
Tackling magnetoencephalography with particle swarm optimization

K.E. Parsopoulos*

Department of Mathematics,
University of Patras,
GR-26110 Patras, Greece
E-mail: kostasp@math.upatras.gr
*Corresponding author

F. Kariotou

Department of Chemical Engineering,
University of Patras,
GR-26500 Patras, Greece
E-mail: fkario@chemeng.upatras.gr

G. Dassios

Department of Chemical Engineering,
University of Patras and ICE/HT-FORTH,
GR-26500 Patras, Greece
E-mail: gdassios@chemeng.upatras.gr

M.N. Vrahatis

Department of Mathematics,
University of Patras,
GR-26110 Patras, Greece
E-mail: vrahatis@math.upatras.gr

Abstract: This paper investigates the performance of particle swarm optimization (PSO) and unified particle swarm optimization (UPSO) in magnetoencephalography (MEG) problems. For this purpose, two interesting tasks are considered. The first is the source localisation problem, also called the ‘inverse MEG problem’, where an unknown excitation source has to be identified, based on a set of sensor measurements that can be contaminated by noise. We refer to the second task as ‘forward task for inverse use’. It consists of the detection of the proper coefficients for approximating the magnetic potential through a spherical expansion, as accurately as possible. Also, the study of their behaviour under variations of the number of available measurements is considered. The obtained results are statistically analysed, providing useful insight regarding the applicability of the employed algorithms on such problems. Also, significant indications regarding the behaviour of several intrinsic dependencies of the problem are derived.

Keywords: particle swarm optimization; PSO; swarm intelligence; bio-inspired computation; magnetoencephalography; MEG; biomedical applications; bioinformatics; inverse problems.

Reference to this paper should be made as follows: Parsopoulos, K.E., Kariotou, F., Dassios, G. and Vrahatis, M.N. (2009) ‘Tackling magnetoencephalography with particle swarm optimization’, *Int. J. Bio-Inspired Computation*, Vol. 1, Nos. 1/2, pp.32–49.

Biographical notes: Konstantinos E. Parsopoulos received his PhD (2005) in Intelligent Optimisation from the Departments of Mathematics and Computer Engineering & Informatics, University of Patras, Greece. His work is focused on intelligent and numerical optimisation, computational intelligence and modelling of intelligent complex systems. He is currently a Lecturer with the Department of Mathematics, University of Patras, Greece.

Fotini Kariotou received her PhD in Applied Mathematics (2002) from the Department of Chemical Engineering, University of Patras, Greece. Her work is focused on analytic mathematical methods in the medical sciences and particularly in electroencephalography, magnetoencephalography and tumor growth. She has a Postdoctoral position in the Department of Chemical Engineering, University of Patras, Greece and also an Associate Lecturer in the Hellenic Open University.

George Dassios received his PhD in Applied Mathematics (1975) from the University of Illinois at Chicago, USA. His work includes wave propagation, scattering theory, biomedical problems and boundary value problems in science and modern technology. He is a Professor of Applied Mathematics in the University of Patras, since 1981.

Michael N. Vrahatis received his PhD in Computational Mathematics (1982) from the University of Patras, Greece. His work includes topological degree theory, systems of nonlinear equations, numerical and intelligent optimisation, data mining and unsupervised clustering as well as computational and swarm intelligence. He is a Professor of Computational Mathematics in the University of Patras, since 2000.

1 Introduction

Bioinformatics is one of the most interesting fields of application in evolutionary computation. The inherent nature of bioinformatics problems often implies modelling difficulties and uncertainties in computation, which cannot be straightforwardly addressed through classical methodologies. Therefore, they constitute a very challenging application field for approaches that require only minor information on the problem at hand and can take full advantage of modern distributed computer systems.

MEG is one of the most interesting biomedical techniques since it provides a means for the study of the functional human brain. This is possible by capturing and studying the magnetic fields produced by excitations of small regions of the brain, using sensor measurements (Bronzan, 1971; Dassios, 2006, 2007, 2008a, 2008b; Dassios and Fokas, 2008a; Dassios et al., 2005, 2007a, 2007b; Dassios and Kariotou, 2003a, 2003b, 2004, 2005; Fokas et al., 1996, 2004; Geselowitz, 1970; Grynspan and Geselowitz, 1973; Ilmoniemi et al., 2005; Nolte and Curio, 1997). These data can be used to model the MEG problem as an optimisation problem, which is usually highly nonlinear, requiring efficient algorithms for its solution.

PSO is a swarm intelligence algorithm for numerical optimisation problems (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 2001). PSO has gained increasing popularity in recent years due to its ability to solve efficiently and effectively a plethora of problems in science and engineering (Abido, 2002; Agrafiotis and Cedeno, 2002; Fourie and Groenwold, 2002; Ourique et al., 2002; Papageorgiou et al., 2005; Parsopoulos and Vrahatis, 2002b, 2004a; Pavlidis et al., 2005; Petalas et al., 2008, 2009; Skokos et al., 2005; Saldam et al., 2002; Ray and Liew, 2002). It has also been shown to be very efficient in biomedical applications (Cedeno and Agrafiotis, 2005; Forghani et al., 2007; Georgiou et al., 2006; Liu et al., 2008; Mohamed and Adel, 2006; Nakib et al., 2007; Qiu et al., 2005; Wachowiak et al., 2004; Xie and Jiang, 2005; Xu et al., 2008; Yang et al., 2008; Zhang and Li, 2007).

Unified particle swarm optimization (UPSO) was introduced by Parsopoulos and Vrahatis (2004b) as a modification of PSO that aggregates its local and global variant, combining their exploration and exploitation abilities without imposing additional requirements in terms of function evaluations. Convergence in probability was proved for a version of UPSO and preliminary experimental

results on both static and dynamic benchmark problems suggested that UPSO can outperform both the global and local variant of the standard PSO (Parsopoulos and Vrahatis, 2005a,c, 2006, 2007).

In this paper, we investigate the performance of PSO and UPSO in two interesting MEG problems. The first one is the well-known ‘inverse problem’, where the goal is to detect an unknown excitation source, using a set of (exact or noisy) sensor measurements.

The second problem, called the ‘forward task for inverse use’, refers to the computation of proper coefficients that optimise the approximations of the magnetic potential through spherical expansions, as well as the study of their tolerance under variations of the number of available sensor measurements.

The paper is organised as follows: the MEG problem is briefly described in Section 2 and the employed algorithms are analysed in Section 3. The experimental setup as well as the corresponding results for the two problems are reported and discussed in Section 4. The paper concludes in Section 5.

2 The MEG problem

One of the most promising, non-invasive methods for studying human brain activity in vivo and in real time is MEG. It is based on the fact that the brain is activated via an electrochemical excitation of a small region in the cerebral tissue, which produces a very weak yet measurable magnetic field outside the head. The biomagnetic signals are registered on a set of sensors divided up almost uniformly upon a helmet which covers the whole head except from the frontal face and the neck. MEG measurements are obtained by the SQUID, which is the most sensitive equipment ever built, since it can actually measure magnetic flux down to 50–500ft. (Hamalainen et al., 1993).

It is well established (Plonsey and Hepner, 1967) that biomagnetic fields live in the realm of the quasistatic Maxwell equations. This implies that, in the source-free exterior space, the magnetic field is represented by the gradient of a harmonic function. We refer to this function as the ‘scalar magnetic potential’ and denote it by U .

For a spherical conductor, which commonly models the human head and for the most widely used source model of a dipolar current with moment Q located at the interior point

r_0 , it is proved that the magnetic potential at an exterior point r is given by (Sarvas, 1987),

$$U(r) = \frac{(Q \times r_0)^\top r}{F(r; r_0)} \quad (1)$$

with

$$F(r; r_0) = \|r\| \|r - r_0\|^2 + \|r - r_0\| r^\top (r - r_0),$$

where $\|r\|$ stands for the measure of the vector r . In this work we address the following two tasks.

2.1 The inverse problem

The inverse MEG problem consists of the identification of a source position and moment from data provided solely by sensor measurements. Given a set of source parameters, which correspond to the moment vector,

$$Q = (q_1, q_2, q_3)^\top,$$

and position vector,

$$r_0 = (r_{01}, r_{02}, r_{03})^\top,$$

of the source, assumed to be normal to each other, we use equation (1) to acquire the values $U(r_i)$ for a number of sensor positions, $r_i, i = 1, 2, \dots, K$, outside the head. Using these values as initial data, our first task consists of applying PSO and UPSO to detect a vector that approximates the aforementioned source (moment and position vector) with a given accuracy.

