On the Use of Particle Swarm Optimization with Multimodal Functions

Susana C. Esquivel
LIDIC (Laboratorio de Investigación
y Desarrollo en Inteligencia Computacional)
Universidad Nacional de San Luis
Ejército de los Andes 950,
San Luis 5700, ARGENTINA
esquivel@unsl.edu.ar

Carlos A. Coello Coello
CINVESTAV-IPN
Evolutionary Computation Group
Depto. de Ing. Elect./Sección de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D. F. 07300, MEXICO
ccoello@cs.cinvestav.mx

Abstract- In this paper, we present two hybrid particle swarm optimization (PSO) algorithms that incorporate a mutation operator similar to the one used with evolutionary algorithms. We study our hybridized PSO algorithm with two schemes called $g-best$ and $l-best$, and we apply them to multimodal functions. The proposed approaches are validated using test functions taken from the specialized literature, and our results are compared with respect to those obtained by other highly competitive PSO algorithms. Our comparative study indicates that the hybridization of PSO with a non-uniform mutation operator significantly improves its performance when dealing with multimodal functions.

1 Introduction

The algorithm now known as particle swarm optimization (PSO) was introduced by James Kennedy and Russell Eberhart in 1995 [8]. PSO is a population-based heuristic search technique in which each particle represents a potential solution within the search space and it is characterized by a position, a velocity and a record of its past performance. At each flight cycle, we evaluate the objective function for each particle with respect to its current position. The obtained value measures the quality of the particle (the equivalent to the fitness value used in evolutionary algorithms).

In the original PSO algorithm, particles "fly" through the search space influenced by two factors: a) the best position of the particle (as recorded in its history), and b) the best global position reached by any particle from the swarm (or population). The equations adopted are the following:

$$vel_{ij} = vel_{ij} + c_1 * r_1 * (p_{ij} - part_{id}) + c_2 * r_2 * (p_{gj} - part_{ij}) \quad (1)$$

$$part_{ij} = part_{ij} + vel_{ij} \quad (2)$$

where vel_{ij} is the velocity of particle i in the jth dimension, c_1 and c_2 are weights applied to the influences of the best positions obtained so far by particle i and to the best particle in the swarm g; r_1 and r_2 are random values (using a uniform distribution) in the interval $[0, 1]$. After the velocity is updated, the new position of the particle i is updated in its jth dimension using equation 2. This process is repeated for each dimension of the particle i and for all the particles in the swarm.

In further work, Shi and Eberhart [15] introduced the concept of *inertia factor*, w, whose goal is to control the amount of a particle's previous velocity that will be kept. In consequence, equation 1 is now expressed as:

$$vel_{ij} = w * vel_{ij} * c_1 * r_1 * (p_{ij} - part_{id}) + c_2 * r_2 * (p_{gj} - part_{ij}) \quad (3)$$

In related work, Kennedy and Eberhart [7] studied the effects of using neighborhoods in PSO. Further related work can be found in several other references (see for example [7, 6, 9, 13]).

In this paper, we are particularly interested in the topology called $lbest$, which includes in its neighborhood only to the nearest neighbors in terms of the particles' indices. In this model, equation 3 is expressed as follows:

$$vel_{ij} = w * velij + c_1 * r_1 * (p_{ij} - part_{ij}) + c_2 * r_2 * (p_{lj} - part_{ij}) \quad (4)$$

where p_{lj} represents to the best particle in the neighborhood of particle i, instead of the best global particle (i.e., with respect to the entire swarm).

Recently, other authors have proposed a hybridization of PSO with evolutionary algorithms concepts [5]. Some time before, Angeline had already proposed a hybrid swarm that uses the original PSO algorithm combined with the concept of selection [2].

The main goal of this paper is to evaluate the performance of two proposed PSO models (a global best and a local best model) when applied to multimodal optimization. For that sake, a well-known benchmark is adopted and results are compared with respect to other PSO-based approaches previously reported in the literature. The remainder of the paper is organized as follows. Section 2 describes the algorithm that implements our PSO global model using mutation. Section 3 describes the algorithm that implements our PSO local best model using mutation. Section 4 provides a description

of the test functions adopted for our study. The parameters fine-tuning and the experimental design are described in Section 5. Section 6 provides our results and their comparison with respect to previous PSO-based algorithms reported in the specialized literature. Finally, Section 7 states our conclusions and some possible paths for future research.

2 Global Best Model

Figure 1 shows the pseudo-code of the global PSO model adopted in this paper. This version of the algorithm is called "synchronous" because the evaluation to find the best global particle is done between iterations (the first time is done on line 9, just before entering the flight loop) and all the particles are updated before performing the next leader selection (line 26).

