PARTICLE SWARM OPTIMIZATION FOR IMPRECISE PROBLEMS

K. E. Parsopoulos
Department of Mathematics, University of Patras Artificial Intelligence
Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece

M. N. Vrahatis
Department of Mathematics, University of Patras Artificial Intelligence
Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece

1. SUMMARY

In this paper the ability of the Particle Swarm Optimization technique to cope with noisy functions is discussed. Results of experiments performed for particle identification by light scattering as well as on well-known test functions are presented and conclusions are derived.

2. INTRODUCTION

In most engineering applications optimization problems of continuous or discrete nature arise very often, making thus optimization techniques of undisputed importance in science and technology. Applications of them can be met in many scientific fields. Indeed, many recent advances in science, economics and engineering rely on numerical techniques for computing globally optimal solutions to corresponding optimization problems [10]. Usually, due to the existence of multiple local and global optima, these problems cannot be solved by classical nonlinear programming techniques.

Most Global Optimization (GO) techniques consist of a “local” and a “global” component. The “local” component is usually a “traditional” (gradient descent) local optimization technique, while the “global” component is used to search globally the search space in a complete, “exhaustive” fashion. These remarks indicate the inherent computational demand of the GO algorithms, which increases non-polynomially, as a function of problem-size, even in the simplest cases.

In practice, most of the above-mentioned techniques detect just *sub-optimal solutions* of the objective function and not global solutions that, in many applications, are not only desirable but also indispensable. Moreover, in many applications there are imprecise values for the input data as well as for the function values. Therefore, the development of robust and efficient GO methods for noisy environments, such as the aforementioned is a subject of considerable ongoing research [7, 18, 23, 25].

The Particle Swarm Optimization (PSO) technique has been developed by Eberhart and Kennedy in 1995 [11] and it is a simple evolutionary algorithm which differs from other evolutionary computation techniques in that it is motivated from the simulation of social behavior. PSO exhibits good performance in finding solutions to static optimization problems [15, 16, 17]. In [18] a first study of the performance of

PSO in noisy and continuously changing environments has been presented. In the following paragraphs this study is extended and further experiments on well-known test functions as well as for particle identification by light scattering are presented.

The remaining of the paper is organized as follows: in Section 3 a general discussion of the optimization problem for noisy functions as well as a simulation of the influence of noise (proportional to a normally distributed random number with zero mean and various values of variance) is given. The PSO method and the particle identification by light scattering problem are briefly presented in Sections 4 and 5 respectively. In Section 6, numerical results are exhibited and in Section 7 concluding remarks are given.

3. OPTIMIZATION OF NOISY FUNCTIONS

Several methods for finding the extrema of a function $f : \mathrm{CL}(S) \subset \mathbb{R}^n \rightarrow \mathbb{R}$, where $\mathrm{CL}(S)$ is the closure of an open and bounded set S, have been proposed. There are many applications in different scientific fields such as mathematics, physics, engineering, computer science etc. Most of them require precise function and gradient values. In many applications though, precise values are either impossible or time consuming to obtain. For example, when the function and gradient values depend on the results of numerical simulations, then it may be difficult or impossible to get very precise values. Or in other cases, it may be necessary to integrate numerically a system of differential equations in order to obtain a function value, so that the precision of the computed value is limited. Furthermore, in many problems the accurate values of the function to be minimized are computationally expensive. Such problems are common in real life applications as in the optimization of parameters in chemical experiments or finite element calculations. With such applications in mind, robust methods are needed, which make good progress with the fewest possible number of function evaluations.

The theory of local optimization provides a large variety of efficient and effective methods for the computation of an optimizer of a smooth function f. For example, Newton-type and quasi-Newton methods exhibit superlinear convergence in the vicinity of a non-degenerate optimizer. However, these methods require the Hessian or the gradient, respectively, in contrast to other optimization procedures, like the simplex method [14], the direction set method of Powell [8], or some other recently proposed methods [6, 7, 25].

In some applications, however, the function to be minimized is only known within some (often unknown and low) precision. This might be due to the fact that evaluation of the function means measuring some physical or chemical quantity or performing a finite element calculation in order to solve partial differential equations. The function values obtained are corrupted by noise, namely stochastic measurement errors or discretization errors. This means that, although the underlying function is smooth, the function values available show a discontinuous behavior. Moreover, no gradient information is available. For small variances in a neighborhood of a point the corresponding function values reflect the local behavior of the noise rather than that of the function. Thus, a finite-difference procedure to estimate the gradient fails [6].