2.2 The forward task for inverse use

Since $U(r)$ is a harmonic function, it enjoys a spherical expansion of the form,

$$U(r) = \sum_{\ell=1}^{\infty} A_\ell \frac{P_\ell(\cos(\theta))}{\rho^{\ell+1}} + \sum_{\ell=1}^{\infty} \sum_{k=1}^{\ell} \left[B_\ell^k \cos(k\phi) + C_\ell^k \sin(k\phi) \right] \frac{P_\ell^k(\cos(\theta))}{\rho^{\ell+1}}, \quad (2)$$

with respect to the Legendre functions,

$$P_\ell(\cos(\theta)) \text{ and } P_\ell^k(\cos(\theta)),$$

where A_ℓ, B_ℓ^k and C_ℓ^k are unknown coefficients depending on Q and r_0 and

$$(\rho, \phi, \theta),$$

are the spherical coordinates of the sensor position r .

It is proved (Dassios and Fokas, 2008b) that the eight first coefficients,

$$A_1, A_2, B_1^1, B_2^1, B_2^2, C_1^1, C_2^1, C_2^2,$$

which correspond to $\ell = 1, 2$, provide more than adequate information in order to analytically acquire the source. However, these coefficients must alleviate large fluctuations in their values when the number of sensor points used to reconstruct the function $U(r)$ from data values, varies. This necessity gives rise to our second task, which is the detection and investigation of the behaviour of these coefficients under variation of the number of sensors measurements.

This is possible by varying the number of sensors from very small to reasonable large values with a fixed increment, while, for each case, detecting with PSO and UPSO the coefficients that provide the smallest possible error in the approximation of $U(r)$ with the spherical expansion of equation (2). Recording and analysing the absolute relative error between the values of the coefficients for consecutive different number of measurements, can provide useful insight regarding their stability.

In the next section, the employed algorithms are described in detail.

3 The employed algorithms

For completeness, we first describe the employed PSO and UPSO algorithms.

3.1 Particle swarm optimization

PSO is considered as one of the most promising population-based algorithms for numerical optimisation problems. It was introduced in 1995 by Eberhart and Kennedy (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) and, since then, it has gained a constantly increasing popularity that is attributed to its efficiency and simplicity.

The fields of social psychology and evolutionary computation constituted the main sources of inspiration for the development of PSO. Fundamental laws encountered in natural swarms and socially organised groups were modelled to simulate patterns and emergent behaviours in nature. Swarm intelligence consists of models that adheres to the following five principles due to Millonas (1994):

- 1 'proximity', i.e., ability to perform space and time computations
- 2 'quality', i.e., ability to respond to environmental quality factors
- 3 'diverse response', i.e., ability to have a wide range of responses
- 4 'stability', i.e., ability to retain stable behaviour under mild environmental changes
- 5 'adaptability', i.e., ability to alter the behaviour when it is inevitable.

PSO was developed based on models that simulated social behaviour, following these properties. Thus, it was categorised as a swarm intelligence algorithm. Although PSO shares many common concepts with evolutionary algorithms, it has also some essential differences, revealed in the following paragraphs.

Similarly to other population-based approaches, PSO operates on a population of search points that probe the search space simultaneously. In the context of PSO, the population is called the *swarm* and its individuals are called the *particles*. The swarm is randomly initialised in the search space and the particles are let to move with an adaptable velocity, visiting new and unexplored regions. The movement is based on a scheme that takes into consideration the particle's own experience as well as the experience of a group of other particles, considered as its neighbours. The search stops as soon as a stopping criterion, usually related to the quality of the best solution found so far or the required computational cost, is achieved.

Let $f : S \subset \mathbb{R}^n \rightarrow Y$ be the objective function and

$$\min_{x \in S} f(x),$$

be the minimisation problem under consideration. PSO assumes a swarm of N particles,

$$\mathbb{S} = \{x_1, x_2, \dots, x_N\},$$

to probe S . Each particle is an n -dimensional vector,

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in})^\top \in S,$$

that moves within S with an adaptable velocity at each iteration of the algorithm. The term *velocity* shall not be confused with the corresponding physical quantity, but it is rather a position shift added to the current particle position.

During its movement, the particle records the best position,

$$p_i = (p_{i1}, p_{i2}, \dots, p_{in})^\top \in S,$$

it has ever visited, i.e., the position with the lowest function value. This position is the piece of information that will be communicated from the particle to a group of other particles that constitute its neighbourhood.

The *neighbourhood* provides a set of abstract communication channels between the particles and it can be defined in several different ways. The most obvious idea is to form neighbourhoods based on the actual distances of the particles using a metric in S . However, this approach is not viable in some high-dimensional spaces with large swarms due to its significant computational cost. Also, it promotes the formation of particle clusters that could be easily trapped around local minimisers. Moreover, there are search spaces where it may be difficult to define a proper metric. For these reasons, the idea of defining abstract neighbourhoods, based on the indices of the particles, gained ground and finally established. According to this idea, a neighbourhood of the particle x_i is a set,

$$\mathbb{N}_i = \{x_{k_1}, x_{k_2}, \dots, x_{k_M}\},$$

where,

$$\{k_1, k_2, \dots, k_M\} \subseteq \{1, 2, \dots, N\},$$

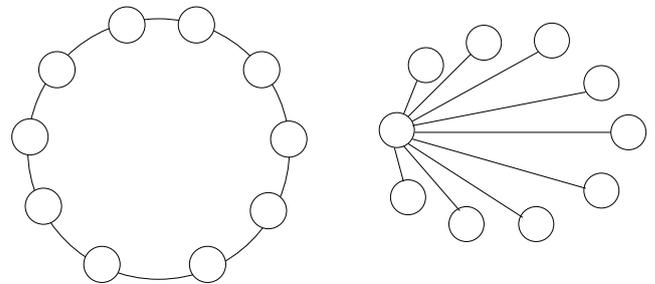
and M is an integer denoting the size of the neighbourhood.

The graph representing the connections among the particles is also called the neighbourhood's *topology*. The most common neighbourhood topology, based on the indices of the particles, is the *ring* topology, illustrated in the left part of Figure 1. Under this topology, each particle is assumed to have only two immediate neighbours, which are the particles with neighbouring indices. For example, the particle x_i in the ring topology has two immediate neighbours, the particles x_{i-1} and x_{i+1} , with x_1 following immediately after x_N . In the ring topology we can define a *radius* that determines the neighbourhoods size. Thus, a neighbourhood of x_i with radius $r_i < N$ is a set

$$\mathbb{N}_i^{r_i} = \{x_{i-r_i}, x_{i-r_i+1}, \dots, x_i, \dots, x_{i+r_i-1}, x_{i+r_i}\}.$$

The special case where $r_i = N$ for each particle $x_i, i = 1, 2, \dots, N$, assumes the whole swarm as the neighbourhood for every particle. Thus, the best positions of all particles are communicated among them and the overall best position is exploited by each particle. This scheme is illustrated in the right part of Figure 1 where the best position is assumed to be identified and communicated to all other particles. Different topologies have also been proposed and studied in the literature (Kennedy, 1999; Mendes et al., 2004; Suganthan, 1999). The use of equal neighbourhood size and topology for all particles is not mandatory in PSO, although it is the most common choice. The variant where each particle assumes the whole swarm as its neighbourhood is called the *global* PSO, also denoted as *gbest*, while in the case of strictly smaller neighbourhoods we have the *local* PSO, also denoted as *lbest*.

Figure 1 The ring (left) and star (right) neighbourhood topology of PSO



Let t denote the iteration counter and g_i be the index of the best position within the neighbourhood \mathbb{N}_i of x_i , i.e.,

$$g_i = \arg \min_j f(p_j),$$

such that $x_j \in \mathbb{N}_i$. Then, the particles and velocities are updated according to the following equations (Clerc and Kennedy, 2002):

$$v_{ij}(t+1) = \chi \left[v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{g_i,j}(t) - x_{ij}(t)) \right], \quad (3)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (4)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n,$$

where χ is the *constriction coefficient*; c_1, c_2 are positive acceleration parameters, called *cognitive* and *social* parameter, respectively; and r_1, r_2 are random numbers uniformly distributed in the range $[0, 1]$. The best positions are then updated at each iteration as follows:

$$p_i(t+1) = \begin{cases} x_i(t+1), & \text{if } f(x_i(t+1)) < f(p_i(t)), \\ p_i(t), & \text{otherwise.} \end{cases}$$

The constriction coefficient is used to control the magnitude of the velocities, alleviating the ‘swarm explosion’ effect that has been shown to be detrimental for the convergence of early PSO versions (Angeline, 1998). In those versions, the parameters were determined empirically, based on extensive experimentation on benchmark problems. In recent versions, the results reported in the PSO stability analysis due to Clerc and Kennedy (2002) and Trelea (2003) imply that parameters are selected such that the relation,

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

holds for $\varphi > 4$, where $\varphi = c_1 + c_2$, to ensure convergence. Further theoretical studies have been also reported in the literature (Cai et al., 2008; Cui and Zeng, 2004).

Also, in early PSO versions, a threshold v_{\max} was imposed on the velocity, such that,

$$|v_{ij}| \leq v_{\max}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n.$$

Although its use is not mandatory in the constriction coefficient version of PSO described by equations (3) and (4), experimental results indicate that it can enhance the algorithms performance (Eberhart and Shi, 2000).

