Glowworm Swarm Optimization Algorithm for Solving Double Numerical Integration

Yan Yang*, Shengjian Liu and Lin Luo
South China Institute of Software Engineering, GU, Guangzhou, 510990, China
E-mail: yangyan_08@yeah.net

Abstract. A new method for solving double numerical integration based on artificial glowworm swarm optimization algorithm (GSO) is presented. GSO is used to optimize the node points on both direction range in rectangular integration domain to get a more precise integration result. Simulation examples of integration show the algorithm is a validated method with high precision and powerful self-adapting. The algorithm has value in numerical calculation and engineering practice.

1. Introduction
Solving double numerical integration is one of the common problems in scientific calculations and engineering technology field. So far, people have made a lot of researches to solve double numerical integration using theoretical and computational methods. There are some traditional methods such as Trapezoid method, Newton-Cotes method, Gauss method, and Simpson’s method [1-3]. With the appearing of intelligence algorithms [4], some scholars calculate the double numerical integration with evolutionary strategy, PSO and AFSA [5-7]. So, it’s very necessary to study the new swarm intelligence algorithm for solving double numerical integration problems. However, if the antiderivative is not easy to obtain or the integrand can not find antiderivative, we would not gain integral value or find imprecise value with the traditional methods.

In 2005, Krishnanand and Ghose put forward artificial glowworm swarm optimization (GSO) algorithm based on glowing to attract mates or prey of glowworm in the nature [8]. GSO algorithm is a new swarm intelligence bionic algorithm and it has good capacity to search for global extremum and more extremums of multimodal optimization problems. The GSO algorithm was used in many fields, such as multimodal optimization problem, noise issues, theoretical foundations, signal source localization, tracking multiple mobile signal sources.

A new method for solving double numerical integration based on artificial glowworm swarm optimization algorithm is presented. In the algorithm, some points are generated randomly in the rectangular integral interval, and each point is regarded as an artificial glowworm. Then the points are optimized according to the moving principle of artificial glowworms and sorted according to concave and convex shape of image. Then calculate the sum of integral by the optimal node which has been optimized and get more precise integration values.

The paper is organized as follows. Section 2 describes the principle of artificial glowworm swarm optimization algorithm. Then the artificial glowworm swarm optimization algorithm for solving integral is described in Section 3. Numerical simulation results and discussions are presented in Section 4. Finally, Section 5 provides some concluding remarks based on the results in Section 4.
2. The principle of GSO algorithm

A set of $N$ glowworms are randomly deployed in a $m$-dimensional workspace. According to the similarity of luciferin value, divide the swarm into $nei$ neighbors and each glowworm $i$ selects a neighbor $j$ with a probability $p_{ij}$ and moves toward it within its decision domains range $R_d(0 < R_d^i < R_s)$, where $R_s$ is a circular sensor range of glowworm $i$. The position of glowworm $i$ is $x_i(x_j \in R^m, i = 1, 2, \ldots, N)$, which is a potential solution. Put $x_i$ into the objective function and gain the fitness function value $J(x_i)$ and luciferin value $l_i$. Estimate the solution with luciferin value.

The algorithm can gain the optimal value of functions. The equations that modeled the luciferin-update, probability distribution used to select a neighbour, movement update and local-decision range update are given as below:

$$l_i(t) = \max \{ (0, (1 - \rho)^*l_i(t - 1) + \gamma^*J(x_i(t))) \} \tag{1}$$
$$P_j(t) = l_j(t) / \sum_{k \in N_j(t)} l_k(t) \tag{2}$$
$$x_i(t + 1) = x_i(t) + step \cdot ((x_j(t) - x_i(t))/\|x_j(t) - x_i(t)\|) \tag{3}$$
$$R_{ij}(t + 1) = R_s / (1 + \beta^*D_{ij}(t)) \tag{4}$$

