A new golden ratio local search based particle swarm optimization

Yanxia Sun, Barend Jacobus van Wyk
Department of Electrical Engineering
Tshwane University of Technology
Pretoria, South Africa

Zenghui Wang
Department of Electrical and Mining Engineering
University of South Africa
Florida, South Africa

Abstract—At beginning of the search process of particle swarm optimization, one of the disadvantages is that PSO focuses on the global search while the local search is weaken. However, at the end of the search procedure, the PSO focuses on the local search as all most all the particles converge to small areas which may make the particle swarm trapped in the local minima if no particle find position near the minima at the beginning of search procedure. To improve the optimization performance, the local search is necessary for particle swarm optimization. In this paper, golden ratio is used to determine the size of the search area. Only two positions need to be checked to find whether there are local positions with lower fitness value around a certain particle position. This method is easy to use. It is also tested using several famous benchmarks with high dimensions and big search space to the efficiency of the proposed method.

Keywords-Golden ratio; Particle swarm optimization; Local search

I. INTRODUCTION

Many systems require solving one or more nonlinear optimization problems. While analytical methods might suffer from slow convergence, and the curse of dimensionality, heuristics-based swarm intelligence can be an efficient alternative. Particle swarm optimization (PSO), part of the swarm intelligence family, is known to effectively solve large-scale nonlinear optimization problems [1][2][3][4]. More and more researchers have been interested in PSO methods which have been investigated from various perspectives such as in artificial life, social psychology, engineering, computer science and so forth [5].

PSO is an evolutionary computation technique developed by Kennedy and Eberhart [6] in 1995, which was inspired by the social behavior of bird flocking and fish schooling. In PSO, each solution is a “bird” in the flock and is referred to as a “particle”. However, the original PSO algorithm [6] is easily trapped in local suboptimal points when applied to problems with many local extremes [7]. PSO has been developed for many years, one of the most famous one is the standard PSO which can be looked as a benchmark PSO to compare with other proposed variants of PSO [8]. The standard PSO has been developed for several versions, that is, SPSO 2006, 2007 and 2011 since the year of 2006. However, similar with most of the variants of PSO, the standard PSOs have a disadvantage that PSO focuses on the global search while the local search is weaken during the beginning of the search process of particle swarm optimization. And during the end of the search procedure, the standard PSO focus on the local search as all most all the particles converge to small areas which may make the particle swarm trapped in the local minima if no particle find position near the minima at the beginning of search procedure. To conquer or weaken this disadvantage, a simple local search technique is proposed in this paper.

The rest of this paper is structured as follows. Section 2 describes the preliminary knowledge about particle swarm optimization. The details of golden ratio local search technique are presented in Section 3. Section 4 gives a description of the experimental settings for the benchmarks and simulation results in comparison with the SPSO 2011. Finally, Section 5 gives some concluding remarks.

II. A BRIEF DESCRIPTION OF PSO

In general, single objective optimisation problems can be represented as the following optimisation problem [3]:

\[
\begin{align*}
    f(X_i) &= \min(g(X_i)), X_i = [x_1^i, x_2^i, \ldots, x_n^i], \\
    \text{subject to } p_j(X_i) &\leq 0, j = 1, 2, \ldots, k.
\end{align*}
\]

Here \(g(\cdot)\) is the objective function without constraints; \(X_i(t)\) denotes the position vector of particle \(i\) consisting of \(n\) variables. Every candidate solution of \(f(X_i)\) is called a “particle”.