The traditional method for optimizing noisy functions is the simplex or polytope method by Nelder and Mead [8, 14]. This method surpasses other well-known optimization methods when dealing with the large noise case. However, this is not valid in the noiseless case. The ability of this method to cope with noise is due to the fact that it proceeds solely by comparing the relative size of the function values, as the proposed method does. The Simplex method does not use a local model of the function f and works without the assumption of continuity. Although this method has poor convergence properties (for a convergence proof of a modified version see [23]), it has been a useful method in many sequential applications, although it is difficult and inefficient to implement in parallel. The method can be deficient when the current simplex is very "flat". This can be avoided by suitable variants (see for example [23]). More sophisticated methods in this direction are discussed in [21].

To study the influence of the imprecise information (regarding the values of the objective function), we simulate imprecision with the following approach. Information about $f(x)$ is obtained in the form of $f_\sigma(x)$, where $f_\sigma(x)$ is an approximation to the true function value $f(x)$, contaminated by a small amount of noise η. Specifically, the function values are obtained as in [7]:

$$(1) \qquad f_\sigma(x) = f(x)(1+\eta), \quad \eta \sim N(0, \sigma^2),$$

where $\eta \sim N(0, \sigma^2)$ denotes a Gaussian-distributed random variable with zero mean and variance σ^2, i.e., relative stochastic errors are used for the test problems. To obtain η, we apply the method of Box and Muller [2], using various values of the variance.

4. THE PARTICLE SWARM OPTIMIZATION TECHNIQUE

As in all evolutionary techniques, in PSO a population of potential solutions is used to search the search space, too. However, PSO differs from other evolutionary algorithms in that there are no DNA inspired operators in order to manipulate the population. Instead, in PSO the population dynamics resembles the movement of a "birds' flock" while searching for food, where social sharing of information takes place and individuals can profit from the discoveries and previous experience of all other companions. Thus, each companion, called *particle*, in the population, which is called *swarm*, is assumed to "fly" over the search space in order to find promising regions of the landscape. In the function minimization case, such regions possess lower function values than other visited previously. In this context, each particle is treated as a point in a D-dimensional space which adjusts its own "flying" according to its flying experience as well as the flying experience of other particles (companions). There are many variants of the PSO proposed so far, after Eberhart and Kennedy introduced this technique [5, 11]. In our experiments a version of this algorithm that is derived by adding an inertia weight to the original PSO dynamics was used [4]. This version is described in the following paragraphs.

First let us define the notation adopted in this paper: the i-th particle of the swarm is represented by the D-dimensional vector $X_i = (x_{i1}, x_{i2},\ldots, \mathrm{x}_{iD})$ and the best particle of the swarm, i.e., the particle with the smallest function value, is denoted by the index g. The best previous position (the position giving the best function value) of the i-th particle is recorded and represented as $P_i = (p_{i1}, p_{i2},\ldots, p_{iD})$, and the position change (velocity) of the i-th particle is V_i=$(v_{i1}, v_{i2},\ldots, v_{iD})$. The particles are then manipulated according to the equations

$$(2) \qquad v_{id} = \chi\,[w\, v_{id} + c_1\, r_1\, (p_{id} - x_{id}) + c_2\, r_2\, (p_{gd} - x_{id})],$$
$$(3) \qquad x_{id} = x_{id} + v_{id},$$

where $d = 1, 2,\ldots, D$; $i = 1, 2,\ldots, N$ and N is the size of the swarm; w is the inertia weight and its role is discussed below; c_1 and c_2 are two positive constants; r_1 and r_2 are two random values into the range [0, 1]; χ is a constriction factor which is used in constrained optimization problems in order to control the magnitude of the velocity (in unconstrained optimization problems it is usually set equal to 1.0).

The first equation is used to calculate the *i*-th particle's new velocity by taking into consideration three terms: the particle's previous velocity, the distance between the particle's best previous and its current position, and, finally, the distance between the swarm's best experience (the position of the best particle in the swarm) and the *i*-th particle's current position. Then, following the second equation, the *i*-th particle flies toward a new position. In general, the performance of each particle is measured according to a predefined fitness function, which is problem-dependent. The role of the inertia weight w is considered very important in PSO convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity, regulating the trade-off between the global and local exploration abilities of the swarm. A large inertia weight facilitates global exploration (searching new areas), while a small one tends to facilitate local exploration, i.e., fine-tuning the current search area. A general rule of thumb suggests that it is better to initially set the inertia to a large value, in order to make better global exploration of the search space, and gradually decrease it to get more refined solutions, thus a linearly decreasing inertia weight value is used. The initial population can be generated either randomly or by using a Sobol sequence generator [20] which ensures that the generated points will be uniformly distributed into the search space.