Where $N_j(t) = \{ j : \| x_j(t) - x_i(t) \| < R_{ij}^d(t); l_j(t) < l_i(t) \}$ is a neighbour of glowworm $i$ consisting of those glowworms that have a relatively higher luciferin value and that are located within a dynamic decision domain. If the luciferin value of glowworm $i$ is greater than $j$’s and the distance between the glowworm $i$ and $j$ is less than the dynamic decision domain, divide glowworm $j$ into the neighbours of glowworm $i$. $D_i(t) = N_j(t) / (\pi^*R_s^2)$ is the neighbour-density of glowworm $i$ at iteration $t$ and $\beta$ is a constant parameter. The constant parameter $\gamma$ affects the rate of change of the neighbourhood range. The constant parameter $\rho$ decides whether algorithm has memory. A value $\rho = 0$ renders the algorithm memory less where the luciferin value of each glowworm depends only on the fitness value of its current position. However, $\rho \in (0,1]$ leads to the reflection of the cumulative goodness of the path followed by the glowworms in their current luciferin values. The constant parameter $\gamma$ can scale the function fitness values. The value of step-size $step$ influences the range of objective function.

3. GSO algorithm for solving double numerical integral

3.1. A subsection

On the $x$ axis direction, the individual glowworm is consisted of position $x$ and luciferin value $LX$, each part has $S$ components; and on the $y$ axis direction, the individual glowworm is consisted of position $Y$ and luciferin value $LY$, each part has $D$ components as follows:

$$\begin{align*}
(X, LX) &= ((x_1, x_2, \ldots, x_s), (lx_1, lx_2, \ldots, lx_s)) \tag{5} \\
(Y, LY) &= ((y_1, y_2, \ldots, y_d), (ly_1, ly_2, \ldots, ly_d)) \tag{6}
\end{align*}$$

where, $S$ and $D$ are the numbers of nodes in the integral interval of $x$ and $y$ axes. $(x_1, x_2, \ldots, x_s)$ is the node of the integral interval, and $(lx_1, lx_2, \ldots, lx_s)$ is the luciferin value related to the node on the $x$ axis direction. $(y_1, y_2, \ldots, y_d)$ is the node of the integral interval, and $(ly_1, ly_2, \ldots, ly_d)$ is the luciferin value related to the node on the $y$ axis direction.
3.2 The algorithm implementation steps

**Step 1.** Initialize a population. Randomly generate a group containing \( N \) glowworm individuals on the two axes direction respectively. On the \( x \) axis direction, each glowworm individual \( (X_i, LX_i) \) contains \( S \) components. And each glowworm individual \( (Y_i, LY_i) \) contains \( D \) components of the \( y \) axis direction. Set the initial luciferin as \( l_0 \), initial decision domains range as \( Rd_0 \), circular sensor range as \( R_1 \), neighbour number as \( nei \), moving step-length as \( step \) and maximum iteration as \( iter \_max \).

**Step 2[6].** Calculate the fitness value. On the axes of two directions, place the each glowworm individual between the left and right endpoints of integral interval, and sort according to the position. On the \( x \) axis direction, the first individual of initial population is corresponding to the first individual on another axis direction, other and so on. Each glowworm divides the integral interval of \( x \) axis into \( S + 1 \) sections and \( S + 2 \) nodes, meanwhile, the glowworm divides the integral interval of \( y \) axis into \( D + 1 \) sections and \( D + 2 \) nodes. So the domain of integration is divided into \((S + 1) \times (D + 1)\) small rectangles. Separately calculate the distance \( d_i \) \((d_j)(i = 1, 2, \cdots, S + 1, j = 1, 2, \cdots, D + 1)\) between two adjacent nodes of the \( S + 2 (D + 2) \) nodes on \( x(y) \) axis, meanwhile, get the area \( (\text{area}_{ij} = d_i \times d_j) \) of each small rectangle. Then, calculate the corresponding function values of four vertexes and midpoint of each small rectangle. Finally, find the minimum and maximum among the five function values. The minimum of function is \( \text{Min}_{ij} \) and the maximum is \( \text{Max}_{ij} \). So the fitness of each glowworm individual is defined as follows:

\[
f(n) = 0.5 \cdot \sum_{i=1}^{S+1} \sum_{j=1}^{D+1} \text{area}_{ij} \cdot |\text{Max}_{ij} - \text{Min}_{ij}|
\]

Where, \( n = 1, 2, \cdots, N \). The more approximate zero the individual fitness value is, the more excellent the individual is. Set the maximum iterations as the termination condition.

