Hybrid Particle Swarm Algorithm with Levy Mutation

Jyoti Garg¹, Ashutosh Gupta²

¹Department of Electronics and Communication Engineering, ABES Engineering College Ghaziabad-201009, INDIA
²Central Electronics Limited, Ghaziabad, INDIA

Abstract—The standard Particle Swarm Optimization (PSO) studies its own previous best solution and the group’s previous best to optimize problems. One problem exists in PSO is its tendency of trapping into local optimum. To avoid this problem, in this paper a new PSO (LHPSO) is presented by combining a Lévy mutation based on Lévy Distribution on the best particle so that the mutated best particle could lead all the rest of particles to the better positions. The idea to introduce Lévy Mutation operator is to increase the probability of a particle escaping from a local optimum. As it is known the variance of Lévy distribution is infinite, so that Lévy mutation could make a particle to have a long jump. LHPSO has been compared with Standard PSO on a set of benchmark functions. The results show that proposed idea is an effective and efficient idea.

Keywords: PSO, Lévy Probability Distribution, Mutation, Exploration, Exploitation.

I. INTRODUCTION

The Particle Swarm Optimization (PSO) was firstly introduced by Kennedy and Eberhart in 1995 [1]. It is a simple evolutionary algorithm which differs from other evolutionary algorithms in which it is motivated from the simulation of social behavior. PSO has shown good performance in finding better solutions to optimize problems and turned out to be another powerful tool besides other evolutionary algorithms such as genetic algorithms [2]. Clerc and Kennedy [11] introduced a constriction factor based on dynamic systems theory in order to guarantee the stability of the swarm, which may be necessary to ensure convergence of the PSO algorithm and also some guidelines for choosing the constriction and weighting coefficients. In a similar way, van den Bergh [4] has also presented some results in this direction. However, a deterministic approach might not be straightforward generalized for the analysis of trajectory of stochastic particles in multi-dimensional search space. Particle Swarm Optimization is an evolutionary algorithm to find optimal regions of search spaces through the interaction of individuals in a population of particles [3]. Each particle is assigned a random velocity and repeatedly moved through search space. Every particle is attracted towards a location of best fitness achieved so far by particle itself and by location of best fitness achieved so far across whole population [1]. The analysis on simple PSO is presented in [5]. One problem found in the standard PSO is that it could easily fall into local optima in many optimization problems. Recently some research has been done in this direction [6]. One reason for PSO to converge to local optima is that particles in PSO can quickly converge to the best position once the best position has no change in a local optimum. When all particles become similar, there is little hope to find a better position to replace the previous best position found so far. Another reason is the lack of population diversity in PSO algorithms is understood to be a factor in their convergence on local optima [3]. Therefore, the addition of a mutation operator to PSO should enhance its global search capacity and thus improve its performance. There are several instances in PSO where mutation is introduced in the swarm. Some mutation operators that have been applied to mutate the position vector in PSO include Gaussian Mutation [8], Cauchy [9], Chaos mutation [10] etc.

This paper presents a new PSO algorithm based on Lévy mutation with the help of Lévy distribution on the best particle so that the mutated best particle could lead all the rest of particles to the global positions. As it is known the variance of Lévy distribution is infinite, so that it could make a particle to have a long jump. Results have been tested on standard benchmark functions. The rest of the paper is organized as follows: Section II describes the original Particle Swarm Optimization algorithm. Section III describes Lévy distribution. In Section IV the proposed LHSPO algorithm is discussed; section V deals with the results and discussion. Finally, this paper concludes with section VI.

II. PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization is motivated from the social behavior of bird flocking. Particles try to get their best position in search space by their best position as well as following their neighbor’s best position. For a D-dimensional search space, the position of the ith particle is represented as 

\[ X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD}) \]

Each Particle maintains a memory of its previous best position.
The best one among all the particles in the population is represented as global best $P_G = (p_{g1}, p_{g2}, p_{g3}, \ldots, p_{gD})$. The velocity of each particle is represented $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$. During each iteration each particle is accelerated toward its previous best position and the neighbor’s global best position. At each iteration the velocity of the particle is updated according to equation (1) that depends on the current velocity, and global position. The next position of the particle is updated according to equation (2) using new velocity and previous position in the search space. This process is iterated anumber of times until best global position is achieved.

$$v_{iD} = wv_{iD} + c_1r_1(P_{gD} - x_{iD}) + c_2r_2(P_{pD} - x_{iD})$$

$$x_{iD} = x_{iD} + v_{iD}$$  \hspace{1cm} (1)

In Equation(1) $w$ is inertia weight, $c_1$ is cognitive acceleration constant and $c_2$ is social acceleration constants. $x_{pD}$ is the position vector of ith particle at that particular iteration and $P_{pD}$ is personal best of particle i. $P_{gD}$ is the global best of all particles at that particular iteration. The second part tells us that any particle that is distant from its cognitive best will accelerate toward it more strongly than it was nearby. Third part represents any particle which is distant from the social best will accelerate toward it more strongly than a particle nearby. The constants $r_1,r_2$ are the uniformly generated random numbers in the range of $[0, 1]$. Equation (2) position of the particle is updated according to the velocity as stated in equation (1).

I. LEVY DISTRIBUTION

For a Levy Process Probability density function is defined as

$$L_{ \alpha, \gamma}(z) = \frac{1}{\pi} \int_0^\infty \exp(-\gamma q^\alpha) \cos(qz) dq \hspace{1cm} , z \in \mathbb{R}$$  \hspace{1cm} (3)

Where $\alpha$ and $\gamma$ both are distribution parameters. $\gamma$ defines the scaling of process and $\alpha$ defines the index of distribution and controls the scale properties of stochastic process $\{z\}$ [7]. Above defined pdf is symmetric w.r.t. $z=0$, $\gamma$ is taken 1 for simplicity. If the series expansion of equation (1) is considered, the approximation of levy stable process is taken as [7].

$$L_{ \alpha, \gamma}(z) \approx \frac{C_{LS}(\alpha)}{z^{1+\gamma\alpha}}$$  \hspace{1cm} (4)

As it is known that analytical form of equation (4) is not known for general $\alpha$ then a random number in Levy distribution is generated in this way [7]. First step is two normal stochastic variables $p,q$ having standard deviation $\sigma_p,\sigma_q$ are taken as shown in equation (5)

$$v = \frac{p}{|q|^\alpha}$$  \hspace{1cm} (5)

By using probability theory, probability distribution of stochastic processes can be found out. Probability density has an important property the asymptotic approximation that is used for large arguments [7]. As in this distribution $\sigma_p,\sigma_q$ Standard deviation can’t be chosen independently for an arbitrary value of $\alpha$. So for convenience $\sigma_q=1$ is assumed and accordingly $\sigma_p$ is calculated. If $n$ Number of different variables are having similar probability distribution are taken according to equation (6).

$$z_n = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} v_k$$  \hspace{1cm} (6)

In equation (6) value of $n^{1/\alpha}$ should be considered as it is characterizing variable $z_n$. $\gamma$. But the convergence of equation (6) is very slow. To improve this, non linear transformation function is used as shown in equation (7).

$$W = \{[K(\alpha) - 1] \exp(- \frac{v}{C(\alpha)}) + 1\}v$$  \hspace{1cm} (7)

If proper value of $K(\alpha),C(\alpha)$ is used, equation (7) shows a faster convergence. As $K(\alpha),C(\alpha)$ depend on $\alpha$ and their values can be calculated in [7]. For efficient algorithm linear transformation can be used as shown in equation (8).

$$z_s = \frac{1}{\gamma} \gamma Z$$  \hspace{1cm} (8)