The standard particle swarm algorithm works by iteratively searching in a region and is concerned with the best previous success of each particle, the best previous success of the particle swarm, the current position and the velocity of each particle [3]. The particle searches the domain of the problem, according to

\[
    V_i(t+1) = \alpha V_i(t) + c_1R_i(P_i - X_i(t)) + c_2R_i(P_g - X_i(t)),
\]

\[
    X_i(t+1) = X_i(t) + V_i(t+1),
\]

where \(V_i = [v_1^i, v_2^i, \ldots, v_n^i]\) is the velocity of particle \(i\); \(X_i = [x_1^i, x_2^i, \ldots, x_n^i]\) represents the position of particle \(i\); \(P_i\) represents the best previous position of particle \(i\) (indicating the best discoveries or previous experience of particle \(i\)); \(P_g\)
represents the best previous position among all particles (indicating the best discovery or previous experience of the social swarm); \( \omega \) is the inertia weight that controls the impact of the previous velocity of the particle on its current velocity and is sometimes adaptive [9]; \( R_1 \) and \( R_2 \) are two random weights whose components \( r_1^j \) and \( r_2^j \) \((j = 1, 2, \ldots, n)\) are chosen uniformly within the interval \([0,1]\) which might not guarantee the convergence of the particle trajectory; \( c_1 \) and \( c_2 \) are the positive constant parameters. Generally the value of each component in \( V_j \) should be clamped to the range \([-v_{\text{max}}, v_{\text{max}}]\) to control excessive roaming of particles outside the search space.

There are some differences between the original PSO and standard PSO [8] about the parameter setting. For standard PSO 2011, the parameters are set as

\[
\omega = \frac{1}{2 \times \log(2)},
\]

\[
c_1 = 0.5 + \log(2),
\]

\[
c_2 = c_1
\]

and the swarm size is \( \text{floor}(10 + 2\sqrt{D}) \).

Here, \( \log() \) is the natural logarithm function, and \( \text{floor()} \) is the round toward negative infinity function. Moreover, the random topology is used in SPSO 2011.

III. GOLDEN RATIO LOCAL SEARCH BASED PSO

As mentioned in Section I, PSO cannot achieve a good local research performance at beginning of search process which results in the particles trap into local minima especially for the optimization problems with a big problem space and a large number of minima. It is possible to improve the optimization performance if some local searching techniques are used to search the local area around the current particle positions during the whole searching process.

According to (2) and (3), PSO works by iteratively searching in a region which has the tendency to the best previous success of each particle and the best previous success of the particle swarm. Hence, it is more possible that the fitness value related to the current position is lower than the random search, and it can be expected that two neighbors, which are different from the neighbor in the topology, are located in the same 'cove' in the searching space. There are potential positions between the closest particle neighbors which have lower fitness value than these two particles. Here the golden ratio is used to determine the local search position as golden ratio has been widely used or shown in many areas, such as architecture [10][11], aesthetics [12], music [13], industrial design [14], and so on. Moreover, the golden ratio is also used in optimization [15] and achieves good efficiency. Here, only one position is chosen based on the golden ratio between the two neighbors as shown in Fig. 1.

Figure 1 Local position determined by golden ratio

In Fig. 1 A and B are two particles. To realize the local search for particle A, the position C is determined by the nearest neighbor B and golden ratio. Here,

\[
\frac{AC}{CB} = 0.618. \quad (4)
\]

Here, \( |\*| \) is the vector norm function. To extend the search space, the opposition based method is used. The opposition concept has been used in evolution optimization algorithm and a good optimization performance was obtained [16]. In this paper, the method of using opposition concept is rotating C with 180°.

Figure 2 Local positions with opposition

To simplify the complexity of the proposed method, only two extra positions, which are C and C', are checked. To realize the proposed method, the following procedure can be used to implement the golden ratio local search based PSO:

1. Initialize the parameters of PSO and swarm by assigning a random position in the problem hyperspace to each particle.
2. Evaluate the fitness function for each particle and set the fitness values as the best personal experience.
3. Randomize the topology if there is no improvement in the best swarm solution.
4. Update the positions of all the particles using (2)-(3).
5. Check for constraint violations. If the position is out of the range, the position is set to the border value and velocity is set to zero.
6. Evaluate the fitness function for each particle.
7. Golden ratio based local search for every particle. Find the best solution among position A, C and C’ and take it as the current position.
8. Repeat steps 3)-7) until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

To reduce the effect of different dimensions with different ranges, the optimisation problem variable interval must therefore be mapped to \([-1, 1]\) and vice versa using

\[
x_j = -1 + \frac{2(x_j - a_j)}{b_j - a_j}, \quad j = 1, 2, \ldots, n. \quad (5)
\]

and

\[
x_j = a_j + \frac{1}{2}(x_j + 1)(b_j - a_j), \quad j = 1, 2, \ldots, n. \quad (6)
\]
Here, \( a_j \) and \( b_j \) represent the lower boundary and the upper boundary of \( x_i(t) \), respectively only one particle is analysed for simplicity.

