

# Particle Identification by Light Scattering Through Evolutionary Algorithms

**K.E. Parsopoulos, E.C. Laskari, M.N. Vrahatis**

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

Tel./Fax number: +30-2610-997348/992965

Email: {kostasp, elena, vrahatis}@math.upatras.gr

## 1. Abstract

Optimization methods are fundamental in science and technology, due to their application in numerous, diverse, fields. In most applications, the corresponding objective functions are multimodal, discontinuous, and frequently, they are contaminated by noise. Such problems cannot be solved effectively through classical Nonlinear Programming techniques. Evolutionary Algorithms are stochastic optimization methods that exploit function values solely. They are not affected by possible discontinuities of the objective function, and have proved to be noise tolerant.

This paper addresses the problem of Particle Identification by Light Scattering, which is, in general, contaminated by noise. To this end, two evolutionary algorithms, namely, Particle Swarm Optimization and Differential Evolution, are employed. In particular, two variants of each algorithm are considered. Both algorithms are efficient, and exhibit robust behavior with respect to noise. Overall, the obtained experimental results are satisfactory.

**Keywords:** Particle Swarm Optimization, Differential Evolution, Particle Identification by Light Scattering, Optimization, Noisy Problems.

## 2. Introduction

Optimization problems arise very often in science, technology and industry. In most applications, and especially in real-life applications, such problems involve the minimization of an objective function, which can be characterized by noise contamination, non-differentiability or even discontinuity. Moreover, the objective function may contain a multitude of local and global minimizers, rendering thus traditional Nonlinear Programming techniques non-applicable. Therefore, the development of robust and efficient methods for such problems, is a subject of considerable ongoing research [Elster, Neumaier (1997), Horst, Pardalos (1995)].

Evolutionary Algorithms (EAs), a class of stochastic algorithms, have been applied to address such problems. This class consists of stochastic optimization methods, which exploit function values only, and no derivative information. A further advantage of these methods is that they can work even on discontinuous functions. EAs use a population of potential solutions to investigate the search space simultaneously. Their operation is heavily based on ideas emanating from natural evolution and genetic dynamics. Genetic Algorithms, Genetic Programming, Evolution Strategies, and Differential Evolution (DE), are the most well-known and widely used EAs [Fogel (1996), Michalewicz (1994), Schwefel (1995), Storn, Price (1997)]. A novel class of algorithms, which were up to

recently categorized as EAs, is Swarm Intelligence (SI). SI algorithms also use a population, called “swarm” in this context, to probe the search space. However, their dynamics have been inspired from the behavior of socially organized organisms, rather than natural evolution. Such algorithms are the Particle Swarm Optimization (PSO), and the Ant Colony Optimization [Corne, Dorigo, Glover (1999), Kennedy, Eberhart (2001)].

Evolutionary Algorithms and SI techniques have proved to be very efficient, and are widely used in a diverse range of scientific and industrial applications, including economic modeling, finance, networks and transportation, database design, image processing, molecular biology, medicine, chemical engineering, design and control, and scheduling [Horst, Pardalos (1995)]. In this paper, the performance of two algorithms, namely Particle Swarm Optimization and Differential Evolution, is investigated and compared on a fundamental problem in Inverse Light Scattering, specifically, the determination of the refractive index and the radius of homogeneous spherical particles suspended in a known medium [Hodgson (2000)].

The paper is organized as follows: the problem of Particle Identification by Light Scattering, is described in Section 3. In Section 4, a general discussion of the optimization problem for noisy functions, is given. The PSO and DE algorithms are described in Sections 5 and 6, respectively. Experimental results are reported in Section 7, and the paper concludes in Section 8.

### **3. Particle Identification by Light Scattering**

A very interesting application, characterized by imprecise function values, is the problem of *Particle Identification by Light Scattering*. Astronomy, Meteorology, Medicine and Bioengineering are just some of the fields in which 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) [Hodgson (2000)]. 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 [Hodgson (2000), Parsopoulos, Vrahatis (2002b)]. 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 [Bohren, Huffman (1983)], which depends on the refractive indices of the particles, the surrounding medium, and the size of the particles present.