The mutation operator that we adopted is the *non-uniform* mutation proposed in [12]. If we assume a chromosome (under real-numbers representation) $s_v^t =< v_1, \ldots, v_m >$ (t is the generation number), and if the element to be mutated is v_k, then the new chromosome will be $s_v^{t+1} =< v_1, \ldots, v_k^{'}, \ldots, v_m >$, where:

$$v_k^{'} = \begin{cases} v_k + \Delta(t, UB - v_k) & \text{if a random digit is 0} \\ v_k - \Delta(t, v_k - LB) & \text{if a random digit is 1} \end{cases} \quad (5)$$

and LB and UB are the lower and upper bounds of the variable v_k. The function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that the probability of $\Delta(t, y)$ being close to 0 increases as t increases. The function suggested by Michalewicz (and adopted in our work) is [12]:

$$\Delta(t, y) = y \cdot \left(1 - r^{\left(1 - \frac{t}{T}\right)^b}\right) \quad (6)$$

where r is a random number in the range $[0..1]$, T is the total number of generations and b is a system parameter determining the degree of dependency on iteration number (we used $b = 5$, as suggested in [12]).

Note that in this work we argue that the use of a non-uniform mutation operator (such as the one previously described) combined with PSO introduces the diversity necessary to avoid getting trapped in local optima when dealing with multimodal functions. Thus, our hypothesis was that by using such a non-uniform mutation operator, even relatively simple topologies (such as the two models adopted in our work) would become highly competitive when dealing with multimodal functions.

3 Local Best Model

Figure 2 shows an asynchronous version of PSO with neighborhoods. In this case, when a particle needs to be updated, we look for the best particle within the neighborhood and such particle influences the velocity of the particle to be updated. It is important to note that, in this case, the neighbors that are to the left of the particle to be updated have already been updated, whereas those to its right are pending. The same mutation operator described in the previous section was adopted in this case.

The two models proposed in this paper (the global best and the local best) follow Carlisle's suggestions [4], whom indicated that it seems natural to work with a synchronous PSO when using the g_best model and with an asynchronous PSO when working with an l_best model.

4 Test Functions

To validate our approach, we chose several multimodal functions. The functions selected turn out to be difficult for any search algorithm because of their several local minima which can produce premature convergence. Note, however, that in all cases, a single global optimum exists. Diversity is a key issue to avoid premature convergence in our PSO approaches and mutation will be the operator responsible for maintaining such required diversity, regardless of the topology adopted (either the global or the local best models).

Note that the functions selected have been widely studied by PSO researchers [10, 16, 13, 5, 14, 1, 11] and, in general, provide a good mixture of test functions in terms of complexity. The easiest of these functions is Ackley's function and the most difficult is Schwefel's function. Their mathematical descriptions are provided next.

1. **Rastrigin's function**:

$$\mathcal{F}_1(\vec{x}) = nA + \sum_{i=1}^{n} x_i^2 - A\cos(wx_i) \quad (7)$$

where $A = 10$, $w = 2\pi$ and $\mathcal{F}_1(\vec{x}^*) = 0$ for $\vec{x}^* = (0, 0, ..., 0)$ is the global optimum.

2. **Griewank's function**:

$$\mathcal{F}_2(\vec{x}) = \frac{1}{4000} + \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) \quad (8)$$

where $\mathcal{F}_2(\vec{x}^*) = 0$ for $\vec{x}^* = (0, 0, ..., 0)$ is the global optimum.

3. **Ackley's function**:

$$\mathcal{F}_3(\vec{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e \quad (9)$$

where $\mathcal{F}_3(\vec{x}^*) = 0$ for $\vec{x}^* = (0, 0, ..., 0)$ is the global optimum.

```
Swarm Initialization
for i = 1 to number of particles do
    for j = 1 to number of dimensions do
        Initialize x_ij with a rnd(Xmin, Xmax) value
        Initialize v_ij with zero value
        copy x_ij in p_ij
    end
end
Search the best global leader and record its position in g
Swarm flight through the Search Space
do
    for i = 1 to number of particles do
        for j = 1 to number of dimensions do
            Update v_ij using p_ij and x_ij
            Prevent explosion of v_ij
            Update x_ij
            if (loop_number < total_loop * prob_mut)
              then Mutate x_ij
            fi
        end for
    Evaluate fitness(x_i)
    if fitness(p_i) < fitness(x_i)
        then update p_i
    fi
    end for
    Search the best global leader and record its position in g
while (loop_number < total_loop)
```

Figure 1: General outline of the synchronous g-PSO Algorithm

4. **Schwefel's function**:

$$\mathcal{F}_4(\vec{x}) = 418.9829n + \sum_{i=1}^{n} x_i \sin\left(\sqrt{\mid x_i \mid}\right) \quad (10)$$

where $\mathcal{F}_4(\vec{x}^*) = 0$ for $\vec{x}^* = (0, 0, ..., 0)$ is the global optimum.

