1132523443	0.6235421132	0.4310187476	227.247s
	GSO	2.4315125e+003	4.1239571e+003	3.4519933e+003	56s

Table.2 E_1 experimental of comparison

Algorithm	Solution	Functions values
Newton's method	(0.15,0.49)	(-0.00168,0.01497)
Secant method	(0.15,0.49)	(-0.00168,0.01497)
Broyden's method	(0.15,0.49)	(-0.00168,0.01497)
Effati's method	(0.1575,0.4970)	(-0.005455,0.00739)
EA approach	(0.157772,0.49458)	(0.001264,0.000969)
PSO	(0.16,0.5001)	(-0.012847,0.022260)
ASFA	(0.15752,0.4983)	(0.144787, 0.088160)
PGSO	(0.1575,0.4969)	(-0.001676, 0.009154)
GSO	(0.15752,0.4983)	(-0.001681, 0.009161)

Table.3 E_2 experimental of comparison

Algorithm	Solution	Functions values
Effati's method	(0.0096,0.9976)	(-0.019223,0.016776)
EA approach	(-0.00138,1.00270	(-0.00276,-6.37E-005)
PSO	(-0.00162,1.02324)	(-0.13542,1.2317)
ASFA	(0.000087,1.00813)	(-3.23E-003,-0.021854)
PGSO	(-0.00006,1.00017)	(-4.24E-004,0.0021218)
GSO	(-0.00222,1.00364)	(-0.551603, 1.646274)

Table.4 $E_3E_4E_5$ experimental of comparison

Equations	Algorithm	Best value	Worst value	Average value	Running time
E_3	PSO	0.2149153531	1.0626433514	1.0003785224	4.024s
	ASFA	0.0420855131	0.6339623349	0.2432108752	401.139s
	LGSO	0.0116202850	0.0131201501	0.0119872412	120.931s
	GSO	6.9591574399	7.2428441893	7.0012453321	117.453s
E_4	PSO	0.0224930425	0.1166318044	0.097765522	2.745s
	ASFA	0.0068685785	0.0076475297	0.007124334	280.548s
	LGSO	0.0019527707	0.0060514520	0.003444587	78.513s
	GSO	0.661728548	0.9835675001	0.887666342	77.048s
E_5	PSO	2.4298693e-004	5.4097120e-004	4.0032155e-004	17.518s
	ASFA	1.2387994e-005	6.6280149e-005	3.8923456e-005	1389.453s
	LGSO	2.6788358e-005	3.0290678e-005	2.9871524e-005	169.743s
	GSO	0.0305069500	0.1231600172	0.095567322	159.381s