From the above discussion it is obvious that PSO, to some extent, resembles the evolutionary algorithms. However, in PSO, instead of using genetic operators (crossover and mutation), each particle updates its own position based on its own search experience and other particles' experience and discoveries. Adding the velocity term to the current position, in order to generate the next position, resembles the mutation operation in evolutionary algorithms. Note that in PSO, however, the mutation operator is guided by the particle's own "flying" experience and benefits by the swarm's "flying" experience. In another words, PSO is considered as performing mutation with a "conscience", as pointed out by Eberhart and Shi [4].

5. PARTICLE IDENTIFICATION BY LIGHT SCATTERING

Astronomy, Meteorology, Medicine and Bioengineering are just some of the fields where the laser light scattering measurements have become an important tool in the determination of the size and optical characteristics of small particles. There are actually two techniques for the determination of particle size using light scattering: Dynamic Light Scattering (DLS) and Static Light Scattering (SLS). In DLS a laser is focused into a small volume of solution containing the colloidal particles and the scattered light is collected over a small solid angle. The phase and polarization of the light scattered by any given molecule is determined by its size, shape, and composition. Random Brownian motion causes the total intensity at the detector to fluctuate at time. Autocorrelation functions are generated from these fluctuations and then inverted to obtain the distribution of particle sizes [3, 9]. On the other hand, in SLS the total intensity of the scattered light is measured as a function of angle and this information is used to determine the particle size distributions. For spheres the angular dependence of the scattered light is described by the Mie scattering function

[1], which depends on the refractive indices of the particle, the surrounding medium, and the size of the particles present [9].

A fundamental problem in inverse light scattering is the determination of the refractive index and the radius of homogeneous spherical particles suspended in a known medium. In order to solve that problem, plane polarized light of known wavelength is scattered from these particles and the intensity I is measured at a series of angles. The standard Lorenz-Mie theory [1] is used to describe the process and the one-dimensional scattering pattern observed is used to characterize both single particles and particle distributions. Thus, having experimental measures $I_s(\theta_1)$, $I_s(\theta_2)$, ..., $I_s(\theta_m)$ of the scattered light intensity for several angles, we wish to determine the corresponding values of the index of refraction n and the particle radius r. The value of n can be either real or complex. The intensity values vary widely relatively to the angle and thus it is better to work with the logarithm of the intensities. The objective function that is used is:

$$(4) \qquad E_1 = \sum_{j=1}^{m} [i_s(\theta_j) - z(\theta_j, r, n)]^2,$$

where i_s=log(I_s) and the $z(.)$ function is the logarithm of the theoretically computed intensity. If $n \in \mathbb{R}$ then the problem of minimizing E_1 is three-dimensional, otherwise, if $n \in \mathbb{C}$, it is two-dimensional. Several techniques have been used to solve the aforementioned problem. For real values of n, random search and multilevel single-linkage clustering have been applied. However, the function E_1 has many local minima and thus the initial guess of the solution is of crucial importance. Since evolutionary techniques do not depend on the initial guess, they appear as a good alternative to cope with the problem [9]. In our study the PSO technique is applied to solve it and the results are very promising as it will be exhibited in the next section.

6. EXPERIMENTAL RESULTS

First an experiment for particle identification by light scattering has been performed. A brief description of the problem has been given in the previous section. The simulated intensity functions $I(\theta)$ were generated using the BHMIE routine of Bohren and Huffman [1]. To these values, random amount of noise with several variance values was added to the function values. In the initial test problem n^*=1.40 and r^*=1.55 μm were used (these values constituted the global minimum) and the values of n and r were bounded by $1 \le n,\ r \le 2$. The angular range was between 0^o and 180^o with data points taken at 9^o increments. The wavelength was set equal to 0.5145 μm and the refractive index of the surrounding medium (water) was set equal to 1.336. The swarm's size was 30, the inertia weight w was gradually decreased from 1.2 toward 0.4 and $c_1 = c_2 = 0.5$. The maximum number of PSO iterations was 150 and the desired accuracy for all 30 runs was 10^{-5}. The results are given in Table 1. The results presented in the tables are the success rate, the mean value of the Frobenius norm of the difference between the obtained and the actual global minimizer, the mean number of iterations and the mean number of function evaluations done.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0 (noiseless)	90%	0.0793	99.9	3027.0
0.01	95%	0.0369	101.6	3078.0
0.03	95%	0.0216	119.5	3615.0
0.05	65%	0.1627	130.3	3939.0

Table 1. Results for n^*=1.40, r^*=1.55.