IV. PROPOSED IDEA (LEVY MUTATION WITH PSO)

Lévy Mutation Particle Swarm Optimization is simple and modified version of Particle Swarm Optimization algorithm, as it uses Levy distribution to mutate the best particle’s position. Sometimes particles don’t explore them by assuming that their neighbor particle is global best particle and start to converge around that neighbor. To avoid this premature convergence, levy mutation is applied on position of best particle so that particles don’t trap in local optima. In the proposed algorithm, initial position and velocity is calculated according to equation (1) and (2). After that, levy mutation operator is used as...
\[ H(j) = \frac{\sum_{i=1}^{\text{popsize}} v[j][i]}{\text{popsize}} \]  
(9)

Where \(v[j][i]\) is the \(i\)th velocity vector of \(j\)th particle in the population, \(\text{popsize}\) is the population size. \(H(j)\) is a weight vector within \([-H_{\text{max}}, H_{\text{min}}]\) and \(H_{\text{max}}\) is set to 1. Then this operator with Levy Distribution is applied on best particle as shown in equation (10).

\[ P_{\text{g}^\text{best}} = P_g + H(j) \star L_j(\alpha) \]  
(10)

Where \(L_j(\alpha)\) represents Levy Distribution. Computational Steps of LHPSO are shown in Flow graph.

V. RESULT AND DISCUSSION

For comparison between PSO and LHPSO, four Standard benchmark functions have been used as shown in Table I. Function \(f_1 f_2, f_3\) are unimodal functions while \(f_4\) is multimodal function. In Experiments dimension of functions are taken 30. Parameters \(c_1 = c_2 = 1.496\) and \(w = 0.7298\) is considered. Population size is taken 20 and generations are taken 100. Comparisons between PSO and LHPSO are shown in Table II.

![Fitness of function \(f_1\) in PSO and LHPSO](image1)

![Fitness of function \(f_2\) in PSO and LHPSO](image2)

Figure 2, 3 shows that LHPSO gives better results than StandardPSO.

### Table I. Benchmark Test Functions Used for LHPSO

<table>
<thead>
<tr>
<th>Function</th>
<th>Dimension</th>
<th>Search Space</th>
<th>(f_{\text{mi}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f_1(x) = \sum_{i=1}^{n} x_i^2)</td>
<td>30</td>
<td>([-5.12, 5.12])</td>
<td>0</td>
</tr>
<tr>
<td>(f_2(x) = \sum_{i=1}^{n} \sum_{j=1}^{j} x_i x_j^2)</td>
<td>30</td>
<td>([-65536, 65536])</td>
<td>0</td>
</tr>
<tr>
<td>(f_3(x) = \sum_{i=1}^{n} (100x_i^2 - x_i^4 + 1))</td>
<td>30</td>
<td>([-30, 30])</td>
<td>0</td>
</tr>
<tr>
<td>(f_4(x) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i)))</td>
<td>30</td>
<td>([-5.12, 5.12])</td>
<td>0</td>
</tr>
</tbody>
</table>
TABLE II. BENCHMARK TEST FUNCTIONS USED FOR LHPSO

<table>
<thead>
<tr>
<th>Test Function</th>
<th>LHPSO Mean Best</th>
<th>LHPSO Standard Deviation</th>
<th>PSO Mean Best</th>
<th>PSO Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1(x)</td>
<td>1.65e-8</td>
<td>7.954e-8</td>
<td>1.79e-7</td>
<td>3.51e-7</td>
</tr>
<tr>
<td>F2(x)</td>
<td>0.165</td>
<td>0.1701</td>
<td>0.398</td>
<td>0.3082</td>
</tr>
<tr>
<td>F3(x)</td>
<td>0.959</td>
<td>1.4376</td>
<td>1.8016</td>
<td>2.8389</td>
</tr>
<tr>
<td>F4(x)</td>
<td>15.04</td>
<td>8.16</td>
<td>37.0721</td>
<td>9.7295</td>
</tr>
</tbody>
</table>

VI. CONCLUSION
In this paper Hybrid PSO is implemented with Lévy Mutation based on Levy Distribution. Due to this PSO will not get stuck into local optima. This also shows a trade off between Exploitation and Exploration. To show LHPSO gives better results than PSO standard benchmark functions have been used. Also in terms of fitness value, LHPSO is more efficient algorithm than Standard PSO.

VII. REFERENCES