IV. EXPERIMENTAL VARIFICATION

To demonstrate the efficiency of the proposed technique, several famous benchmarks are used to compare the proposed method and SPSO 2011 (Matlab version) [8]. The parameters are set as SPSO 2011 and the particle swarm was run for 50 trials per function with a maximum iteration of 1000 for every trial.

The benchmarks are listed in Table 1. Five famous benchmark functions are chosen and every one will be tested for two different dimensions (30 and 100).

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Formulation</th>
<th>Dim</th>
<th>Initial Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere function</td>
<td>( f_i(x) = \sum_{i=1}^{n} x_i^2 )</td>
<td>10</td>
<td>±500</td>
</tr>
<tr>
<td>Rastrigin function</td>
<td>( f_i(x) = \sum_{i=1}^{n} \left[ x_i^2 - 10 \cos(2\pi x_i) + 10 \right] )</td>
<td>10</td>
<td>±500</td>
</tr>
<tr>
<td>Rosenbrock function</td>
<td>( f_i(x) = \sum_{i=1}^{n} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] )</td>
<td>10</td>
<td>±500</td>
</tr>
<tr>
<td>Griewank function</td>
<td>( f_i(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 + \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 )</td>
<td>10</td>
<td>±500</td>
</tr>
<tr>
<td>Ackley function</td>
<td>( f_i(x) = -20 \exp(-0.2 \sqrt{\sum_{i=1}^{n} x_i^2}) - \exp(\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e )</td>
<td>10</td>
<td>±500</td>
</tr>
</tbody>
</table>

The benchmarks are listed in Table 1. The proposed method was run for 50 trials per function with a maximum iteration of 1000 for every trial.

The benchmarks are listed in Table 1. Five famous benchmark functions are chosen and every one will be tested for two different dimensions (30 and 100).

As can be seen from Table 2, the proposed method can greatly improve the optimization performance and the stability of optimization when the dimensions are 10 and 30. However, there is a little improvement when the dimension is 100, which was caused by the improvement of the complication of the optimization problems. It also means that more local searches should be done when the number of dimension is big.

V. CONCLUSION

This paper proposed a golden ratio local search based particle swarm optimization. This method also used the opposition concept to improve the local search ability. This method was applied to several famous benchmarks and achieved a good optimization results especially the dimension of the searching space is not very big. Furthermore, this proposed technique can be easily embedded inside other evolutionary optimization algorithms. In the future, we will focus on improving the optimization performance when the dimension is big using some local searching strategies.

ACKNOWLEDGMENT

The authors thank the Matlab version codes of SPSO 2011 by Mahamed Omran on the particle swarm central website [8].

