PSO algorithm based on GSO and application in the Constrained optimization

Hongxia Liu
Department of the Computer application Engineering
Daqing Vocational College
DaQing,HeilongJiang
gxun2008@163.com

Feng Zhou
Faculty of Information Technology
Shandong Institute of Commerce and Technology
Ji'Nan, ShanDong
zhoufengkey@163.com

***Abstract*--In particle swarm algorithm, the introduction of glowworm algorithm, according to domain to determine a perception range, within the scope of perception of all the particles find an extreme value point sequence, which apply roulette method, choose a particle instead of global extreme value.So as to scattered particle, and avoid the local minima.Experimental results demonstrate the feasibility and effectiveness of the method.**

Keyword-Particle swarm optimization;Glowworm swarm optimization; Constrained optimization.

I. INTRODUCTION

Particle Swarm Optimization (PSO) [1] is a population-based continuous optimization technique proposed by Kennedy and Eberhart. Like ant colony optimization algorithms or genetic algorithms, PSO is biologically inspired. In this case, the algorithm is inspired by the social behavior of animals living in groups. The algorithm simulates a simplified social milieu in a swarm of potential solutions (called "particles"), which means that a single particle bases its search not only on its own experience but also on the information given by its neighbors in the swarm. This paradigm leads to successful results and contributes to the popularity of PSO. But particle swarm algorithm itself trapped into local optimal solution easily，in order to solve the shortcomings, put the perception range of glowworm algorithm into PSO method, scattered particle method with roulette, avoid algorithm into the local extremum.

II. ALGORITHM INTRODUTION

A. glowworm swarm optimization

In GSO[3], physical agents $\{i: i=1,2,\cdots,n\}$ are considered that are initially randomly deployed $\{x_i(0): x_i \in R^m, i=1,2,\cdots,n\}$ in the objective function space R^m, each agent in the swarm decides its direction of movement by the strength of the signal picked up from its neighbors. This is somewhat similar to the luciferin induced glow of a glowworm which is used to attract mates or prey. The brighter the glow, the more is the attraction. It is inspired from this behavior; we come up with an intelligent algorithm of function optimization. In GSO algorithm, the glowworms in GSO are endowed with other behavioral mechanisms (not found in their natural counterparts) that enable them to selectively interact with their neighbors and decide their movements at iteration.

Each glowworm i encodes the objective 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 neighborhood. The set of neighbors $N_i(t)$ of glowworm i consists of those glowworms that have a relatively higher luciferin value and that are located within a dynamic decision domain whose range $r_d^{\,i}$ is bounded above by a circular sensor range r_s $(0 < r_d^{\,i} < r_s)$. By the following formula to determine the numbers of glowworm within a dynamic decision domain:

$$N_i(t) = \{ j : \left\| x_j(t) - x_i(t) \right\| < r_d^{\,i}; l_i(t) < l_j(t) \} \quad (1)$$

Where, $x_j(t)$ is the location of glowworm j at the t-th iteration, $l_j(t)$ is the luciferin value of glowworm j at the t-th iteration;

Update formula of the dynamic decision domain as follows:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \quad (2)$$

Where, $r_d^{\,i}(t+1)$ is the decision domain of the glowworm i at the $(t+1)$-th iteration (radius), r_s is a circular sensor range, n_t represents the neighborhood threshold, the parameter β affects the rate of change of the neighborhood range.

Luciferin-update formula:

$$l_i(t) = (1-\rho) l_i(t-1) + \gamma J(x_i(t)) \quad (3)$$

Where, $l_i(t)$ is the luciferin value of glowworm i at the t-th iteration, $\rho \in (0,1)$ represents control parameters, the parameter γ only scales the function fitness values, $J(x_i(t))$ is the objective function value.

Each glowworm i selects a neighbor j with a probability $p_{ij}(t)$ and moves toward it. According to (4) update the location. These movements that are based only on local information, enable the glowworms to partition into disjoint subgroups, exhibit a simultaneous taxis-behavior toward and eventually co-locate at the multiple optima of the given objective function.

Location-update formula:

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

B. Particle Swarm Optimization

PSO is essentially a population-based algorithm. It starts with a random initialization of a swarm of particles. Each particle is modeled by its position in the search space and its corresponding velocity. In a d -dimensional search space, the position and the velocity of the i th particle can be represented as $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ and $V_i = (v_{i1}, v_{i2}, \cdots, v_{id})$, respectively. Each particle i .The neighborhood of each particle can be chose using either a fixed topology, or time-varying topology, or a random topology .The quality of a given position is evaluated with respect to an objective function.