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 [Bohren, Huffman (1983)], 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,  $I_s(\theta_1), I_s(\theta_2), \dots, I_s(\theta_m)$ , 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:

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

where,  $i_s = \log(I_s)$ , and,  $z(\cdot)$ , is the logarithm of the theoretically computed intensity. If  $n$  is complex, then the problem of minimizing  $E$  is three-dimensional, otherwise, if  $n$  is real, it is two-dimensional. In practice, the aforementioned objective function is contaminated by noise.

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 [Hodgson (2000), Zakovic, Ulanowski, Bartholmew-Biggs (1998)]. However, the function  $E$ , has numerous 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 proper alternative to cope with this problem [Hodgson (2000)].

#### 4. Optimization of Noisy Functions

Several methods for computing the extrema of a function,

$$f: CL(S) \subset \mathbf{R}^D \rightarrow \mathbf{R},$$

where  $CL(S)$ , is the closure of an open and bounded set  $S$ , have been proposed. Traditional methods require precise function and gradient values. In many applications however, 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 obtain precise values. 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, and detect the optimal solution with the highest possible accuracy.

The theory of local optimization provides a large variety of efficient and effective methods for the computation of an optimizer of a smooth function,  $f(x)$ . 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 [Nelder, Mead (1965)], the direction set method of Powell [Fletcher (1987)], or some other recently proposed methods [Elster, Neumaier (1997), Vrahatis, Androulakis, Manoussakis (1996)].

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.

The traditional method for optimizing noisy functions is the Simplex or polytope method by Nelder and Mead [Fletcher (1987), Nelder, Mead (1965)]. 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 [Torczon (1991)]), it has proved to be useful in numerous sequential applications. However, it is difficult and inefficient to implement in parallel. Moreover, the method can be deficient when the working simplex is very “flat”. This can be avoided through suitable variants (see for example [Torczon (1991)]). More sophisticated methods in this direction are discussed in [Powell (1992)].

To study the influence of imprecise information (regarding the values of the objective function), we simulate imprecision through the following approach. Information about  $f(x)$ , is obtained in the form of  $f_\sigma(x)$ , which is an approximation to the true function value,  $f(x)$ , contaminated by a small amount of noise,  $\eta$ . Specifically, the function values are obtained as in [Elster, Neumaier (1997)]:

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

where,  $\eta \sim N(0, \sigma^2)$ , denotes a Gaussian-distributed random variable with zero mean and standard deviation  $\sigma$ , i.e., relative stochastic errors are used for the test problems. To obtain  $\eta$ , we apply the method of Box and Muller [Box, Muller (1958)], using various values of the standard deviation.

## 5. Particle Swarm Optimization

PSO is a population-based algorithm that exploits a population of individuals, to probe promising regions of the search space. In this context, the population is called *swarm* and the individuals are called *particles*. Each particle moves with an adaptable velocity within the search space, and retains in its memory the best position it ever encountered. In the *global* variant of the PSO the best position ever attained by all individuals of the swarm is communicated to all the particles. In the *local* variant, each particle is assigned to a neighborhood consisting of a prespecified number of particles. In this case, the best position ever attained by the particles that comprise the neighborhood is communicated among them. The present paper, considers the global variant of PSO only.

Assume a  $D$ -dimensional search space,  $S \subset \mathbf{R}^D$ , and a swarm consisting of  $N$  particles. The  $i$ -th particle is in effect a  $D$ -dimensional vector  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ . The velocity of this particle is also a  $D$ -dimensional vector,  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ . The best previous position encountered by the  $i$ -th particle is a point in  $S$ , denoted as  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})^T$ . Assume  $g$ , to be the index of the particle that attained the best previous position among all the individuals of the swarm. Then, according to a version of PSO which incorporates a parameter called *constriction factor*, the swarm is manipulated using the following equations [Clerc, Kennedy (2001)]:

$$V_i^{(t+1)} = \chi (V_i^{(t)} + c_1 r_1 (P_i^{(t)} - X_i^{(t)}) + c_2 r_2 (P_g^{(t)} - X_i^{(t)})), \quad (3)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}, \quad (4)$$

where  $i = 1, 2, \dots, N$ ;  $\chi$  is the constriction factor;  $c_1$  and  $c_2$  denote the *cognitive* and *social* parameters respectively;  $r_1, r_2$  are random numbers uniformly distributed in the range  $[0,1]$ ; and  $t$ , stands for the counter of iterations. The value of the constriction factor is typically obtained through the formula [Clerc, Kennedy (2001)]:

$$\chi = \frac{2k}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \quad (5)$$

for  $\varphi > 4$ , where  $\varphi = c_1 + c_2$ , and  $k = 1$ . Different configurations of  $\chi$  as well as a theoretical derivation of Eq. (5), can be found in [Clerc, Kennedy (2001)].