The initialisation of the swarm and velocities is usually performed randomly in the search space, following a uniform distribution. The selection of such an initialisation is based on its simplicity and wide applicability since all modern computer systems are equipped with uniform random number generators. Nevertheless, more sophisticated initialisation schemes can enhance the performance of PSO (Parsopoulos and Vrahatis, 2002a). The best positions are assumed to coincide initially with the particle positions.

The search procedure of a population-based algorithm such as PSO consists of two main phases, *exploration* and *exploitation*. The former is responsible for the detection of the most promising regions in the search space, while the latter promotes convergence of the particles towards the best solution detected so far. These two phases can take place either once or successively during the execution of the algorithm.

An experimentally verified fact, which can be also intuitively inferred, is the dependence of PSO convergence speed on the information diffused among the particles through their interactions. Thus, a global variant, where the overall best position is available to all particles at each iteration, converges faster than a local variant, since all particles are attracted by the same best position. Therefore, it is distinguished for its exploitation ability. On the other hand, the local variant has better exploration properties, since the information regarding the best position of each neighbourhood is gradually communicated to the rest of the particles through their neighbours. Thus, the attraction towards a specific point is weaker, preventing the swarm from getting trapped in suboptimal solutions.

Evidently, proper selection of the neighbourhood size affects PSO trade-off between exploration and exploitation, albeit there is no formal procedure to determine the optimal size. The aforementioned neighbourhood-related properties of different PSO variants stand behind the development of UPSO, which is described in the next section.

3.2 Unified particle swarm optimization

UPSO was introduced by Parsopoulos and Vrahatis (2004b) as a PSO variant that harnesses the local and global PSO, combining their exploration and exploitation properties.

Let x_i be the i -th particle of the swarm. Let also g be the index of the best particle in the whole swarm and g_i be the index of the best particle in the neighbourhood of x_i , as described in the previous section. If $\mathcal{G}_i(t+1)$ denotes the updated velocity of x_i under the global PSO variant, i.e.,

$$\mathcal{G}_{ij}(t+1) = \chi \left[v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{g_i,j}(t) - x_{ij}(t)) \right], \quad (6)$$

and $\mathcal{L}_i(t+1)$ denotes the updated velocity of x_i under the local PSO variant, i.e.,

$$\mathcal{L}_{ij}(t+1) = \chi \left[v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{g_i,j}(t) - x_{ij}(t)) \right], \quad (7)$$

then, the main UPSO scheme is defined by replacing equations (3) and (4) in the standard PSO scheme, with the following equations (Parsopoulos and Vrahatis, 2004b):

$$U_{ij}(t+1) = u \mathcal{G}_{ij}(t+1) + (1-u) \mathcal{L}_{ij}(t+1), \quad (8)$$

$$x_{ij}(t+1) = x_{ij}(t) + \mathcal{U}_{ij}(t+1), \quad (9)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, n.$$

The parameter $u \in [0, 1]$ is called the *unification factor* and it balances the influence of the global and local search directions (velocities).

The standard local PSO is obtained by setting $u = 0.0$ in equation (8), while $u = 1.0$ corresponds to the standard global PSO. All intermediate values of $u \in (0, 1)$ define composite UPSO variants that combine the exploration and exploitation properties of the global and local PSO. Obviously, values near 0.0 favour the local PSO component, thereby promoting exploration, while values near 1.0 promote exploitation since the dominant component is that of global PSO. An extensive experimental study on the parameter u was conducted by Parsopoulos and Vrahatis (2007). The rest of the parameters adhere to the same principles as for the standard PSO case.

Besides the main UPSO scheme defined above, a stochastic parameter can also be incorporated in equation (8) to further enhance the exploration ability of the algorithm. In this case, equation (8) becomes either,

$$\mathcal{U}_{ij}(t+1) = r_3 u \mathcal{G}_{ij}(t+1) + (1-u) \mathcal{L}_{ij}(t+1), \quad (10)$$

which is mostly based on the local variant, or,

$$\mathcal{U}_{ij}(t+1) = u \mathcal{G}_{ij}(t+1) + r_3 (1-u) \mathcal{L}_{ij}(t+1), \quad (11)$$

which is mostly based on the global variant (Parsopoulos and Vrahatis, 2004b). The parameter $r_3 \sim \mathcal{N}(\mu, \sigma^2)$ is a normally distributed random number with mean μ and standard deviation σ . The use of r_3 imitates mutation in evolutionary algorithms. However, the mutation in UPSO is biased towards directions consistent with the PSO dynamic, in contrast to the pure random mutation used in evolutionary algorithms.

Following the assumptions of Matyas (1965), convergence in probability was proved for the mutated UPSO variants (Parsopoulos and Vrahatis, 2004b). UPSO has been shown to be effective, outperforming the standard PSO in several optimisation problems (Kotsireas et al., 2006; Parsopoulos and Vrahatis, 2005a, 2005b, 2005c, 2006, 2007).

4 Experimental results

Two different sets of experiments were conducted. In the first set, the ability of PSO and UPSO to address the inverse

(source identification) problem was investigated, while, in the second set, the most promising approach from the first set of experiments was used to compute the coefficients of the spherical expansion and investigate their behaviour with respect to the available number of sensors. The experimental setups per task as well as the obtained results are reported in detail in the following sections.

4.1 Experimental setup for the inverse problem

In this set of experiments, we investigated the ability of PSO and UPSO in locating the position of an unknown source when only the positions of a number of sensors and their corresponding measurements are available.

For this purpose, three source points were randomly generated inside a 3-dimensional sphere of radius $r_{\text{sphere}} = 9$, assuming that the origin coincides with the centre of the sphere. The source points are described by their moment, Q and position vector r_0 , as described in Section 2 and they are reported in Table 1. Also, three different levels of the number of sensors, namely $K_{\text{sensor}} = 50, 100, 200$, were considered and the corresponding equidistant sensor positions were generated on a sphere of radius $r_{\text{kask}} = 10$. For each sensor position, r_k , $k = 1, 2, \dots, K_{\text{sensor}}$, a measurement, $U_k = U(r_k)$, is obtained through equation (1) for a given source.

We investigated the case of exact measurements as well as the case of measurements contaminated by noise. In the latter, three different levels of multiplicative noise were considered. More specifically, each measurement assumed a noisy value,

$$U'_k = U_k (1 + \eta), \quad k = 1, 2, \dots, K_{\text{sensor}},$$

where $\eta \sim \mathcal{N}(0, \sigma_{\text{meas}}^2)$ is a normally distributed random number with zero mean and standard deviation σ_{meas} . The selection of normally distributed noise was based on the fact that measurement errors in nature and technology are very often modelled using this distribution. Also, multiplicative noise was preferred since the measurement noise is usually specified as a percentage of the actual measurement of the equipment. Thus, a noise value of η corresponds to an $100\eta\%$, percentage of measurement alteration. There is only the need to retain values of $\eta > -1$ in order to avoid changing the sign of U_k . In our noisy experiments we used three different values for $\sigma_{\text{meas}} = 0.01, 0.05, 0.10$.

Table 1 The three source points defined by their Q and r_0 components

Source	Vector	Components		
1	Q	-1.326594766376694	2.725358603156122	-2.288518082508507
	r_0	-1.896352580757411	-2.523289249725142	-1.905677167021398
2	Q	-0.365605139798790	0.558853821949274	-0.661437543925054
	r_0	4.061952539256966	0.816869532093309	-1.555037320414602
3	Q	-7.081215162607069	0.271084188968493	4.561706488935927
	r_0	2.334806885106570	1.455721569544961	3.537852962411117

The unknown source, as described by its $Q = (q_1, q_2, q_3)^\top$ and $r_0 = (r_{0_1}, r_{0_2}, r_{0_3})^\top$, was encoded as a vector that contains all components in a single array. Thus, each particle x_i constituted a potential source described as:

$$x_i = \left(q_1^{(i)}, q_2^{(i)}, r_{0_1}^{(i)}, r_{0_2}^{(i)}, r_{0_3}^{(i)} \right)^\top,$$

resulting in a five-dimensional, highly nonlinear optimisation problem. The third component of $Q^{(i)}$ was computed as,

$$q_3^{(i)} = \frac{q_1^{(i)} r_{0_1}^{(i)} + q_2^{(i)} r_{0_2}^{(i)}}{r_{0_3}^{(i)}},$$

in order to preserve orthogonality between $Q^{(i)}$ and $r_0^{(i)}$ (although special care was taken in the case of $r_{0_3}^{(i)} = 0$).

For each particle, the function values $U_k^{(i)}$, $k = 1, 2, \dots, K_{\text{sensor}}$, were computed through equation (1), using the aforementioned generated sensor positions, r_k , $k = 1, 2, \dots, K_{\text{sensor}}$, and x_i as the potential source location. The summed square-error against all measurements,

$$f(x_i) = \sum_{k=1}^{K_{\text{sensor}}} (U_k - U_k^{(i)})^2, \quad (12)$$

was used as the objective function value of x_i (in the noisy case, the corresponding noisy values were used).