5 Experimental Setup

Table 1: Function parameters

Function	n	Xmin	Xmax
$\mathcal{F}_1$	30	-5.12	5.12
$\mathcal{F}_2$	30	-600	600
$\mathcal{F}_3$	30	-30	30
$\mathcal{F}_4$	30	-500	500

The parameters adopted for our two PSO models are shown in Table 1 for each of the test functions previously indicated. Note that in all cases we used the same number of dimensions (30) and that the only difference were the ranges of the variables.

In all our experiments, we used 20 particles, and 10,000 iterations (i.e., a total of 200,000 evaluations of the objective function). The values of the additional parameters are the following: $c_1 = c_2 = 1.3$, $w = 0.3$, initial mutation rate = 0.9 (this value decreases as the iteration number increases), neighborhood radius = 4 (20% of the population size as suggested in [9]). The values of these parameters were empirically derived after performing a set of experiments.

All our experiments were conducted on a Notebook CompaqPresario 1692 with an AMDK62 processor running at 450 Mhz and with 96 MB of RAM. The algorithms were implemented under the Linux operating system, using the GNU C++ compiler. For each of the test functions previously described, we performed 100 runs per model (global and local).

6 Comparison of Results

We compare results with respect to Peer et al. [13], who studied 6 different neighborhood topologies for the so-called *guaranteed convergence PSO* (GCPSO) [16]. We used the same number of fuction evaluations as Peer et al. [13] to allow a fair comparison of results.

```
1. Swarm Initialization
2. for i = 1 to number of particles do
3.     for j = 1 to number of dimensions do
4.         Initialize x_ij with a rnd(Xmin, Xmax) value
5.         Initialize v_ij with zero value
6.         copy x_ij in p_ij
7.     end
8. end
9. Swarm flight through the Search Space
11. do
12.     for i = 1 to number of particles do
13.     Search better in the k_neighborhood of particle x_i and record it in l
14.         for j = 1 to number of dimensions do
15.             Update v_ij using p_ij and p_lj
16.             Prevent explosion of v_ij
17.             Update x_ij
18.             if (loop_number < total_loop * prob_mut)
19.               then Mutate x_ij
20.             fi
21.         end for
22.     Evaluate fitness(x_i)
23.     if fitness(p_i) < fitness(x_i)
24.         then update p_i
25.     fi
26.     end for
27. while (loop_number < total_loop)
```

Figure 2: General outline of the asynchronous l-PSO Algorithm

Peer et al. [13] use the prefixes G and P to identify if GCPSO (G) or standard PSO (P) are being used. The subscripts g, l and v refer to the neighborhood topology used: $gbest$ (g), $lbest$ (l) or von Neumann (v). We use the prefix M (for mutation-extended PSO) and subscripts g (global model) and l (local model). We follow Peer et al.'s notation in which $\overline{x}(s)$ is reported, where $\overline{x}$ is the average value obtained and s is the standard deviation of the results. We also indicate the median and the average range between the best and worst values obtained.

From Table 2, we can see that both our global and our local models outperformed the 6 models studied in [13]. Our $lbest$ model was able to converge to the global optimum of Rastrigin's function without much difficulties. In this case, the mean of the number of iterations at which the best solution was reached by our l_best model was 7763 (median was 7787) with an average processing time of 15 seconds (these values are not reported by Peer et al. [13]).

On Table 3 we can see the comparison of results between our two PSO models and the 6 topologies adopted by Peer et al. [13] when dealing with Griewank's function. In this case, results are similar, although we can see that the M_l algorithm found the best average values with the lowest standard deviation. In this case, the mean of the number of iterations at which the best solution was reached by our l_best model was 7178 (median was 7160) with an average processing time of 18.5 seconds

On table 4, we can see that for Ackley's funcion, our two models had perfect convergence in the 100 runs performed (i.e., in all the runs, the entire swarm converged to the global optimum). This contrasts with the performance presented by the 6 topologies studied by Peer et al. [13]. P_l had the best average behavior from the 6 topologies studied by Peer et al., but still was unable to achieve perfect convergence in all the runs performed. In this case, the mean of the number of iterations at which the best solution was reached by our l_best model was 7763 (median was 7787) with an average processing time of 15 seconds

On table 5, we compare our two PSO models with respect to the 6 PSO topologies studied by Peer et al. [13] in Schwefel's function. Note that in this case, our two models again presented better average results than any of the other techniques with respect to which were compared. Not only are the average and the median values lower for our two PSO models, but also their corresponding standard deviations. In this case, the mean of the number of iterations at which the best solution was reached by our l_best model was 8911 (median was 8918) with an average processing time of 15.5 seconds.