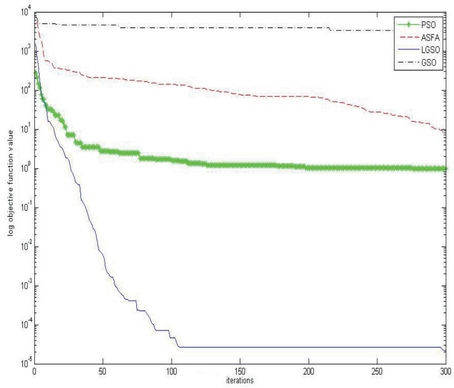

Fig. 2 F_1 curves of the objective functions value

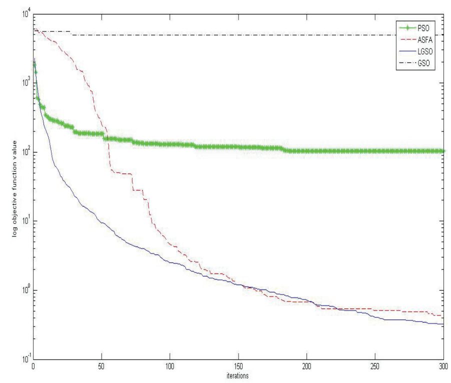

Fig. 4 F_3 curves of the objective functions value

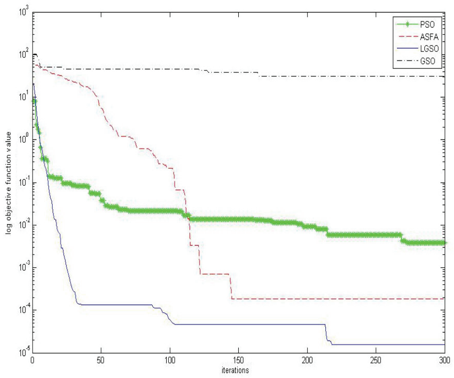

Fig. 3 F_2 curves of the objective functions value

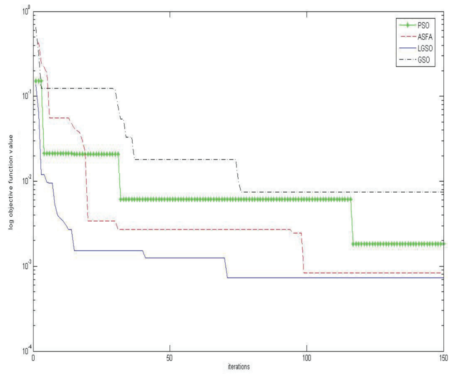

Fig. 5 E_1 curves of the objective functions value

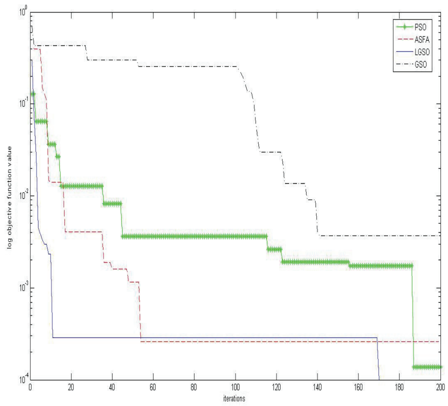

Fig. E_2 curves of the objective functions value

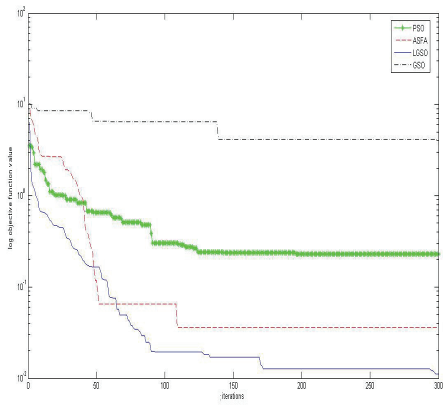

Fig. 7 E_3 curves of the objective functions value

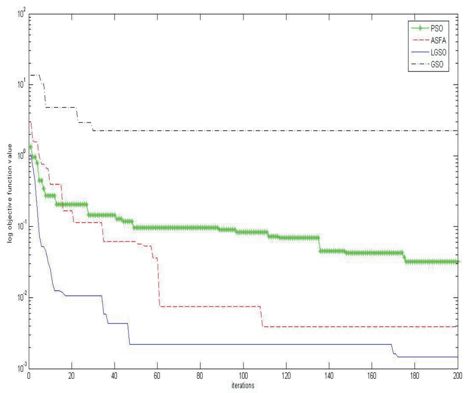

Fig. 8 E_4 curves of the objective functions value

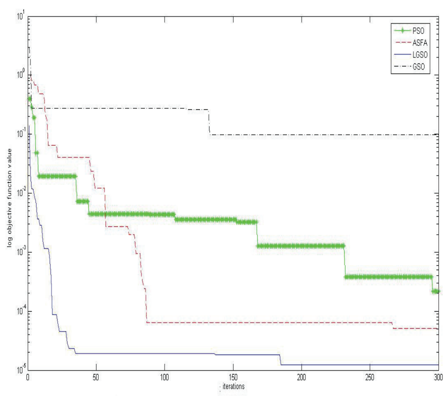

Fig. 9 E_5 curves of the objective functions value

Conclusions

For Glowworm swarm optimization (GSO) the accuracy is not high, the disadvantage of slow convergence, this paper presents leader glowworm swarm optimization (LGSO). In basic GSO algorithm foundation, introduce leader mechanisms. Before each generation of the algorithm, set the best glowworm's position as the leader in the current generation. After each generation, all glowworms are moved to the location of the leader, so that the glowworm swarm has high ability of searching global optimization, and improving the algorithm's ability in high dimensional space optimization. Experiments show that the new algorithm has strong global search ability and fast convergence rate, accuracy has greatly improved, and can be effectively used in high-dimensional problems of nonlinear equations.