**Step 3.** Judge termination condition. If the condition meets the termination condition, the algorithm ends and outputs optimal solutions. Otherwise, turns to step5.

**Step 4.** According to the artificial GSO algorithm, update the populations and calculate the fitness value of new individual. For glowworm \( i \) on the \( x \) axis direction, the update formulas of luciferin value and position are as follows:

\[
x_{is}(t + 1) = x_{is}(t) + step \cdot ((x_{is}(t) - x_{is}(t)) / \|x_{is}(t) - x_{is}(t)\|)
\]

For glowworm \( i \) on the \( y \) axis direction, the update formulas of luciferin value and position are as follows:

\[
y_{id}(t + 1) = y_{id}(t) + step \cdot ((y_{id}(t) - y_{id}(t)) / \|y_{id}(t) - y_{id}(t)\|)
\]

where, \( s = 1, 2, \cdots, S + 1, \ d = 1, 2, \cdots, D + 1, \ step \), \( \gamma \) and \( \rho \) are nonnegative.

**Step 5.** Repeat step 4 until it reaches the termination condition and choose the best individual as the result.

**Step 6[6].** End the algorithm. The integral value is approximately calculated as follows:

\[
I = \left( \sum_{i=1}^{S+1} \sum_{j=1}^{D+1} (g_{i,j} + g_{i+1,j} + g_{i,j+1} + g_{i+1,j+1} + g_{mid}) \cdot \text{area}_{ij} \right) / 5
\]
where, $g_{ij}$, $g_{i+1,j}$, $g_{i,j+1}$, $g_{i+1,j+1}$ and $g_{mid}$ are the corresponding function values of four vertexes and midpoint of the small rectangle. $area_{ij}$ is the area of the small rectangle.

4. Numerical simulation experiments

In order to verify the feasibility and validity of the algorithm, code the algorithm in Matlab8.0 and calculate examples in references [1] and [6]. Then compare the results with the ones calculated with traditional Composite Trapezoid method and Composite Simpson method. The computer configuration is Celeron(R) Dual-Core, 1.8 GHz and 1GB memory. Main parameters of algorithm are set as follows:

Set the initial luciferin $l_0 = 5$, moving step-length $step = 0.03$, initial decision domains range $Rd_0 = 1$, circular sensor range $R_c = 1$, neighbour number $nei = 5$, $\rho = 0.4$, $\beta = 0.08$; $\gamma = 0.6$ and maximum iteration $iter_{max} = 50$. The tests are run independently for 30 times.

**Example 1**[1]: Calculate the integral $I_1 = \int_0^1 dx \int_0^{x^2} dy = 2 \ln 2 - \ln 3$

The accurate value is 0.28768210 and the swarm size $N$ is fixed to 50 in this paper. The comparison results are shown in table 1. The results show that the GSO algorithm can better calculate the integral value, and the error is small.

| Segmentation points | Integration method | Integral value | Relative error |
|---------------------|--------------------|----------------|----------------|
| 16                  | GSO                | 0.287645538034047 | 3.6516965927478e-005 |
|                     | Composite Trapezoid method | 0.2880803 | 0.0013843 |
|                     | PSO                | 0.2879031  | 0.0007681 |
| 64                  | GSO                | 0.287596879612012 | 8.522038798797205e-005 |
|                     | Composite Trapezoid method | 0.2877084 | 0.0000917 |
|                     | PSO                | 0.2876986  | 0.0000572 |

The figure 1 is curve graph of independent running times and the best integral values with 64 segments on two axes. The graph shows that almost each optimal integral value is close to the accurate value in 30 times experiments, that is to say, the algorithm is effective and feasible.
Example 2[1]: Solve the area \( I_2 = \int_D dx dy = \int_a^b du \int_c^d \frac{u}{(1+y)^2} dv = \frac{(b-a)(d^2-c^2)}{2(1+a)(1+b)} \) of a closed region \( D \), which is consist of several lines such as \( x + y = c \), \( x + y = d \), \( y = ax \) and \( y = bx \) \((0 < c < d, 0 < a < b)\).