REFERENCES

TABLE II. RESULTS USING THE STANDARD PSO 2011 AND THE PROPOSED METHOD

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>best</th>
<th>Mean</th>
<th>Worst</th>
<th>Std.dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere function f₁</td>
<td>Standard PSO 2011</td>
<td>0.2020</td>
<td>0.5903</td>
<td>1.2049</td>
<td>0.2199</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>4.7832×10⁻⁴</td>
<td>0.0142</td>
<td>0.0476</td>
<td>0.0123</td>
</tr>
<tr>
<td>Rastrigin function f₂</td>
<td>Standard PSO 2011</td>
<td>17.0480</td>
<td>40.4833</td>
<td>56.0410</td>
<td>9.2735</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>8.8062</td>
<td>29.4936</td>
<td>50.4781</td>
<td>10.4372</td>
</tr>
<tr>
<td>Rosenbrock function f₃</td>
<td>Standard PSO 2011</td>
<td>23.0128</td>
<td>56.3832</td>
<td>108.4931</td>
<td>16.9555</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>2.6188</td>
<td>9.0025</td>
<td>18.9887</td>
<td>3.4451</td>
</tr>
<tr>
<td>Griewank function f₄</td>
<td>Standard PSO 2011</td>
<td>0.0086</td>
<td>0.0684</td>
<td>0.1562</td>
<td>0.0324</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>4.2879×10⁻⁴</td>
<td>0.0041</td>
<td>0.0419</td>
<td>0.0061</td>
</tr>
<tr>
<td>Ackely function f₅</td>
<td>Standard PSO 2011</td>
<td>1.4301</td>
<td>2.3176</td>
<td>3.0554</td>
<td>0.3911</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>0.0364</td>
<td>0.3413</td>
<td>1.2653</td>
<td>0.2762</td>
</tr>
<tr>
<td>Sphere function f₆</td>
<td>Standard PSO 2011</td>
<td>3.0458</td>
<td>6.9523</td>
<td>8.7257</td>
<td>0.9279</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>1.9123</td>
<td>2.7449</td>
<td>3.9559</td>
<td>0.4840</td>
</tr>
<tr>
<td>Rastrigin function f₇</td>
<td>Standard PSO 2011</td>
<td>199.1225</td>
<td>233.9243</td>
<td>266.6047</td>
<td>15.7870</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>109.5946</td>
<td>185.5221</td>
<td>229.6229</td>
<td>24.9829</td>
</tr>
<tr>
<td>Rosenbrock function f₈</td>
<td>Standard PSO 2011</td>
<td>260.4547</td>
<td>377.9755</td>
<td>461.0138</td>
<td>41.7120</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>175.9785</td>
<td>218.4976</td>
<td>259.2466</td>
<td>21.8027</td>
</tr>
<tr>
<td>Griewank function f₉</td>
<td>Standard PSO 2011</td>
<td>0.1862</td>
<td>0.3066</td>
<td>0.3923</td>
<td>0.0434</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>0.1625</td>
<td>0.2744</td>
<td>0.3959</td>
<td>0.0484</td>
</tr>
<tr>
<td>Ackely function f₁₀</td>
<td>Standard PSO 2011</td>
<td>3.1866</td>
<td>3.5137</td>
<td>3.7619</td>
<td>0.1336</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>2.2784</td>
<td>2.8398</td>
<td>3.2952</td>
<td>0.2273</td>
</tr>
<tr>
<td>Sphere function f₁₁</td>
<td>Standard PSO 2011</td>
<td>23.8052</td>
<td>27.2318</td>
<td>29.4010</td>
<td>1.2801</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>23.0614</td>
<td>27.2534</td>
<td>29.1615</td>
<td>1.2253</td>
</tr>
<tr>
<td>Rastrigin function f₁₂</td>
<td>Standard PSO 2011</td>
<td>823.2026</td>
<td>890.9073</td>
<td>947.3594</td>
<td>23.7985</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>792.0004</td>
<td>881.0822</td>
<td>934.9773</td>
<td>35.2341</td>
</tr>
<tr>
<td>Rosenbrock function f₁₃</td>
<td>Standard PSO 2011</td>
<td>1340</td>
<td>1606</td>
<td>1828</td>
<td>99.9118</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>1378</td>
<td>1602</td>
<td>1763</td>
<td>90.2874</td>
</tr>
<tr>
<td>Griewank function f₁₄</td>
<td>Standard PSO 2011</td>
<td>0.3364</td>
<td>0.4229</td>
<td>0.5061</td>
<td>0.0326</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>0.3195</td>
<td>0.4242</td>
<td>0.4992</td>
<td>0.0303</td>
</tr>
<tr>
<td>Ackely function f₁₅</td>
<td>Standard PSO 2011</td>
<td>3.5200</td>
<td>3.6831</td>
<td>3.7948</td>
<td>0.0579</td>
</tr>
<tr>
<td></td>
<td>Proposed method</td>
<td>3.5148</td>
<td>3.6709</td>
<td>3.7664</td>
<td>0.0551</td>
</tr>
</tbody>
</table>