In a different version of PSO, a parameter called *inertia weight*, is used, and the swarm is manipulated according to the formulae [Kennedy, Eberhart (2001)]:

$$V_i^{(t+1)} = w V_i^{(t)} + c_1 r_1 (P_i^{(t)} - X_i^{(t)}) + c_2 r_2 (P_g^{(t)} - X_i^{(t)}), \quad (6)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}, \quad (7)$$

where  $i = 1, 2, \dots, N$ ; and  $w$  is the inertia weight, while all other variables are the same as in the constriction factor version. There is no explicit formula for the determination of  $w$ , which is employed to control the impact of the previous history of velocities on the current one. However, since 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, it appears intuitively appealing to initially set the inertia weight to a large value, in order to promote global exploration of the search space, and gradually decrease it to get more refined solutions [Kennedy, Eberhart (2001)]. The superiority of this approach against the selection of a constant inertia weight, has been experimentally verified. Thus, an initial value around 1 and a gradual decline towards 0 can be considered as a good choice for  $w$ .

Proper fine-tuning of the parameters  $c_1$  and  $c_2$ , results in faster convergence and alleviation of local minima. As default values,  $c_1 = c_2 = 2$  have been proposed, but experimental results indicate that alternative configurations, depending on the problem at hand, can produce superior performance [Kennedy, Eberhart (2001)].

The initialization of the swarm and the velocities, is usually performed randomly in the search space, although more sophisticated initialization techniques can enhance the overall performance of the algorithm [Parsopoulos, Vrahatis (2002a)]. For uniform random initialization in a multidimensional search space, a Sobol Sequence Generator can be used.

## 6. The Differential Evolution Algorithm

The Differential Evolution (DE) algorithm has been developed by Storn and Price [Storn, Price (1997)]. It is a parallel direct numerical search method, which utilizes  $N$ ,  $D$ -dimensional parameter vectors  $x_{i,G}$ ,  $i = 1, \dots, N$ , as a population for each iteration (generation) of the algorithm. The initial population is taken to be uniformly distributed in the search space. At each generation, the *mutation* and *crossover* operators are applied on the individuals, giving rise to a new population, which is subsequently subjected to the selection phase. The selection phase effectively identifies the  $N$  best points from both populations to comprise the next generation.

According to the *mutation* operator, for each vector  $x_{i,G}$ ,  $i = 1, \dots, N$ , a *mutant vector* is generated through the equation:

$$v_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}), \quad (8)$$

where  $r_1, r_2, r_3 \in \{1, \dots, N\}$ , are mutually different random indexes, and,  $F \in (0, 2]$ . The indexes  $r_1, r_2, r_3$ , also need to differ from the current index,  $i$ . Consequently, to apply mutation,  $N$  must be greater than or equal to 4.

Following the mutation phase, the *crossover* operator is applied on the population. Thus, a *trial vector*,

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \dots, u_{Di,G+1}), \quad (9)$$

is generated, where,

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1}, & \text{if } (\text{randb}(j) \leq CR) \text{ or } j = \text{rnbr}(i), \\ x_{ji,G}, & \text{if } (\text{randb}(j) > CR) \text{ and } j \neq \text{rnbr}(i), \end{cases} \quad (10)$$

where,  $j = 1, 2, \dots, D$ ;  $\text{randb}(j)$ , is the  $j$ -th evaluation of a uniform random number generator in the range  $[0,1]$ ;  $CR$  is the (user specified) crossover constant in the range  $[0,1]$ ; and,  $\text{rnbr}(i)$  is a randomly chosen index from the set  $\{1, 2, \dots, D\}$ .