Regarding the parameters of the algorithms, UPSO was used with values of u ranging from 0.0 to 1.0 with increments of 0.1, i.e.,

$$u = 0.0, 0.1, 0.2, \dots, 1.0,$$

in order to gain intuition regarding the efficiency of both local and global variants (recall that $u = 0.0$ and $u = 1.0$ correspond to the local and global variant of the standard PSO, respectively). Moreover, UPSO with mutation, described by equation (10), was also applied with $\mu = 0$ and $\sigma_{\text{UPSO}} = 1$.

In all experiments, a swarm of size equal to 50 was used with the default parameters $\chi = 0.729$, $c_1 = c_2 = 2.05$ (Clerc and Kennedy, 2002). The algorithm was terminated either when a position with objective function value (defined by equation (12)) smaller than 10^{-16} was detected or a maximum of 3000 iterations was reached. Since all potential solutions are chosen to lie within a sphere of radius $r_{\text{sphere}} = 9$ centred at the origin, the particles were all bounded within the range $[-9, 9]^5$. All parameter values are summarised in Table 2.

Table 2 Parameters for the experiments of the inverse problems

Type	Description	Values
Problem-related	Number of sensors	$K_{\text{sensor}} = 50, 100, 200$
	Sphere (head) radius	$r_{\text{sphere}} = 9$
	Kask radius	$r_{\text{kask}} = 10$
	Measurement noise	$\sigma_{\text{meas}} = 0.01, 0.05, 0.1$
	Solution bounds	$[-9, 9]^5$
	Required solution accuracy	10^{-16}
	UPSO-related	Unification factor
Mutation strength		$\sigma_{\text{UPSO}} = 1$
Velocity update parameters		$\chi = 0.729,$ $c_1 = c_2 = 2.05$
Swarm size		$N = 50$
Maximum iterations		3000

4.2 Results for the inverse problem

The results for the cases of exact as well as noisy measurements are analysed in the following subsections.

4.2.1 The case of exact measurements

We initially investigated the noiseless case where measurements are considered to be accurate. For each value of the unification factor, source and number of sensors, a total of 100 experiments were conducted for the plain UPSO as well as for the mutated UPSO approach, resulting in a total number of 18000 runs of the algorithms. In these experiments, the number of successes, i.e., the number of experiments (out of 100) where the solution was detected with the desired accuracy (10^{-16}) within the maximum number of 3000 iterations was recorded. The mean as well as the standard deviation, minimum and maximum number of required iterations averaged over all successful experiments was also recorded for each case.

The UPSO variant with $u = 0.1$ as well as the local PSO (i.e., $u = 0.0$) were the only algorithms that achieved a success rate of 100% in all cases, i.e., they were able to detect the solution with the required accuracy within the available number of iterations. The corresponding results for the two successful approaches are reported in Tables 3 and 5. The rest of the algorithms were characterised by loss of efficiency as u was increasing towards 1.0. This is a strong indication that the objective function is highly nonlinear; thereby algorithms that promote exploration are more efficient. The results for the less efficient approaches are omitted due to space limitations.

Table 3 contains the results for the local PSO variant. Three rows of the table are devoted to each source, one per different number of sensors. Local PSO can be obtained by the UPSO equations for $u = 0.0$. Thus, the mutated term of equation (10) vanishes. For this reason, no mutation was applied and the corresponding indicator in the fourth column of the table is 0 for each case. It can be observed that for the Sources 1 and 3, the mean required number of iterations increases with the number of sensors. On the other hand, for the Source 2 there was a slight decrease in the mean number when 100 sensors were used instead of 50. However, an increase was observed again for the case of 200 sensors. This is an expected observation, since an increase in the number of sensors offers more error terms in equation (12), thereby requiring higher computational cost.

In order to justify statistically the aforementioned observation, a Wilcoxon rank sum test was performed between the samples of required iterations received for the different numbers of sensors and the results are reported in Table 4. More specifically, for each source, we compared statistically the samples of the required iterations for the 100 experiments between the different numbers of sensors, under the null hypothesis that ‘the medians of the samples are equal’. Acceptance of the null hypothesis is denoted by 0 and rejection by 1. The statistical tests were performed in a significance level of 0.05 and the corresponding p -values are reported along with the decision (0 or 1) for rejecting the null hypothesis, inside a parenthesis after each p -value. As we can see, in most cases the samples were statistically different, although in two cases for the Source 2, the null hypothesis could not be rejected. This implies that, although there is a general trend for increased computational cost under higher numbers of sensors, the final cost depends heavily on the source point.

Similar analysis was performed for the case of UPSO with $u = 0.1$, except that we considered also the mutated variant of equation (10). The results are reported in Table 5 for both plain and mutated UPSO. We can observe that the plain UPSO outperformed in all cases its mutated counterpart, requiring almost half iterations in all cases. Nevertheless, both UPSO approaches clearly outperformed the local PSO, exhibiting smaller mean number of iterations and smaller standard deviations, which indicates that they were better with respect to both efficiency and robustness.

The most successful UPSO variant (the one without mutation) was statistically investigated, similarly to the local PSO, in order to identify statistical significance in the differences of its performance under different numbers of sensors. The corresponding hypothesis tests are reported in Table 6. Obviously, UPSO seems to be less affected by the addition of small number of sensors, since in most cases the null hypothesis of equal medians between the samples of 50 and 100 sensors as well as of 100 and 200 sensors could not be rejected. However, a significant increase (from 50 to 200 sensors) corresponded to statistically different performance of the algorithm for the two out of three sources. Again, the source position proved to play a substantial role on the performance of the algorithm.

Table 3 Results of the inverse problem without noise for the local PSO ($u = 0.0$)

u	Source	K_{sensor}	Mutation	Suc	Mean	Std	Min	Max
0.0	1	50	0	100%	1136.79	132.96	841.00	1501.00
		100	0	100%	1213.59	194.46	655.00	1802.00
		200	0	100%	1284.04	166.35	877.00	1708.00
	2	50	0	100%	1296.50	140.15	968.00	1616.00
		100	0	100%	1249.28	137.00	898.00	1611.00
		200	0	100%	1276.91	121.32	1032.00	1692.00
	3	50	0	100%	545.78	57.69	416.00	741.00
		100	0	100%	581.51	58.10	490.00	854.00
		200	0	100%	600.69	56.04	438.00	758.00

Note: Statistics on the required number of iterations are reported.

Table 4 Wilcoxon rank sum hypothesis tests at significance level 0.05 for the results of Table 3

u	Source	Number of sensors		
		50–100	100–200	50–200
0.0	1	1.662650e-03 (1)	5.324471e-03 (1)	1.951387e-10 (1)
	2	2.569073e-02 (1)	1.644320e-01 (0)	2.507944e-01 (0)
	3	8.218731e-06 (1)	2.883652e-03 (1)	1.176060e-11 (1)

Note: The p -value as well as the acceptance (0) or rejection (1) of the null hypothesis for equal medians are reported.

Table 5 Results of the inverse problem without noise for UPSO with $u = 0.1$, with and without mutation

u	Source	K_{sensor}	Mutation	Suc	Mean	StD	Min	Max
0.1	1	50	0	100%	347.67	65.70	256.00	733.00
			1	100%	674.82	98.12	512.00	1042.00
		100	0	100%	351.60	48.81	248.00	513.00
			1	100%	716.66	101.31	457.00	966.00
		200	0	100%	361.59	44.72	290.00	612.00
			1	100%	778.44	97.95	541.00	1006.00
	2	50	0	100%	376.54	54.48	245.00	615.00
			1	100%	802.78	104.27	609.00	1172.00
		100	0	100%	373.53	42.57	289.00	529.00
			1	100%	762.90	101.40	531.00	1071.00
		200	0	100%	385.96	48.03	291.00	503.00
			1	100%	766.22	81.94	554.00	1043.00
	3	50	0	100%	231.66	19.77	179.00	307.00
			1	100%	318.52	27.18	248.00	417.00
		100	0	100%	237.76	24.99	198.00	388.00
			1	100%	333.53	28.99	266.00	442.00
		200	0	100%	246.01	22.99	204.00	373.00
			1	100%	351.76	31.69	278.00	464.00

Note: Statistics on the required number of iterations are reported.

Table 6 Wilcoxon rank sum hypothesis tests at significance level 0.05 for the results of Table 5 without mutation

u	Source	Number of sensors		
		50–100	100–200	50–200
0.1	1	2.350101e-01 (0)	6.523472e-02 (0)	1.471446e-03 (1)
	2	9.454514e-01 (0)	7.956398e-02 (0)	9.061419e-02 (0)
	3	7.264613e-02 (0)	2.252211e-03 (1)	1.212955e-06 (1)

Note: The p -value as well as the acceptance (0) or rejection (1) of the null hypothesis for equal medians are reported.