The accurate value is 0.2500000 and the swarm size \( N \) is fixed to 50. \( a,b,c \) and \( d \) is fixed to 1.2,1 and 2, respectively. The comparison results are shown in table 2. The results show that GSO algorithm can obtain the integral approximation and the error is small, but not as good as the value in reference.

The figure 2 is curve graph of independent running times and the best integral values with 64 segments on two axes. The graph shows that almost each optimal integral value is close to the accurate value in 30 times experiments, that is to say, the algorithm is effective and feasible.

Table 2. The comparison results of three algorithms

| Segmentation points | Integration method | Integral value | Relative error |
|---------------------|--------------------|----------------|---------------|
| 32                  | GSO                | 0.251378612311633 | 4.018547724479737e-004 |
|                     | Composite Trapezoid method | 0.2500214 | 0.0000857 |
|                     | PSO                | 0.2500138 | 0.0000553 |
| 64                  | GSO                | 0.250401854772448 | 4.018547724483623e-004 |
|                     | Composite Trapezoid method | 0.2500055 | 0.0000222 |
|                     | PSO                | 0.2500036 | 0.0000145 |

Example 3[1]: Calculate the integral \( I_3 = \int_0^d dx \int_0^e e^{-(x+y)} dy = (1 - e^{-1}) \int_0^d e^{-x} dx \)

The accurate value is 0.4720828, and the antiderivative of the integrand is not exist. The swarm size \( N \) is fixed to 50 in this paper. The comparison results are shown in table 3. The results show that GSO algorithm can obtain the integral approximation and the error is small, but not as good as the value in reference, meanwhile, the results show that the more segmentation points in GSO algorithm, the closer the integral value is to the accurate value, and the smaller the error is.

Table 3. The comparison results of three algorithms

| Segmentation points | Integration method | Integral value | Relative error |
|---------------------|--------------------|----------------|---------------|
| 2                   | GSO                | 0.373586386532772 | 0.098496413467228 |
|                     | Composite Trapezoid method | 0.4719058 | 0.0003749 |
|                     | PSO                | 0.4720343 | 0.0001028 |
| 4                   | GSO                | 0.400106665084095 | 0.071976134915905 |
|                     | Composite Trapezoid method | 0.4720991 | 0.0000345 |
|                     | PSO                | 0.4720882 | 0.0000114 |
| 8                   | Composite Trapezoid method | 0.428222136526588 | 0.043860663473412 |
|                     | PSO                | 0.4720907 | 0.0000167 |
| 32                  | GSO                | 0.461386526750600 | 0.010696273249400 |
|                     | Composite Trapezoid method | 0.4720837 | 0.0000020 |
|                     | PSO                | 0.4720832 | 0.0000020 |
The figure 3 is curve graph of independent running times and the best integral values with 32 segments on two axes. The graph shows that almost each optimal integral value is close to the accurate value in 30 times experiments, that is to say, the algorithm is effective and feasible.

Example 4[1]: Calculate the integral \( I_4 = \int_{0.01}^{0.99} \frac{1}{x} dy + \int_{0}^{1} (y + \frac{1}{x}) dy = \frac{0.99}{2} + 2 \ln 10 \)

The accurate value is 5.1001700 and the integrand is meaningless at the origin. The swarm size N is fixed to 100 in this paper. The comparison results are shown in table 4. The results show that GSO algorithm can obtain a higher precision and smaller error value when the numbers of segments both are 64 on two axes.