To decide whether or not the vector  $u_{i,G+1}$  will be a member of the population of the next generation, it is compared to the initial vector  $x_{i,G}$ . Thus,

$$x_{i,G+1} = \begin{cases} u_{i,G+1}, & \text{if } f(u_{i,G+1}) < f(x_{i,G}), \\ x_{ji,G}, & \text{otherwise.} \end{cases}$$

The procedure described above is considered as the standard variant of the DE algorithm. Different mutation and crossover operators have been applied with promising results [Storn, Price (1997)]. In order to classify the different variants, the scheme  $DE / x / y / z$ , is used, where  $x$  specifies the mutated vector (“rand” for randomly selected individual or “best” for selection of the best individual);  $y$  is the number of difference vectors used; and,  $z$  denotes the crossover scheme (the scheme described here is due to independent binomial experiments, and thus, it is denoted as “bin”) [Storn, Price (1997)]. According to this description scheme, the DE variant described above is denoted as DE/rand/1/bin. One highly beneficial scheme that deserves special attention is the DE/best/2/bin scheme, where,

$$v_{i,G+1} = x_{best,G} + F(x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G}). \quad (11)$$

The usage of two difference vectors seems to improve the diversity of the population, if  $N$  is high enough.

## 7. Experimental Results

In this section, experimental results are reported. The simulated intensity functions,  $I(\theta)$ , have been generated using the BHMIE routine of Bohren and Huffman [Bohren, Huffman (1983)]. To these values, noise has been added, according to Eq. (2). Using these routines, the user can set an arbitrary global minimizer, and test the ability of his algorithm to detect this global minimizer. The selected global minimizers are the same as in [Hodgson (2000)], and the corresponding objective functions are heavily multimodal.

Two versions of PSO, one with inertia weight, and another with constriction factor, as well as the DE/rand/1/bin and DE/best/2/bin variants of DE, have been used. In all experiments, the size of the PSO's swarm and the DE's population have been set equal to 30. The inertia weight,  $w$ , has been gradually decreased from 1.2 towards 0.1, while  $\chi = 0.729$ , and  $c_1 = c_2 = 2.05$ . Regarding the DE parameters, the default values  $F = 0.8$ , and  $CR = 0.5$ , have been used. Each algorithm has been executed 20 independent times, for each one of the four different noise levels (i.e. standard deviation  $\sigma$ , of the noise), namely 0, 0.01, 0.05, and 0.1, which corresponds to a noise percentage between 0% and 10%. For each run, 200 iterations, (which corresponds to a budget of 6000 function evaluations) have been performed, for each variant of the two algorithms. The best position ever attained during the run, has been recorded, and its original function value

(i.e. without noise) as well as its distance from the actual global minimizer have been computed. Moreover, the results are averaged over the 20

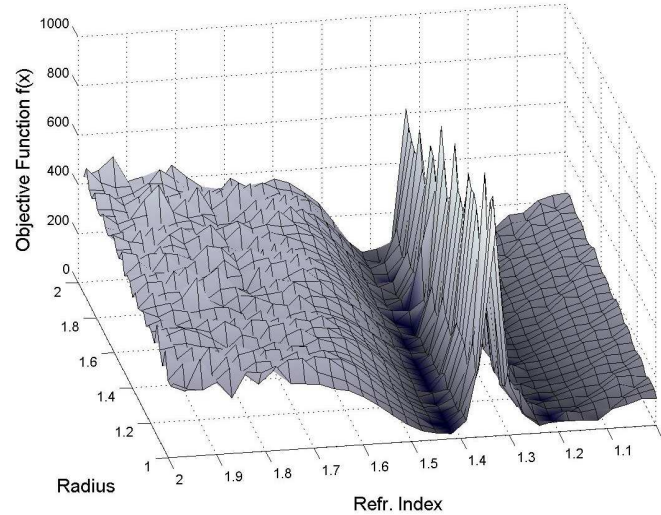


Fig. 1: The objective function  $f(x)$ , for  $n^* = 1.40$  and  $r^* = 1.55 \mu m$ .

runs, and their median, the standard deviation, as well as the minimum, are reported in the tables.

Initially, the global minimizer  $n^* = 1.40$  and  $r^* = 1.55 \mu m$  has been selected. The values of  $n$  and  $r$  have both been bounded in the range  $[1,2]$ . The angular range has been set between 0 and 180 degrees with data points taken at 9 degrees increments. The wavelength and the refractive index of the surrounding medium (water) have been set equal to  $0.5145 \mu m$  and 1.336, respectively. The corresponding objective function, defined by Eq. (1), is given in Figure 1, while the results are reported in Table 1. In another experiment, the global minimizer  $n^* = 1.65$  and  $r^* = 4 \mu m$  has been selected. The values of  $n$  have been bounded in  $[1,2]$ , while  $r$  has been bounded in  $[3,5]$ . The results are reported in Table 2. Moreover, a complex minimizer has been considered with similar results.