Summarising the results, UPSO outperformed local PSO in all cases, exhibiting significantly better efficiency and robustness. Also, UPSO has shown to be less effected by relatively small increases in the number of sensors than local PSO. Nevertheless, in all cases, the results seem to depend on the source position. In the next section we investigate the case where noise is introduced in the measurements.

4.2.2 The case of noisy measurements

In this set of experiments, the obtained measurements for all different numbers of sensors are considered to be contaminated by noise, as described in Section 4.1. The two most successful algorithms reported in the previous section, namely the local PSO and UPSO with $u = 0.1$ without mutation, were exposed to the noisy problem for different levels of noise.

Since in the noisy case we are not able to identify exactly a solution by its function value, the algorithms were let to run at each experiment for a maximum number of 3000 iterations and the distance of the final best solution from the actual one under the ℓ_2 -norm, was considered as the performance quality measure.

The results for the local PSO are reported in Table 7. As expected, for given number of sensors, increasing levels of noise resulted in worst solutions, in terms of their distance from the actual source. Also, for given level of noise (e.g., 1%) the algorithm's performance in some cases was improved by adding sensors, while in other cases it deteriorated, depending on the source. Thus, it is not clear whether, in the presence of noise (which is the

case closer to reality), providing more measurements can enhance the efficiency of the algorithm. A possible explanation is that more measurements provide more information regarding the source location but also impose a higher accumulated error to the objective function of equation (12). This is also reflected to the hypothesis tests reported in Table 8, were in all cases the algorithm exhibits statistically significant differences under changes of the number of sensors.

Almost identical results were obtained for UPSO, as reported in Table 9. The existence of noise seems to degrade the performance of UPSO to the levels of plain local PSO, distorting the search directions that provide UPSO with a performance advantage. Thus, the detected solutions are almost identical with that of local PSO, differing only in the last few decimal digits of each component. This is reflected to the identical mean distances reported in the Tables 7 and 9. The same observation holds also for the hypothesis tests reported in Table 10. Therefore, in the case of noise the two algorithms exhibit almost identical behaviour for the considered levels of noise.

Summarising the results, in the noisy case there seems to be no crucial difference that could distinguish the two algorithms. In all cases, statistically significant differences in their performance are observed under changes in the number of sensors. These differences are in some cases beneficial for the algorithms, while in other cases are detrimental, depending on the source location, which has been shown again to play a crucial role.

Table 7 Results of the inverse problem with noise for the local PSO ($u = 0.0$)

u	Source	K_{sensor}	Noise	Mean	Std	Min	Max
0.0	1	50	1%	3.540907e-02	6.230506e-09	3.540906e-02	3.540909e-02
			5%	2.585456e-01	9.259604e-09	2.585456e-01	2.585457e-01
			10%	3.411434e-01	1.486605e-08	3.411434e-01	3.411434e-01
		100	1%	3.507655e-02	6.427307e-09	3.507653e-02	3.507657e-02
			5%	1.304847e-01	1.195694e-08	1.304846e-01	1.304847e-01
			10%	2.825110e-01	2.165706e-08	2.825110e-01	2.825111e-01
		200	1%	4.380892e-02	6.263326e-09	4.380890e-02	4.380893e-02
			5%	1.865135e-01	1.268799e-08	1.865134e-01	1.865135e-01
			10%	5.076858e-01	2.315223e-08	5.076857e-01	5.076858e-01
	2	50	1%	1.122122e-02	2.416897e-09	1.122122e-02	1.122123e-02
			5%	1.078816e-01	1.068674e-08	1.078816e-01	1.078816e-01
			10%	3.795751e-01	1.275627e-08	3.795751e-01	3.795751e-01
		100	1%	2.144142e-02	3.742187e-09	2.144141e-02	2.144143e-02
			5%	3.166544e-02	9.424213e-09	3.166541e-02	3.166546e-02
			10%	1.886913e-01	1.316551e-08	1.886913e-01	1.886914e-01
		200	1%	1.145255e-02	2.745366e-09	1.145254e-02	1.145256e-02
			5%	1.884985e-02	5.071426e-09	1.884984e-02	1.884986e-02
			10%	1.118533e-01	7.548534e-09	1.118533e-01	1.118533e-01
	3	50	1%	5.071279e-02	3.852004e-09	5.071278e-02	5.071280e-02
			5%	1.729159e-01	1.476072e-08	1.729159e-01	1.729160e-01
			10%	6.403098e-01	1.958203e-08	6.403098e-01	6.403099e-01
		100	1%	2.919449e-02	4.292731e-09	2.919447e-02	2.919449e-02
			5%	9.882122e-02	6.412320e-09	9.882120e-02	9.882124e-02
			10%	8.840037e-01	1.620393e-08	8.840037e-01	8.840038e-01
		200	1%	1.935374e-02	4.447081e-09	1.935373e-02	1.935376e-02
			5%	1.212993e-01	7.702303e-09	1.212993e-01	1.212993e-01
			10%	2.413499e-01	1.833406e-08	2.413498e-01	2.413499e-01

Note: Statistics on the distance between the best solution found and the actual one are reported.

Table 8 Wilcoxon rank sum hypothesis tests at significance level 0.05 for the results of Table 7

u	Source	Noise	Number of sensors		
			50–100	100–200	50–200
0.0	1	1%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
		5%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
		10%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
	2	1%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
		5%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
		10%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
	3	1%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
		5%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)
		10%	2.562144e-34 (1)	2.562144e-34 (1)	2.562144e-34 (1)

Note: The p -value as well as the acceptance (0) or rejection (1) of the null hypothesis for equal medians are reported.

4.3 Experimental setup for the forward task for inverse use

For our second task, we considered the most promising variant of the experiments for the inverse problem, namely UPSO with $u = 0.1$. The algorithm used the same parameter values with the inverse problem, except the maximum number of iterations, which was increased to 5000 iterations, since the problem's dimension for the new task is increased to eight. The three source points used for the inverse problem were also used here.

For a given source point, the number of sensors varied from $K_{\text{sensor}} = 10$ up to 1000 with increments of 10. For each value of K_{sensor} , an optimisation of the approximated potential function U , defined by equation (2), in the unknown coefficients was conducted by the aforementioned UPSO variant. Each particle of the swarm consisted of eight potential coefficients and it was bounded within $[-100, 100]^8$. Each experiment was repeated five times and the final value of each coefficient was averaged over the five experiments, in order to avoid possible deficiencies due to the algorithm's stochasticity.

For two consecutive values, K_1 and K_2 of K_{sensor} , the absolute relative error between the averaged values of each coefficient was recorded,

$$\varepsilon_{K_1 \rightarrow K_2}^Y = \left| \frac{Y_{K_2} - Y_{K_1}}{Y_{K_1}} \right|,$$

where Y_{K_i} stands for the averaged value of coefficient Y computed using K_i sensors. These errors were statistically analysed to provide information regarding the behaviour of the coefficients under variations of the number of measurements, as it is described in the next section.

4.4 Results for the forward task for inverse use

For each source point, the absolute relative error per coefficient was recorded for varying number of measurements, $K_{\text{sensor}} = 10, 20, 30, \dots, 1000$, as described in the previous section. Thus, for each coefficient, a sample of 99 such errors,

$$\varepsilon_{10 \rightarrow 20}, \varepsilon_{20 \rightarrow 30}, \dots, \varepsilon_{990 \rightarrow 1000},$$

were obtained. The mean, standard deviation, minimum and maximum value of these errors were computed and reported in Table 11.