Each particle i has its own best location $P_i = (p_{i1}, p_{i2}, \cdots, p_{id})$,which corresponds to the best location particle i has reached until time k .The global best location is denoted by $P_g = (p_{g1}, p_{g2}, \cdots, p_{gd})$,which represents the best location reached by the neighbors of the i th particle. From time k to time $k+1$, each velocity is updated using the following equation:

$$v_{id}^{k+1} = w v_{id}^{k} + c_1 r_1 (p_{id}^{k} - x_{id}^{k}) + c_2 r_2 (p_{gd}^{k} - x_{gd}^{k}) \quad (5)$$

The computation of the position at time $k+1$ is given by:

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (6)$$

Where w is a constant, called inertia weight, c_1 and c_2 are constants called acceleration coefficients, and r_1 and r_2 are two independent random numbers uniformly distributed in [0, 1] for each dimension at each time step. w controls the influence of the previous direction of displacement. c_1 controls the influence of the particle's memory on the particle's behaviors, and c_2 controls the influence of the swarm on the particle's behavior. The combination of the values of w , c_1 and c_2 may favor either intensification or diversification.

III. PARTICLE SWARM OPTIMIZATION ALGORITHM BASED ON THE GLOWWORM THOUGHT

A. Described the Algorithm

Specific means is: Set the sense range and particle threshold, by current particle as the center, sense range as a radius, form a regional, as shown in Fig. 1. Calculated the distance between the particles i and j , comparing with the pre-set threshold, if less than threshold and particles j fitness value is smaller. Record the serial number of particle j .if in the near particle i , no have better particle, thinks that this particle is a local excellent value. The particles were chosen as a local extreme value point sequence, what use roulette method, choose a particle to replace the global extremum pg . Find all particles extreme value point sequence through the comparison. Then use roulette method to disperse particles. Avoid into the local extremum.Penalty function using adaptive type penalty function method, $f(\mathrm{x})$:Goal function, $g_i(\mathrm{x})$ and $\mathrm{f_i(x)}$ are the constraint condition. as follow:

$$\Phi(x) = f(x) + \lambda(t) \times \{\sum_{i=1}^{N} (\max[0, g_i(x)]^2 + \sum_{j=1}^{p} |h_j(x)|\} \quad (7)$$

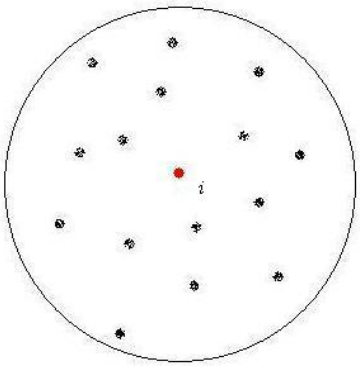

Figure 1. Sketch map of sense range

B. Algorithm steps

Step 1 Initialization: velocity, position, maximum iterating times, particle swarm scale and accelerated factor etc;

Step 2 Select qualified particles in the sense range，composing an extreme value point sequence;

Step 3 Sorting out by roulette method, one of extreme value point sequence particle replace global extreme value pg ;

Step 4 With basic particle swarm algorithm searching, find out optimal value;

4.1 Updated particle speed and position, and handled them if over the regional;

4.2 Updated the individual extremum;

Step 5 Judge whether meet the end conditions, yes, turn step6, exit the loop, output the result; else, turn step2, continue to search;

Step 6 Output target value and stop algorithm.

IV. NUMERICAL SIMULATION AND ANALYSIS

TABLE1. Optimal solution of each algorithm for expansion rope

Decision variables	New	HPSO[11]	CPSO[12]	Ref. [5]	Ref. [4]	Ref. [6]	Ref. [7]
$x_1(d)$	0.051667	0.051706	0.051728	0.050000	0.053396	0.051480	0.051989
$x_2(D)$	0.355224	0.357126	0.357644	0.315900	0.399180	0.351661	0.363965
$x_3(P)$	11.217496	11.265083	11.244543	14.250000	9.185400	11.632201	10.890522
$g_1(x)$	0.000000	-0.001267	-0.000845	-0.000014	0.000019	-0.002080	-0.000013
$g_2(x)$	-0.1451501	-0.003782	-0.000013	-0.003782	-0.000018	-0.000110	-0.000021
$g_3(x)$	-4.051547	-3.938301	-4.051300	-3.938302	-4.123832	-4.026318	-4.061338
$g_4(x)$	-0.728214	-0.756066	-0.727090	-0.756067	-0.698283	-04.026318	-0.722698
$f(x)$	**0.0126652**	**0.0126652**	0.012675	0.012833	0.012730	0.012705	0.012681

TABLE2. Statistical results of each algorithm for expansion rope

Methods	Best	Mean	Worst	Std. Dev.
New	**0.0126652**	**0.0126742**	0.012754	**6.3548e-6**
HPSO[11]	0.0126652	0.0127072	0.012719	1.5824e-5
CPSO[12]	0.012675	0.012730	0.012924	5.198500e-5
ref. [5]	0.012833	N/A	N/A	N/A
ref. [4]	0.012730	N/A	N/A	N/A
ref. [6]	0.012705	0.012769	0.012822	3.939000e-5
ref. [7]	0.012681	0.012742	0.012973	5.900000e-5