Table 2: Comparison of results for Rastrigin's function. We compare our 2 models (M_g and M_l) with respect to the 6 topologies studied in [13].

Algorithms	$\overline{x}(s)$	Median	Range
G_g	71.925 (18.692)	71.637	[31.839:139.29]
G_l	61.202 (15.415)	60.692	[27.859:93.526]
G_v	55.837 (14.501)	54.723	[29.849:99.496]
P_g	72.204 (18.678)	68.652	[29.849:136.31]
P_l	64.567 (14.941)	63.677	[31.839:112.43]
P_v	54.355 (15.353)	51.738	[22.884:98.501]
M_g	46.891 (15.554)	45.031	[4.4447:119.50]
M_l	3.0016 (3.8211)	2.6209	[0.0000:42.796]

Table 3: Comparison of results for Griewank's function. We compare our 2 models (M_g and M_l) with respect to the 6 topologies studied in [13].

Algoritmos	$\overline{x}(s)$	Median	Range
G_g	0.0162 (0.0219)	0.0074	[1e-19:0.1077]
G_l	0.0039 (0.0075)	1e-19	[1e-19:0.0393]
G_v	0.0104 (0.0145)	0.0074	[0.000:0.8549]
P_g	0.1353 (0.3154)	0.0405	[2e-19:2.1642]
P_l	0.0051 (0.0086)	1e-19	[0.000:0.0394]
P_v	0.0134 (0.0190)	0.0074	[0.0000:0.0903]
M_g	0.0253 (0.0224)	0.0193	[0.0247:0.0247]
M_l	0.0014 (0.0047)	0.0000	[0.0000:0.1367]

Table 4: Comparison of results for Ackley's function. We compare our 2 models (M_g and M_l) with respect to the 6 topologies studied in [13].

Algoritmos	$\overline{x}(s)$	Median	Range
G_g	2.0181 (1.3102)	1.9565	[1e-14:9.3975]
G_l	0.2794 (0.5424)	7e-15	[7e-15:2.221]
G_v	0.6824 (0.8269)	7e-15	[4e-15:2.5799]
P_g	3.6708 (1.5625)	3.3444	[3e-14:8.9018]
P_l	0.0667 (0.2676)	7e-15	[7e-15:1.3404]
P_v	0.7098 (0.8450)	7e-15	[4e-15:2.8870]
M_g	0.0000 (0.0000)	0.0000	[0.000:0.0000]
M_l	0.0000 (0.0000)	0.0000	[0.000:0.0000]

Table 5: Comparison of results for Schwefel's function. We compare our 2 models (M_g and M_l) with respect to the 6 topologies studied in [13].

Algoritmos	$\overline{x}(s)$	Median	Range
G_g	4539.0 (706.06)	4550.2	[2882.2:6534.1]
G_l	4762.1 (509.36)	4797.8	[3395.3:5863.1]
G_v	4496.9 (707.83)	4451.5	[2862.3:6356.5]
P_g	4535.6 (722.33)	4510.8	[2803.1:6179.5]
P_l	4634.0 (642.22)	4609.5	[2329.4:6219.5]
P_v	4273.4 (565.86)	4333.1	[2664.9:5449.1]
M_g	1498.9 (304.60)	1500.2	[1498.9:1498.9]
M_l	1832.9 (192.00)	1255.2	[1305.2:11390.]

In general, we can see from the results presented that our two PSO models presented a highly competitive performance. It is also worth noticing that such a performance improved when dealing with more complex functions (e.g., Rastrigin's and Schwefel's functions), where the 6 PSO topologies studied by Peer et al. [13] presented a much poorer performance.

We believe that this good performance of our two models is mainly due to the exploratory power of our non-uniform mutation operator, which avoids that PSO gets easily trapped in local optima. This is particularly important when dealing with highly multimodal functions such as those studied in this paper and therefore the better behavior of our two approaches. It is also important to indicate that in the study conducted by Peer et al. [13], at least one of the topologies compared (von Neumann's topology) requires a more complex implementation. This additional implementation complexity was compensated by a better performance with respect to the other 5 topologies analyzed by Peer et al. [13]. However, in this paper we have shown that the simple addition of a mutation operator can considerably improve on the performance of a PSO algorithm even without using a complex topology, at least when dealing with multimodal functions.