Table 9 Results of the inverse problem with noise for UPSO with $u = 0.1$

u	Source	K_{sensor}	Noise	Mean	Std	Min	Max
0.1	1	50	1%	$3.540907e-02$	$8.304670e-09$	$3.540906e-02$	$3.540909e-02$
			5%	$2.585456e-01$	$1.037415e-08$	$2.585456e-01$	$2.585457e-01$
			10%	$3.411434e-01$	$2.083517e-08$	$3.411433e-01$	$3.411434e-01$
		100	1%	$3.507655e-02$	$8.130485e-09$	$3.507653e-02$	$3.507657e-02$
			5%	$1.304847e-01$	$1.471345e-08$	$1.304846e-01$	$1.304847e-01$
			10%	$2.825110e-01$	$3.071261e-08$	$2.825110e-01$	$2.825111e-01$
		200	1%	$4.380892e-02$	$6.494797e-09$	$4.380890e-02$	$4.380893e-02$
			5%	$1.865135e-01$	$1.683970e-08$	$1.865134e-01$	$1.865135e-01$
			10%	$5.076858e-01$	$3.024798e-08$	$5.076857e-01$	$5.076859e-01$
	2	50	1%	$1.122122e-02$	$3.263589e-09$	$1.122121e-02$	$1.122123e-02$
			5%	$1.078816e-01$	$1.459702e-08$	$1.078816e-01$	$1.078816e-01$
			10%	$3.795751e-01$	$1.800545e-08$	$3.795751e-01$	$3.795752e-01$
		100	1%	$2.144142e-02$	$5.435641e-09$	$2.144141e-02$	$2.144143e-02$
			5%	$3.166543e-02$	$1.305754e-08$	$3.166541e-02$	$3.166546e-02$
			10%	$1.886913e-01$	$1.840231e-08$	$1.886913e-01$	$1.886914e-01$
		200	1%	$1.145255e-02$	$3.931817e-09$	$1.145254e-02$	$1.145256e-02$
			5%	$1.884985e-02$	$7.649541e-09$	$1.884983e-02$	$1.884987e-02$
			10%	$1.118533e-01$	$1.322483e-08$	$1.118533e-01$	$1.118533e-01$
	3	50	1%	$5.071279e-02$	$4.020203e-09$	$5.071278e-02$	$5.071280e-02$
			5%	$3.136403e-01$	$1.407244e+00$	$1.729159e-01$	$1.424535e+01$
			10%	$6.403098e-01$	$2.299225e-08$	$6.403098e-01$	$6.403099e-01$
		100	1%	$2.919448e-02$	$5.838540e-09$	$2.919447e-02$	$2.919450e-02$
			5%	$9.882122e-02$	$6.413688e-09$	$9.882121e-02$	$9.882124e-02$
			10%	$8.840037e-01$	$1.595273e-08$	$8.840037e-01$	$8.840037e-01$
200		1%	$1.935375e-02$	$6.113328e-09$	$1.935373e-02$	$1.935376e-02$	
		5%	$1.212993e-01$	$8.137813e-09$	$1.212993e-01$	$1.212993e-01$	
		10%	$2.413498e-01$	$2.057229e-08$	$2.413498e-01$	$2.413499e-01$	

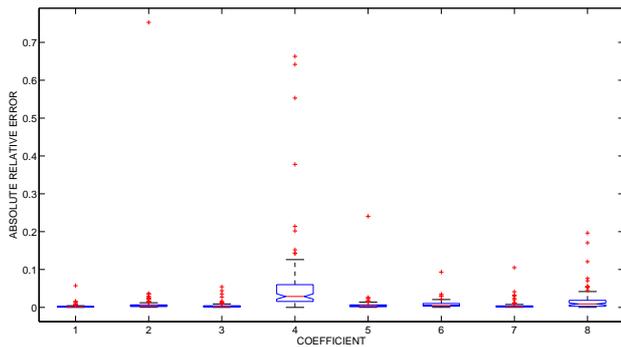
Note: Statistics on the distance between the best solution found and the actual one are reported.

Table 10 Wilcoxon rank sum hypothesis tests at significance level 0.05 for the results of Table 9

u	Source	Noise	Number of sensors		
			50–100	100–200	50–200
0.0	1	1%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
		5%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
		10%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
	2	1%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
		5%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
		10%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
	3	1%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
		5%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)
		10%	2.562144e–34 (1)	2.562144e–34 (1)	2.562144e–34 (1)

Note: The p -value as well as the acceptance (0) or rejection (1) of the null hypothesis for equal medians are reported.

Also, boxplots of the samples are illustrated in Figures 2, 3 and 4, for Sources 1, 2 and 3, respectively. Each boxplot consists of a box and whisker plot, with lines at the lower quartile, median and upper quartile values of the sample. The whiskers are the lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers. The notches at the boxes represent a robust estimate of the uncertainty about the medians for box-to-box comparison. Boxes whose notches do not overlap indicate that the medians of the two groups differ at significance level of 5%. Moreover, line plots for each coefficient are illustrated in Figures 5, 6 and 7, for the Sources 1, 2 and 3, respectively.

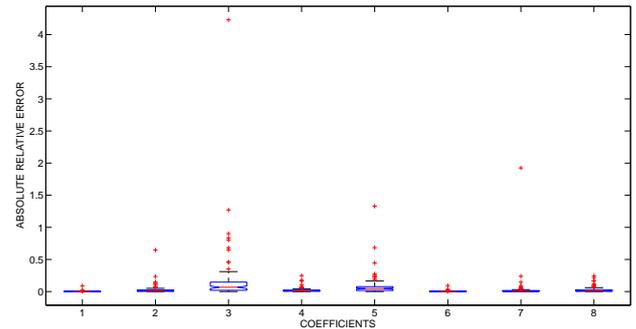
Figure 2 Boxplots for the absolute relative error per coefficient for Source 1 (see online version for colours)

Note: Indices are as follows: 1 – A_1 , 2 – A_2 , 3 – B_1^1 ,

$$4 - B_2^1, 5 - B_2^2, 6 - C_1^1, 7 - C_2^1, 8 - C_2^2$$

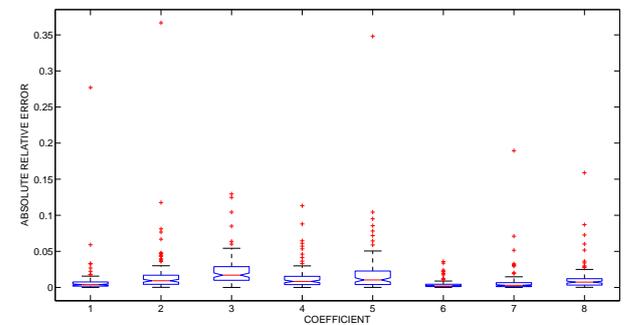
As we observe in the boxplots of Figures 2, 3 and 4, each coefficient exhibits different sensitivity with respect to the magnitude of its error under different number of sensors, depending on the source. For example, coefficient B_2^1 (denoted with index 4) has the widest range of values for Source 1, having also the most outliers (denoted with the

cross symbol beyond the boxplots whiskers), as illustrated in Figure 2. On the other hand, the coefficients B_1^1 and B_2^2 denoted with the indices 3 and 5 have the most wide ranges for Source 2, as depicted in Figure 3.

Figure 3 Boxplots for the absolute relative error per coefficient for Source 2 (see online version for colours)

Note: Indices are as follows: 1 – A_1 , 2 – A_2 , 3 – B_1^1 ,

$$4 - B_2^1, 5 - B_2^2, 6 - C_1^1, 7 - C_2^1, 8 - C_2^2$$

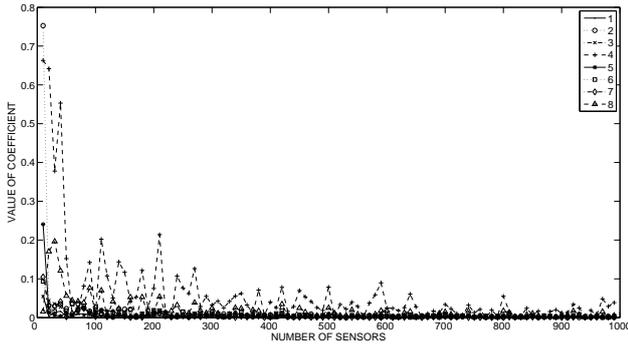
Figure 4 Boxplots for the absolute relative error per coefficient for Source 3 (see online version for colours)

Note: Indices are as follows: 1 – A_1 , 2 – A_2 , 3 – B_1^1 ,

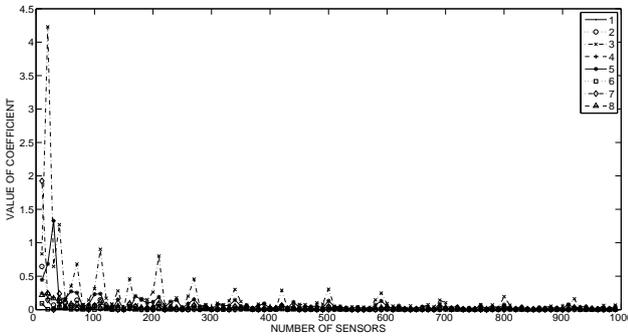
$$4 - B_2^1, 5 - B_2^2, 6 - C_1^1, 7 - C_2^1, 8 - C_2^2$$

Table 11 Statistics for the absolute relative error per coefficient for the three different source points