It is obvious that PSO always outperformed the DE algorithm. The inertia weight version has always been better than the constriction factor, and, the DE/rand/1/bin variant proved superior to the DE/best/2/bin variant. Moreover, it seems that, the addition of a small amount of noise, increased the success rate of both algorithms, by helping the alleviation of local minima. DE exhibits smaller standard deviations at the obtained results, which indicates that its results did not vary widely during the 20 runs.

## 8. Conclusions

A study of the performance of the PSO and DE algorithms to address efficiently the Particle Identification by Light Scattering problem, has been presented. Experiments using the BHMIE routine of Bohren and Huffman [Bohren, Huffman (1983)], for the simulation of the intensity function, have been performed, and multiplicative noise has been added to the corresponding function values.



$\sigma$	Method	Function Value			Distance from Global		
		Median	St.D.	Min	Median	St.D.	Min
0	PSOC	1.69E-13	2.98E00	2.46E-17	2.33E-09	1.24E-01	2.02E-11
	PSOW	2.71E-21	9.94E-01	7.31E-26	5.20E-13	4.72E-02	1.73E-15
	DE1	1.94E-06	5.93E-04	8.10E-10	8.72E-06	1.18E-04	2.66E-07
	DE2	2.70E-04	7.52E-02	1.70E-08	1.15E-04	1.29E-03	1.19E-06
0.01	PSOC	3.51E-15	2.05E00	6.18E-19	3.75E-10	8.26E-02	8.29E-12
	PSOW	5.29E-20	2.22E00	2.27E-24	1.99E-12	9.30E-02	4.17E-15
	DE1	2.21E-07	1.25E-04	1.79E-09	3.74E-06	2.89E-05	1.68E-07
	DE2	7.15E-04	3.59E-01	1.48E-07	1.62E-04	3.36E-03	3.63E-06
0.05	PSOC	3.29E-15	3.41E00	2.01E-17	5.92E-10	1.39E-01	2.49E-11
	PSOW	2.69E-20	2.04E00	1.57E-25	1.63E-12	8.26E-02	9.93E-16
	DE1	8.79E-07	1.23E-03	3.82E-10	6.37E-06	4.15E-05	4.57E-08
	DE2	8.70E-04	2.32E-01	5.83E-07	1.90E-04	1.74E-03	3.95E-06
0.1	PSOC	9.61E-15	2.61E00	1.39E-17	4.44E-10	1.13E-01	1.74E-11
	PSOW	5.48E-21	2.23E00	1.10E-24	4.24E-13	9.33E-02	1.11E-14
	DE1	9.19E-06	3.51E-03	6.13E-09	1.18E-05	2.81E-04	7.46E-07
	DE2	9.56E-06	2.20E-02	4.84E-08	2.97E-05	4.62E-04	1.01E-06

Table 1: Results for the case  $n^* = 1.40$  and  $r^* = 1.55$ .

In all experiments, PSO outperformed the DE algorithm, with the inertia weight version exhibiting superior performance. In some cases, the presence of a small amount of noise seems to help the method to avoid local minima of the objective function and locate the global one. Even in the case where noise was equal to 10% of the function value, both PSO and DE performed satisfactorily, with PSO producing two times more accurate solutions, given a specific budget of function