Table 4. The comparison results of three algorithms

| Segmentation points | Integration method   | Integral value | Relative error               |
|---------------------|----------------------|----------------|----------------------------|
| 64                  | GSO                  | 5.099686404664276 | 4.835953357247291e-004     |
|                     | Composite Trapezoid method | ——           | ——                         |
|                     | PSO                  | ——            | ——                         |
| 256                 | GSO                  | 5.097627708319678 | 0.002542291680323          |
|                     | Composite Trapezoid method | 5.1123580 | 0.0023897                  |
|                     | PSO                  | 5.1048279     | 0.0009132                  |

The figure 4 is curve graph of independent running times and the best integral values with 64 segments on two axes. The graph shows that almost each optimal integral value is close to the accurate value in 30 times experiments, that is to say, the algorithm is effective and feasible.

Example 5[6]: Calculate the integral \( I_5 = \int_{1}^{2} \int_{0}^{1.5} \ln(x + 2y) dy dx \)

The accurate value is 0.42955452600 and the swarm size N is fixed to 100 in this paper. When the number of segments is 100 on both axes, the comparison results are shown in table 5. The results show that GSO algorithm can obtain a higher precision and smaller error value.

Table 5. The comparison results of three algorithms

| Segmentation points | Integration method   | Integral value | Relative error               |
|---------------------|----------------------|----------------|----------------------------|
| 100                 | GSO                  | 0.429557437360488 | 2.911360487800607e-006     |
|                     | Composite Simpson method | 0.4295524387 | 0.00000486                  |
|                     | PSO                  | 0.42955439267315 | 0.00000031                  |
The figure 5 is curve graph of independent running times and the best integral values. The graph shows that almost each optimal integral value is close to the accurate value in 30 times experiments, that is to say, the algorithm is effective and feasible.

Figure 5. The curve graph of independent running times and the best integral values

5. Conclusions
In this paper, we put forward a new method for solving double numerical integration based on artificial glowworm swarm optimization algorithm. GSO is used to optimize the node points on the both direction range in rectangular Integration domain to get a more precise Integration result. Simulation examples show the algorithm is effective and feasible and it is a validated method with high precision and powerful self-adapting. The algorithm has value in numerical calculation and engineering practice.

Acknowledgements
This research was supported by the Key platform and Major Program of Guangdong Province under Grant No.2018KQNCX392 and by South China Institute of Software Engineering of Guangzhou University under Grant No. ky201823.

References
[1] ZHU Lei. About Gauss2Per Kai integral approximation [J]. JOURNAL OF HEFEI UNIVERSITY OF TECHNOLOGY, 2007,30(5):655-656.
[2] ZHENG Hua-sheng, TANG Jing-lun, WEI Di. Constructions of High Accurate Numerical Integration Formula and Its Applications [J]. Mathematics in Practice and Theory,2007,37(15):141-148.
[3] CHEN Fu-long. Two-dimensional numerical value integral’s computing method [J]. Computer Engineering and Applications,2007,43(19):32-34.
[4] PENG Xi-yuan, PENG Yu, DAI Yu-feng. Swarm Intelligence Theory and Applications [J],ACTA ELECTRONICA SINICA, Vol.31 No.12A Dec.2003:1982-1988.
[5] ZHOU Yong-Quan, ZHANG Ming, ZHAO Bin. Solving Numerical Integration Based on Evolution Strategy Method [J]. CHINESE JOURNAL OF COMPUTERS, Vol.31, No. 2, Feb. 2008:196-206.
[6] WEI Xing-qiong, ZHOU Yong-quan. Double integral research based on PSO [J]. Computer Engineering and Design, 2009,30(20):4705-4707.
[7] NIE Li-ming, ZHOU Yong-quan. Using artificial fish-swarm algorithm for solving two-dimensional numerical integration [J]. Computer Engineering and Applications,2009,45(10):34-37.
[8] K.N. Krishnanand and D. Ghose. Glowworm Swarm Optimisation: A New Method For
Optimising Multi-modal Functions[J]. Int. J. Computational Intelligence Studies, Vol. 1, No. 1, 2009, pp. 93-119.