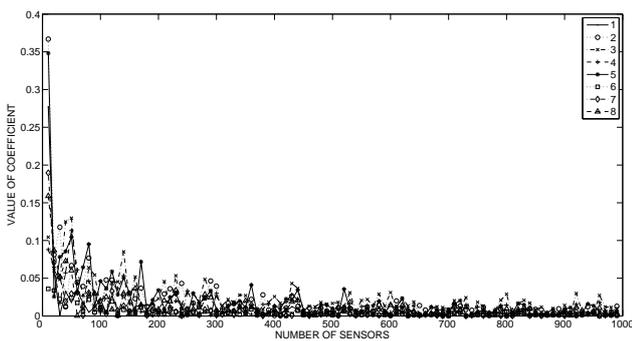
<i>Source</i>	<i>Coefficient</i>	<i>Mean</i>	<i>Std</i>	<i>Min</i>	<i>Max</i>
1	A_1	$2.490319e-03$	$6.120053e-03$	$2.887405e-05$	$5.721209e-02$
	A_2	$1.412178e-02$	$7.501637e-02$	$2.989729e-04$	$7.527117e-01$
	B_1^1	$4.507019e-03$	$8.227689e-03$	$1.547161e-05$	$5.422713e-02$
	B_2^1	$6.287132e-02$	$1.121695e-01$	$7.042640e-06$	$6.629800e-01$
	B_2^2	$7.690445e-03$	$2.408808e-02$	$3.213835e-05$	$2.406614e-01$
	C_1^1	$8.477095e-03$	$1.133411e-02$	$4.504846e-05$	$9.332840e-02$
	C_2^1	$5.001746e-03$	$1.207610e-02$	$3.940336e-05$	$1.048814e-01$
	C_2^2	$1.860749e-02$	$3.042333e-02$	$5.014709e-05$	$1.963787e-01$
	2	A_1	$2.947133e-03$	$9.190543e-03$	$1.585956e-05$
A_2		$2.944760e-02$	$7.183022e-02$	$8.392572e-05$	$6.460923e-01$
B_1^1		$1.769297e-01$	$4.612871e-01$	$2.859028e-04$	$4.229604e+00$
B_2^1		$2.269677e-02$	$3.620677e-02$	$4.442574e-04$	$2.479617e-01$
B_2^2		$8.218983e-02$	$1.571869e-01$	$7.672256e-05$	$1.328881e+00$
C_1^1		$3.979562e-03$	$1.024991e-02$	$2.001097e-06$	$9.261034e-02$
C_2^1		$3.533910e-02$	$1.932437e-01$	$7.775762e-05$	$1.925033e+00$
C_2^2		$2.647546e-02$	$4.287386e-02$	$3.620287e-04$	$2.461121e-01$
3		A_1	$9.095403e-03$	$2.827707e-02$	$3.973026e-06$
	A_2	$1.903460e-02$	$3.994171e-02$	$2.123445e-04$	$3.667366e-01$
	B_1^1	$2.353916e-02$	$2.313453e-02$	$7.602856e-05$	$1.295982e-01$
	B_2^1	$1.439589e-02$	$1.811302e-02$	$2.426462e-05$	$1.132401e-01$
	B_2^2	$2.073995e-02$	$3.909941e-02$	$1.427720e-04$	$3.480031e-01$
	C_1^1	$4.371840e-03$	$6.414704e-03$	$4.601086e-05$	$3.593720e-02$
	C_2^1	$8.203973e-03$	$2.107555e-02$	$7.560453e-05$	$1.895634e-01$
	C_2^2	$1.279361e-02$	$2.051233e-02$	$1.852000e-04$	$1.588921e-01$

Figure 5 Plots for the absolute relative error for all coefficients for Source 1

Note: Indices are as follows: 1 – A_1 , 2 – A_2 , 3 – B_1^1 ,
4 – B_2^1 , 5 – B_2^2 , 6 – C_1^1 , 7 – C_2^1 , 8 – C_2^2

Figure 6 Plots for the absolute relative error for all coefficients for Source 2

Note: Indices are as follows: 1 – A_1 , 2 – A_2 , 3 – B_1^1 ,
4 – B_2^1 , 5 – B_2^2 , 6 – C_1^1 , 7 – C_2^1 , 8 – C_2^2

Figure 7 Plots for the absolute relative error for all coefficients for Source 3

Note: Indices are as follows: 1 – A_1 , 2 – A_2 , 3 – B_1^1 ,
4 – B_2^1 , 5 – B_2^2 , 6 – C_1^1 , 7 – C_2^1 , 8 – C_2^2

Overall, the coefficients C_1^1 and C_2^1 , denoted with indices 6 and 7, respectively, exhibited the most robust behaviour. Figures 5, 6 and 7 reveal also that fluctuations for all coefficients become milder when at least 600 sensors are used, although there is no significant improvement when

their number is further increased up to 1000 sensors. This is an indication that higher number of sensors do not necessarily correspond to more robust approximation through equation (2). Finally, we must notice that also in this problem the position of the source had a crucial impact on the results, in all cases.

5 Conclusions

This paper presented an application of PSO and UPSO to MEG problems. The experimental results in two different tasks revealed the effectiveness of PSO and especially UPSO for tackling source localisation problems as well as best estimating the coefficients of approximations to the magnetic potential function. Furthermore, it provided useful insight regarding the behaviour of the coefficients under variations in the number of sensors where the measurements are recorded.

Naturally, the large amount of data obtained through numerous experiments cannot be fully reported in limited availability of space. For this reason, we selected to analyse only the most interesting and promising approaches. Further investigation is needed to fully reveal the potential of application of these algorithms on MEG problems. Future investigation will contain experiments with real data, as well as the study of these problems under different isolated and clusters of sources.

References

- Abido, M.A. (2002) ‘Optimal design of power system stabilizers using particle swarm optimization’, *IEEE Trans. Energy Conversion*, Vol. 17, No. 3, pp.406–413.
- Agrafiotis, D.K. and Cedeno, W. (2002) ‘Feature selection for structure-activity correlation using binary particle swarms’, *Journal of Medicinal Chemistry*, Vol. 45, No. 5, pp.1098–1107.
- Angeline, P.J. (1998) ‘Evolutionary optimization versus particle swarm optimization: philosophy and performance differences’, in Porto, V.W., Saravanan, N., Waagen, D. and Eiben, A.E. (Eds.): *Evolutionary Programming*, Vol. 7, pp.601–610, Springer.
- Bronzan, J.B. (1971) ‘He magnetic scalar potential’, *American Journal of Physics*, Vol. 39, pp.1357–1359.
- Cai, X., Cui, Z., Zeng, J. and Tan, Y. (2008) ‘Dispersed particle swarm optimization’, *Information Processing Letters*, Vol. 105, No. 6, pp.231–235.
- Cedeno, W. and Agrafiotis, D. (2005) ‘A comparison of particle swarms techniques for the development of quantitative structure-activity relationship models for drug design’, in *IEEE 2005 Computational Systems Bioinformatics Conference*, pp.322–331.
- Clerc, M. and Kennedy, J. (2002) ‘The particle swarm-explosion, stability and convergence in a multidimensional complex space’, *IEEE Trans. Evol. Comput.*, Vol. 6, No. 1, pp.58–73.
- Cui, Z. and Zeng, J. (2004) ‘A guaranteed global convergence particle swarm optimizer’, in *Lecture Notes in Computer Science*, Vol. 3066, pp.762–767, Springer-Verlag.