As can be seen, the addition of noise with small variance value increased the success rate of PSO by helping the alleviation of local minima, while the addition of noise with variance value around 0.5 decreased the overall performance of the algorithm. Similar are the results for n^*=1.65 and r^*=4.0 which is an area of the search space with much more local minima than the one in the previous experiment. The parameters were bounded by $1 \leq n \leq 2$ and $3 \leq r \leq 5$ and the swarm size was increased to 50 while the accuracy was set to 10^{-3}. The results are exhibited in Table 2.

Variance	Success Rate	Mean Distance	Mean Iterations	Mean Func. Eval.
0 (noiseless)	90%	0.1237	111.33	3370.0
0.01	95%	0.1601	70	2130.0
0.03	83%	0.4886	136.2	4530.0
0.05	94%	0.0008	89	2700.0

Table 2. Results for n^*=1.65, r^*=4.

Furthermore, six well known two-dimensional optimization test problems were considered in order to check the performance of the PSO technique: De Jong [22], Six-Hump Camel [12], Banana Valley [19], Freudstein-Roth [13], Goldstein-Price [12] and Rastrigin [22]. At each function evaluation, noise was added to the actual function value according to Relation (1) for different values of the variance and for each one of the variance values, 100 runs of the PSO algorithm were performed. In all experiments the values of the parameters for the PSO algorithm were the same as in the previous experiments except the swarm's size that was set equal to 20.

The interval inside which the initial swarm was taken, the desired accuracy and the results obtained for the different values of variance for each test problem, are exhibited in Tables 3-8. For all experiments the maximum allowed number of PSO iterations was 5000.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0.01	100%	0.000004	1694	33900
0.02	100%	0.000026	1687.56	33771.2
0.04	100%	0.0038	1683.8	33696
0.07	100%	0.00007	1690.56	33831.2
0.09	100%	0.000008	1691.52	33850.4

Table 3. Results for the DeJong function. Initial interval: $[-5, 5]^2$, desired accuracy: 10^{-6}.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0.01	100%	0.02784	1668.88	33397.6
0.02	100%	0.02411	1719.56	34411.2
0.04	100%	0.09616	1757.08	35161.6
0.07	100%	0.02526	1875.72	37534.4
0.09	100%	0.13648	1839.72	36814.4

Table 4. Results for the Six-Hump Camel function. Initial interval: $[-1, 1]^2$, desired accuracy: 10^{-3}.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0.01	100%	0.014606	1950.52	39030.4
0.02	96%	0.034635	2057.76	41175.2
0.04	92%	0.015245	2108.32	42186.4
0.07	92%	0.016156	2111.6	42252
0.09	88%	0.020198	2272.56	45471.2

Table 5. Results for the Banana Valley function. Initial interval: $[-3, 3]^2$, desired accuracy: 10^{-6}.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0.01	100%	0.000271	1729.84	34616.8
0.02	100%	0.000021	1741	34840
0.04	100%	0.000041	1742.08	34861.6
0.07	92%	0.487113	1988.16	39783.2
0.09	100%	0.039007	1723.68	34493.6

Table 6. Results for the Freudstein-Roth function. Initial interval: $[0, 10]^2$, desired accuracy: 10^{-6}.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0.01	100%	0.158967	1628.52	32590.4
0.02	100%	0.058264	1637.68	32773.6
0.04	100%	0.070651	1637.84	32776.8
0.07	100%	0.213214	1661.4	33248
0.09	100%	0.033704	1658.08	33181.6

Table 7. Results for the Goldstein-Price function. Initial interval: $[-1, 1]^2$, desired accuracy: 10^{-3}.

Variance	Succ. Rate	Mean Dist.	Mean Iter.	Mean Func. Ev.
0.01	100%	0.131205	1650.2	32121.4
0.02	100%	0.153097	1654.92	33118.4
0.04	100%	0.129110	1672.36	33467.2
0.07	100%	0.095561	1691.44	33848.8
0.09	100%	0.128160	1697.16	33963.2

Table 8. Results for the Rastrigin function. Initial interval: $[-1, 1]^2$, desired accuracy: 10^{-3}.

It is clear that noise addition caused no crucial instability to the PSO algorithm. Furthermore, the success rates remained high (not smaller than 90% except one case), even when the value of noise's variance was high. The mean number of iterations done by the PSO did not change significantly even when the variance increased from 0.01 to 0.09. The same behavior was noticed for the mean number of function evaluations.

Results for all experiments indicate that PSO is a very noise-tolerant technique and quite proper for minimization of noisy functions.