$\sigma$	Method	Function Value			Distance from Global		
		Median	St.D.	Min	Median	St.D.	Min
0	PSOC	3.80E-13	8.45E-01	1.13E-14	1.74E-08	1.58E-01	1.44E-09
	PSOW	2.68E-12	8.99E-03	7.02E-15	3.68E-08	1.47E-04	1.56E-09
	DE1	8.82E-06	2.79E-03	2.81E-08	5.67E-05	5.60E-04	4.39E-06
	DE2	9.23E-04	2.54E-02	1.08E-06	5.31E-04	2.38E-03	3.23E-05
0.01	PSOC	1.16E-13	2.53E-11	6.24E-15	1.08E-08	7.40E-08	1.38E-09
	PSOW	5.85E-13	5.41E-11	1.16E-17	1.78E-08	1.15E-07	1.15E-10
	DE1	1.92E-06	3.96E-05	6.73E-10	3.16E-05	6.20E-05	2.74E-07
	DE2	1.09E-04	9.02E-02	5.60E-07	2.36E-04	3.55E-03	4.23E-06
0.05	PSOC	4.13E-13	8.49E-01	1.41E-15	1.97E-08	1.58E-01	9.43E-10
	PSOW	1.01E-11	1.64E-10	1.63E-14	6.62E-08	2.19E-07	4.40E-09
	DE1	2.64E-06	1.38E-04	5.74E-09	3.49E-05	1.53E-04	2.28E-06
	DE2	7.32E-04	5.62E-02	8.66E-08	7.01E-04	7.90E-04	5.98E-06
0.1	PSOC	1.47E-12	9.69E-01	2.16E-13	3.47E-08	2.10E-01	5.83E-09
	PSOW	3.55E-12	9.10E-01	2.22E-14	5.98E-08	5.03E-07	4.66E-09
	DE1	4.81E-06	7.63E-05	1.20E-08	3.83E-05	1.43E-04	3.08E-06
	DE2	1.51E-03	5.76E-02	7.59E-07	1.16E-03	4.45E-03	2.98E-05

Table 2: Results for the case  $n^* = 1.65$  and  $r^* = 4$ .

evaluations. On the other hand, the standard deviation of the results obtained through DE, has been smaller, indicating that the algorithm is more robust compared to PSO. Further work is required to investigate thoroughly the performance of PSO and DE in noisy applications, while the development of

composite algorithms, which preserve the advantages of both methods, seems a promising research field.

## 9. References

- C.F. Bohren, D.R. Huffman (1983): *Absorption and Scattering of Light by Small Particles*. Wiley, New York.
- G.E.P. Box, M.E. Muller (1958): A note on the generation of random normal deviates. *Ann. Math. Statistics*, (29), pp. 610-611.
- M. Clerc, J. Kennedy (2001): The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.*, 6(1), pp. 58-73.
- D. Corne, M. Dorigo, F. Glover (1999): *New Ideas in Optimization*. McGraw-Hill.
- C. Elster, A. Neumaier (1997): A method of trust region type for minimizing noisy functions. *Computing*, (58), pp. 31-46.
- R. Fletcher (1987): *Practical Methods of Optimization*. Wiley, New York.
- D. Fogel (1996): *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ.
- R.J.W. Hodgson (2000): Genetic algorithm approach to particle identification by light scattering. *J. Colloid and Interface Science*, (229), pp. 399-406.
- R. Horst, P.M. Pardalos (1995): *Handbook of Global Optimization*. Kluwer, London.
- J. Kennedy, R.C. Eberhart (2001): *Swarm Intelligence*. Morgan Kaufmann Publishers.
- Z. Michalewicz (1994): *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.
- J.A. Nelder, R. Mead (1965): A simplex method for function minimization. *Computer Journal*, (7), pp. 308-313.
- K.E. Parsopoulos, M.N. Vrahatis (2002a): Initializing the particle swarm optimizer using the nonlinear simplex method. In A. Grmela, N.E. Mastorakis (eds.), *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, pp. 216-221, WSEAS Press.
- K.E. Parsopoulos, M.N. Vrahatis (2002b): Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, (1), pp. 235-306.
- M.J.D. Powell (1992): A direct search optimization method that models the objective and constraint functions by linear interpolation. *Techn. Report DAMTP 1992/NA5*, Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge, England.
- H.-P. Schwefel (1995): *Evolution and Optimum Seeking*. Wiley, New York.
- R. Storn, K. Price (1997): Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, (11), pp. 341-359.
- V. Torczon (1991): On the convergence of the multidimensional search algorithm. *SIAM J. Optimization*, (1), pp. 123-145.
- M.N. Vrahatis, G.S. Androulakis, M.E. Manoussakis (1996): A new unconstrained optimization method for imprecise function and gradient values. *Journal of Mathematical Analysis and Applications*, (197), pp. 586-607.
- S. Zakovic, Z. Ulanowski, M.C. Bartholmew-Biggs (1998): Application of global optimisation to particle identification using light scattering. *Inverse Problems*, 4(4), pp. 1053-1067.