Acknowledgment. *This work was supported by the Grants 0991086 from Guangxi Science Foundation. Funded By Open Research Fund Program of Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China under Grant: IPIU012011001.*

REFERENCES

[1] Krishnanand K. N. D. Ghose D. "Glowworm Swarm Optimization: A New Method for Optimizing Multi-modal Functions".Computational Intelligence Studies, 1(1):93-119, 2009.
[2] Krishnanand K.N. "Glowworm Swarm Optimization: A Multimodal Function Optimization Paradigm with Applications to Multiple Signal Source Localization Tasks". Indian: Department of Aerospace Engineering, Indian Institute of Science, 2007.
[3] Krishnanand K.N. and Ghose, D. "Theoretical Foundations for Rendezvous of Glowworm-inspired Agent Swarms at Aultiple Locations".,Robotics and Autonomous Systems, 7(56): 549-569, 2008.
[4] Krishnanand K.N. and Ghose D. "A Glowworm Swarm Optimization Based Multi-robot System for Signal Source Localization". Design and Control of Intelligent Robotic Systems, 53-74,2009.
[5] Krishnanand K.N. and Ghose, D. "Chasing Multiple Mobile Signal Sources: A Glowworm Swarm Optimization Approach". In Third Indian International Conference on Artificial Intelligence (IICAI 07), Indian,2007.
[6] Kennedy J, Eberhart R C. "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Service Center, IV: 1942-1948, 1995.
[7] Xiao lei Li, Jixi an lan. "Artificial Fish-swarm Algorithm: Bottom-up Optimization Model". Trans Annual Meeting of Chinese Process Systems Engineering Society., pp.76-82, 2001.
[8] Xiao lei Li, Zhejiang Shoo and Jixi an lan. "An Optimizing Method based on Autonomous Animates: Fish-swarm Algorithm", Systems Engineering-Theory & Practice. 22(11):32-38, 2002,
[9] Kokum Deep, Jadish Chan Banal. "Mean Particle Swarm Optimization for Function Optimization". Computational Intelligence Studies, 1(1):72-91. 2009.
[10] Carina Grosan and Ajith Abraham. "A New Approach for Solving Nonlinear Equations Systems".IEEE Trans. Systems and Humans, 38(3):698-714, 2008.
[11] Krishnanand K. N. D. Ghose D. "Glowworm Swarm Optimization for Simultaneous Capture of Multiple Local Optima of Multima of Multimodal Functions", Swarm Intell, 3:87-124, 2009.
[12] P. Van Hentenryck, D. McAllester, and D. Kapur, "Solving Polynomial Systems Using a Branch and Prune Approach," SIAM J. Numer. Anal., 34(2):797-827, 1997.
[13] B. W. Bader, "Tensor–Krylov Methods for Solving Large-scale Systems of Nonlinear Equations," SIAM J. Numer. Anal., 43(3): 1321-1347, 2005.
[14] B. Salimbahrami and B. Lohmann, "Order Reduction of Large Scale Second-order Systems Using Krylov Subspace Methods," Lin. Alg. Appl., 415(2/3): 385-405, 2006.
[15] S. Effati and A. R. Nazemi, "A new method for solving a system of the nonlinear equations," Appl. Math. Comput.,168(2): pp. 877-894, 2005.

***Authors**: prof. dr Yongquan Zhou, College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006, China, E-mail: yongquanzhou@126.com; dr Jiakun Liu, College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006, China.*