7. CONCLUSIONS

A study of the ability of the Particle Swarm Optimization method to minimize effectively noisy functions has been given. The experimental results indicate that in the presence of noise the PSO method is very stable and efficient. In fact, in many cases, the presence of noise seems to help PSO to avoid local minima of the objective function and locate the global one. Even in the cases where the standard deviation of the noise is large, PSO is able to move closely to the global minimizer's position. Thus, PSO has the ability to cope with noisy environments effectively and in a stable manner. Further work shall be done to check the performance of PSO in other dynamic environments and especially in other real-life applications.

8. REFERENCES

[1] C.F. Bohren, D.R. Huffman, *Absorption and Scattering of Light by Small Particles*, Wiley, New York, 1983.

[2] G.E.P. Box, M.E. Muller, A note on the generation of random normal deviates, *Ann. Math. Statistics* 29, 610-611, 1958.

[3] W. Brown (Ed.), *Dynamic Light Scattering: The Method and Some Applications*, Clarendon Press, Oxford, 1993.

[4] R.C. Eberhart, Y.H. Shi, Evolving Artificial Neural Networks, *Proc. Int. Conf. Neural Networks and Brain*, Beijing, P.R. China, PL5-PL13, 1998.

[5] R.C. Eberhart, P.K. Simpson, R.W. Dobbins, *Computational Intelligence PC Tools*, Academic Press Professional, 1996.

[6] C. Elster, A. Neumaier, A grid algorithm for bound constrained optimization of noisy functions, *IMA J. Numer. Anal.* 15, 585-608, 1995.

[7] C. Elster, A. Neumaier, A method of trust region type for minimizing noisy functions, *Computing* 58, 31-46, 1997.

[8] R. Fletcher, *Practical Methods of Optimization*, John Wiley, 1987.

[9] R.J.W. Hodgson, Genetic Algorithm Approach to Particle Identification by Light Scattering, *J. Colloid and Interface Science* 229, 399-406, 2000.

[10] R. Horst, P.M. Pardalos, N.V. Thoai, *Introduction to Global Optimization*, Kluwer Academic Publishers, 1995.

[11] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, *Proc. IEEE Int. Conf. Neural Networks*, Piscataway, NJ, USA, 1942-1948, 1995.

[12] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1999.

[13] J.J. More, B.S. Garbow, K.E. Hillstrom, Testing Unconstrained Optimization Software, *ACM Trans. Math. Soft.* 7(1), 17-41, 1981.

[14] J.A. Nelder, R. Mead, A simplex method for function minimization, *Computer J.* 7, 308-313, 1965.

[15] K.E. Parsopoulos, V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Objective function “stretching” to alleviate convergence to local minima, *Nonlinear Analysis TMA* 47, 3419-3424, 2001.

[16] K.E. Parsopoulos, V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Stretching technique for obtaining global minimizers through Particle Swarm Optimization, *Proceedings of the PSO Workshop*, Indianapolis, USA, 22-29, 2001.

[17] K.E. Parsopoulos, M.N. Vrahatis, Modification of the Particle Swarm Optimizer for locating all the global minima, *Artificial Neural Networks and Genetic Algorithms*, V. Kurkova *et al.* (Eds.), Springer, 324-327, 2001.

[18] K.E. Parsopoulos, M.N. Vrahatis, Particle Swarm Optimizer in noisy and continuously changing environments, *Artificial Intelligence and Soft Computing*, M.H. Hamza (Ed.), IASTED/ACTA Press, 289-294, 2001.

[19] D.A. Pierre, *Optimization Theory and Applications*, Dover Publications, 1986.

[20] W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery, *Numerical Recipes in Fortran 77*, Cambridge University Press, 1992.

[21] M.J.D. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, *Numerical Analysis Reports*, DAMTP 1992/NA5, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1992.

[22] R. Storn, K. Price, Differential Evolution-A simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *J. Global Optimization* 11, 341-359, 1997.

[23] V. Torczon, On the convergence of the multidimensional search algorithm, *SIAM J. Optimization* 1, 123-145, 1991.

[24] M.N. Vrahatis, G.S. Androulakis, J.N. Lambrinos, G.D. Magoulas, A class of gradient unconstrained minimization algorithms with adaptive stepsize, *J. Comp. Appl. Math.* 114, 367-386, 2000.

[25] M.N. Vrahatis, G.S. Androulakis, M.E. Manoussakis, A new unconstrained optimization method for imprecise function and gradient values, *J. Math. Anal. Appl.* 197, 586-607, 1996.