TABLE3. Optimal solution of each algorithm for welded stripe

Decision variables	New	HPSO[11]	CPSO[12]	Ref.[9]	Ref. [10]	Ref. [6]	Ref. [7]
$x_1(h)$	0.204845	0.205730	0.204381	0.2455	0.2489	0.2088	0.2060
$x_2(l)$	3.470145	3.470489	3.505107	6.1960	6.1730	3.4205	3.4713
$x_3(t)$	9.036641	9.036624	9.033546	8.2730	8.1789	8.9975	9.0202
$x_4(b)$	0.205729	0.205730	0.205878	0.2455	0.2533	0.21	0.2065
$g_1(x)$	-3.245e-008	-0.025399	-12.839796	-5743.826517	-5758.603777	-0.337812	-0.074092
$g_2(x)$	-4.4128e-006	-0.053122	-1.247467	-4.715097	-255.576901	-353.902604	-0.266227
$g_3(x)$	-5.45618e-009	0	-0.001498	0	-0.004400	-0.0012	-0.000495
$g_4(x)$	-3.431148	-3.432980	-3.429347	-3.020289	-2.982866	-3.411865	-3.430043
$g_5(x)$	-0.079935	-0.080730	-0.079381	-0.120500	-0.123900	-0.0838	-0.080986
$g_6(x)$	-0.028574	-0.235540	-0.235536	-0.234208	-0.234160	-0.235649	-0.235514
$g_7(x)$	-2.802e-006	-0.033154	-11.681355	-3604.275002	-4465.270928	-363.232384	-58.66644
$f(x)$	**1.724635**	1.724852	1.728024	2.385937	2.433116	1.748309	1.728226

TABLE4. Statistical results of each algorithm for welded stripe

Methods	Best	Mean	Worst	Std. Dev.
New	1.724635	**1.737210**	**1.765217**	0.018745
HPSO	1.724852	1.749040	1.814295	0.040049
CPSO	1.728024	1.748831	1.782143	0.012926
Ref. [9]	2.385937	N/A	N/A	N/A
Ref.10]	2.433116	N/A	N/A	N/A
Ref. [6]	1.748309	1.771973	1.785835	**0.011220**
Ref. [7]	1.728226	1.792654	1．993408	N/A

From table1, $f(x)$ express that the elastic string is minimum; average solution and the standard deviation from this paper were less than literature algorithm. Also means that this paper algorithm(New) is stable, namely good robustness; and gets the worst solution smaller than HPSO algorithm slightly.

E.g.2 Design problem for welded strip [8]: aim to seek four design variable T_s、R、L、T_h and $b(x_4)$ to meet shear stress τ、bending stress σ、buckling load on the bar P_c、end deviation δ and boundary conditions. This problem is designed welded beam for minimum cost.

From the table3, improved method better than literature algorithm, no matter the optimal solution, average solution and the worst solution, this again confirms the fact that the average performance of this paper algorithm better than other algorithms.

V. CONCLUSIONS

This paper put perception range thought of glowworm algorithm applied to particle swarm algorithm; find out the extreme value point sequence within the scope, use roulette method to choose a particle as global extreme value. This method can overcome the defects that particle swarm algorithm into the local optimal solution easily.

REFERENCES

[1] Kenndey J, Eberhart R. Particle Swarm Optimization [A]// Proceedings of IEEE International Conference on Neural Networks[c]. IEEE Press, 1995:1942-1948.

[2] Krishnanand,K.N.D. Ghose,D. Glowworm swarm optimization: a new method for optimizing multi-modal functions[J].Computational Intelligence Studies, 2009,1(1):93-119.

[3] Hongxia Liu, Yongquan Zhou. A Novel Hybrid Optimization Algorithm Based on Glowworm Swarm and Fish School [J] (Journal of Computational Information Systems, 2010, 13(6):4533-4541.

[4] Arora J S. Introduction to Optimization Design. New York: McGraw-Hill, 1989.

[5] Belegundu Ashok D.Jasbir S. A study of Mathematical Programming Methods for Structural Optimization [J] International Journal for Numerical Methods in Engineering, 1985, 9(21):1583-1599.

[6] Coello C A C. Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry, 2000, 41:113-127.

[7] Coello C A C, Montes E M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Advanced Engineering Informatics, 2002, 7(16): 193-203.

[8] Rao S S. Engineering optimization (third Ed).New York: Wiley, 1996.

[9] Deb K. Optimal design of a welded beam via genetic algorithms. AIAA Journal, 1991, 29:2013-2015.

[10] Ragsdell K M, Phillips D T. Optimal design of a class of welded structures using geometric programming. ASME Journal of Engineering for Industries, 1967, 98:1021-1025.

[11] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation, 2007, 186: 1407-1422.

[12] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence, 2007, 20: 89-99.