7 Conclusions and Future Work

In this paper, we have addressed the use of particle swarm optimization in multimodal optimization. There is evidence in the literature related to the fast convergence properties of PSO [9]. However, such a fast convergence can sometimes become premature convergence, because PSO approaches may get trapped in local optima from which they cannot escape. This undesirable behavior seems to be evident mainly in multimodal function where many false attractors (i.e., local optima) exist and there is only one global optimum that we wish to reach.

In this paper, we have proposed the use of two simple PSO models (based on neighborhoods) hybridized with a non-uniform mutation operator taken from the evolutionary algorithms literature. Our main hypothesis revolved around the fact that mutation is a powerful diversity maintenance mechanism and thus, we thought that adding it to PSO would improve its performance particularly when dealing with multimodal functions.

To validate our hypothesis, we compared our two PSO models with respect to the 6 PSO models studied by Peer et al. in a recent paper [13]. Our results indicate that our hybrid PSO models outperform the other highly competitive (and in some cases more complex) PSO topologies studied by Peer et al. [13]. We could see that our local best model (l_best) had the best overall performance of all the approaches compared. Our local best model outperformed our global best model (except in Ackley's function in which both models achieved perfect convergence), which seems to indicate that local neighborhoods may be more useful for multimodal optimization than global neighborhoods.

As part of our future work, we intend to extend our two PSO models to non-stationary environments (i.e., dynamic functions) [3]. In order to do that, we will certainly need to use additional mechanisms that allow PSO to better track a moving target. For that sake we also intend to use some of the mechanisms that have been developed for evolutionary algorithms [4].

Acknowledgements

The first author acknowledges support from LIDIC (Laboratorio de Investigación y Desarrollo en Inteligencia Computacional) and from the Universidad Nacional de San Luis. The second author acknowledges support from the Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 32999-A.

Bibliography

[1] P. J. Angeline. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII, 7th International Conference EP'98 (California, USA)*, pages 601–610. Springer-Verlag. Lecture Notes in Computer Science No. 1447, March 1998.

[2] P. J. Angeline. Using selection to improve particle swarm optimization. In *International Conference on Evolutionary Computation*, Piscataway, NJ., 1998. IEEE Service Center.

[3] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, Boston/Dordecht/London, 2002.

[4] A. J. Carlisle. *Applying the Particle Swarm Optimizer to Non-Stationary Environments*. PhD thesis, Auburn University, December 2002.

[5] N. Higashi and I. H. Particle swarm optimization with gaussian mutation. In *Swarm Intelligence Symposium (Indianapolis, Indiana)*, pages 72–79, Piscataway, NJ., April 2003. IEEE Service Center.

[6] J. Kennedy. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Congress on Evolutionary Computation, CEC'99 (Whasington DC, USA)*, volume 3, pages 1931–1938, Piscataway, NJ., July 1999. IEEE Service Center.

[7] J. Kennedy and R. C. Eberhart. A new optimizer using particle swarm theory. In *Proceedings of Sixth International Symposium on Micro Machine and Humaan Science (Nogoya, Japan)*, pages 39–43, Piscataway, NJ., 1995. IEEE Service Center.

[8] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of International Conference on Neural Networks (Perth, Australia)*, pages 1942–1948, Piscataway, NJ., 1995. IEEE Service Center.

[9] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, California, USA, 2001.

[10] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Congress on Evolutionary Computation(Honolulu, Hawaii)*, Piscataway, NJ., May 2002. IEEE Service Center.

[11] M. Lovbjerg and T. Krink. Extending particle swarm optimisers with self-organized criticality. In *Congress on Evolutionary Computation*, pages 1588–1593, Piscataway, NJ., 2002. IEEE Service Center.

[12] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.

[13] E. S. Peer, F. van den Bergh, and A. P. Engelbrecht. Unsing neighbourhoods with the guaranteed convergence pso. In *Swarm Intelligence Symposium (Indianapolis, Indiana)*, pages 235–242, Piscataway, NJ., 2003. IEEE Service Center.

[14] J. Riget and J. S. Vesterstrom. A diversity-guided particle swarm optimizer - the arpso. Technical Report 2002-02, Evalife, University of Aarhus,Denmark, 2002.

[15] Y. Shi and R. C. Eberhart. A modified particle swarm optimizer. In *International Conference on Evolutionary Computation (Anchorage, AK)*, Piscataway, NY., 1998. IEEE Service Center.

[16] F. van den Bergh. *An Analysis of Particle Swarm Optimization*. PhD thesis, Faculty of Natural and Agricultural Science, University of Petroria, Pretoria, South Africa, November 2002.