- Dassios, G. (2006) 'What is recoverable in the inverse magnetoencephalography problem?', *Contemporary Mathematics*, Vol. 408, pp.181–200.
- Dassios, G. (2007) 'The magnetic potential for the ellipsoidal MEG problem', *Journal of Computational Mathematics*, Vol. 25, No. 2, pp.145–156.
- Dassios, G. (2008a) 'Electric and magnetic activity of the brain in spherical and ellipsoidal geometry', in *Lecture Notes from the Minicourse on Mathematics of Emerging Biomedical Imaging*, Springer-Verlag.
- Dassios, G. (2008b) 'Neuronal currents and EEG-MEG fields', *Mathematical Medicine and Biology*, Vol. 25, pp.133–139.
- Dassios, G. and Fokas, A.S. (2008a) 'Electro-magnetoencephalography and fundamental solutions', *Quarterly of Applied Mathematics*.
- Dassios, G. and Fokas, A.S. (2008b) *Electroencephalography and magnetoencephalography: Dipoles and beyond*.
- Dassios, G. and Kariotou, F. (2003a) 'Magnetoencephalography in ellipsoidal geometry', *Journal of Mathematical Physics*, Vol. 44, No. 1, pp.220–241.
- Dassios, G. and Kariotou, F. (2003b) 'On the Geselowitz formula in biomagnetics', *Quarterly of Applied Mathematics*, Vol. 61, No. 2, pp.387–400.
- Dassios, G. and Kariotou, F. (2004) 'On the exterior magnetic field and silent sources in magnetoencephalography', *Abstract and Applied Analysis*, Vol. 4, pp.307–314.
- Dassios, G. and Kariotou, F. (2005) 'The direct MEG problem in the presence of an ellipsoidal shell inhomogeneity', *Quarterly of Applied Mathematics*, Vol. 63, No. 4, pp.601–618.
- Dassios, G., Fokas, A.S. and Hadjiloizi, D. (2007a) 'On the complementarities of electroencephalography and magnetoencephalography', *Inverse Problems*, Vol. 23, pp.2541–2549.
- Dassios, G., Fokas, A.S. and Kariotou, F. (2005) 'On the non-uniqueness of the inverse MEG problem', *Inverse Problems*, Vol. 21, pp.L1–L5.
- Dassios, G., Giapalaki, S., Kandili, A.N. and Kariotou, F. (2007b) 'The exterior magnetic field for the multilayer ellipsoidal model of the brain', *Q. J. Mech. Appl. Math.*, Vol. 60, No. 1, pp.1–25.
- Eberhart, R.C. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', in *Proceedings 6th Symposium on Micro Machine and Human Science*, pp.39–43, IEEE Service Centre, Piscataway, NJ.
- Eberhart, R.C. and Shi, Y. (2000) 'Comparing inertia weights and constriction factors in particle swarm optimization', in *Proc. 2000 IEEE CEC*, pp.84–88, IEEE Service Centre, Piscataway, NJ.
- Fokas, A.S., Gelfand, I.M. and Kurylev, Y. (1996) 'Inversion method for magnetoencephalography', *Inverse Problems*, Vol. 12, pp.L9–L11.
- Fokas, A.S., Kurylev, Y. and Marinakis, V. (2004) 'The unique determination of neuronal currents in the brain via magnetoencephalography', *Inverse Problems*, Vol. 20, pp.1067–1082.
- Forghani, N., Forouzanfar, M. and Forouzanfar, E. (2007) 'MRI fuzzy segmentation of brain tissue using IFCM algorithm with particle swarm optimization', in *Proc. 22nd Int. Symp. Computer and Information Sciences (ISCIS)*, pp.1–4.
- Fourie, P.C. and Groenwold, A.A. (2002) 'The particle swarm optimization algorithm in size and shape optimization', *Struct. Multidisc. Optim.*, Vol. 23, pp.259–267.
- Georgiou, V.L., Pavlidis, N.G., Parsopoulos, K.E., Alevizos, P.D. and Vrahatis, M.N. (2006) 'New self adaptive probabilistic neural networks in bioinformatics and medical tasks', *International Journal on Artificial Intelligence Tools*, Vol. 15, No. 3, pp.371–396.
- Geselowitz, D.B. (1970) 'On the magnetic field generated outside an inhomogeneous volume conductor by internal current sources', *IEEE Trans. Magn.*, Vol. 6, pp.346–347.
- Grynspan, F. and Geselowitz, D.B. (1973) 'Model studies of the magnetocardiogram', *Biophysical Journal*, Vol. 13, pp.911–925.
- Hamalainen, M., Hari, R., Ilmoniemi, R.J., Knuutila, J. and Lounasmaa, O. (1993) 'Magnetoencephalography theory, instrumentation and applications to non-invasive studies of the working human brain', *Review of Modern Physics*, Vol. 65, pp.413–497.
- Ilmoniemi, R.J., Hamalainen, M.S. and Knuutila, J. (2005) 'The forward and inverse problems in the spherical model', in *Biomagnetism: Applications and Theory*, pp.278–282, Pergamon Press.
- Kennedy, J. (1999) 'Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance', in *Proc. IEEE Congr. Evol. Comput.*, pp.1931–1938, IEEE Press, Washington, D.C., USA.
- Kennedy, J. and Eberhart, R.C. (1995) 'Particle swarm optimization', in *Proc. IEEE Int. Conf. Neural Networks*, Vol. 4, pp.1942–1948, IEEE Service Centre, Piscataway, NJ.
- Kennedy, J. and Eberhart, R.C. (2001) *Swarm Intelligence*, Morgan Kaufmann Publishers.
- Kotsireas, I.S., Koukouvinos, C., Parsopoulos, K.E. and Vrahatis, M.N. (2006) 'Unified particle swarm optimization for Hadamard matrices of Williamson type', in *Proceedings of the 1st International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS)*, pp.113–121, Beijing, China.
- Liu, Y., Zhou, J. and Chen, Y. (2008) 'Ensemble classification for cancer data', in *Proc. 2008 Int. Conf. Biomedical Engineering and Informatics (BMEI)*, pp.269–273.
- Matyas, J. (1965) 'Random optimization', *Automatization and Remote Control*, Vol. 26, pp.244–251.
- Mendes, R., Kennedy, J. and Neves, J. (2004) 'The fully informed particle swarm: simpler, maybe better', *IEEE Trans. Evol. Comput.*, Vol. 8, No. 3, pp.204–210.
- Millonas, M.M. (1994) 'Swarms, phase transitions and collective intelligence', in Palaniswami, M., Attikiouzel, Y., Marks, R., Fogel, D. and Fukuda, T., (Eds.): *Computational Intelligence: A Dynamic System Perspective*, pp.137–151, IEEE Press, Piscataway, NJ.
- Mohamed, O. and Adel, A. (2006) 'Particle swarm optimization based stroke volume influence on mean arterial pressure', in *Proc. Int. Conf. Biomedical and Pharmaceutical Engineering (ICBPE)*, pp.508–512.
- Nakib, A., Roman, S., Oulhadj, H. and Siarry, P. (2007) 'Fast brain MRI segmentation based on two-dimensional survival exponential entropy and particle swarm optimization', in *Proc. 29th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBS)*, pp.5563–5566.

- Nolte, G. and Curio, G. (1997) 'On the calculation of magnetic fields based on multipole modelling of focal biological current sources', *Biophysical Journal*, Vol. 73, pp.1253–1262.
- Ourique, C.O., Biscaia, E.C. and Carlos Pinto, J. (2002) 'The use of particle swarm optimization for dynamical analysis in chemical processes', *Computers and Chemical Engineering*, Vol. 26, pp.1783–1793.
- Papageorgiou, E.I., Parsopoulos, K.E., Stylios, C.D., Groumpos, P.P. and Vrahatis, M.N. (2005) 'Fuzzy cognitive maps learning using particle swarm optimization', *Journal of Intelligent Information Systems*, Vol. 25, No. 1, pp.95–121.
- Parsopoulos, K.E. and Vrahatis, M.N. (2002a) 'Initializing the particle swarm optimizer using the nonlinear simplex method', in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, pages 216–221.
- Parsopoulos, K.E. and Vrahatis, M.N. (2002b) 'Recent approaches to global optimization problems through particle swarm optimization', *Natural Computing*, Vol. 1, No. 2–3, pp.235–306.
- Parsopoulos, K.E. and Vrahatis, M.N. (2004a) 'On the computation of all global minimisers through particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.211–224.
- Parsopoulos, K.E. and Vrahatis, M.N. (2004b) 'UPSO: a unified particle swarm optimization scheme', in *Lecture Series on Computer and Computational Sciences, Vol. 1, Proceedings of the International Conference of Computational Methods in Sciences and Engineering (IC-CMSE)*, pp.868–873, VSP International Science Publishers, Zeist, Netherlands.
- Parsopoulos, K.E. and Vrahatis, M.N. (2005a) 'Unified particle swarm optimization for solving constrained engineering optimization problems', in Wang, L., Chen, K. and Ong, Y.S. (Eds.): *Lecture Notes in Computer Science (LNCS)*, Vol. 3612, pp.582–591, Springer-Verlag.
- Parsopoulos, K.E. and Vrahatis, M.N. (2005b) 'Unified particle swarm optimization for tackling operations research problems', in *Proceedings of the IEEE 2005 Swarm Intelligence Symposium*, pp.53–59, Pasadena, CA, USA.
- Parsopoulos, K.E. and Vrahatis, M.N. (2005c) 'Unified particle swarm optimization in dynamic environments', in Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D. and Squillero, G. (Eds.): *Lecture Notes in Computer Science (LNCS)*, Vol. 3449, pp.590–599, Springer-Verlag.
- Parsopoulos, K.E. and Vrahatis, M.N. (2006) 'Studying the performance of unified particle swarm optimization on the single machine total weighted tardiness problem', in Sattar, A. and Kang, B.H. (Eds.): *Lecture Notes in Artificial Intelligence (LNAI)*, Vol. 4304, pp.760–769, Springer.
- Parsopoulos, K.E. and Vrahatis, M.N. (2007) 'Parameter selection and adaptation in unified particle swarm optimization', *Mathematical and Computer Modelling*, Vol. 46, No. 1–2, pp.198–213.
- Pavlidis, N.G., Parsopoulos, K.E. and Vrahatis, M.N. (2005) 'Computing Nash equilibria through computational intelligence methods', *Journal of Computational and Applied Mathematics*, Vol. 175, pp.113–136.
- Petalas, Y.G., Parsopoulos, K.E. and Vrahatis, M.N. (2008) 'Stochastic optimization for detecting periodic orbits of nonlinear mappings', *Nonlinear Phenomena in Complex Systems*, Vol. 11, No. 2, pp.285–291.
- Petalas, Y.G., Parsopoulos, K.E. and Vrahatis, M.N. (2009) 'Improving fuzzy cognitive maps learning through memetic particle swarm optimization', *Soft Computing*, Vol. 13, No. 1, pp.77–94.
- Plonsey, R. and Hepner, D.B. (1967) 'Considerations of quasistationarity in electrophysiological systems', *Bulletin of Mathematical Biophysics*, Vol. 29, pp.657–664.
- Qiu, L., Li, Y. and Yao, D. (2005) 'A feasibility study of EEG dipole source localization using particle swarm optimization', in *Proceedings of the IEEE 2005 Congress on Evolutionary Computation*, Vol. 1, pp.720–726, IEEE Press.
- Ray, T. and Liew, K.M. (2002) 'A swarm metaphor for multiobjective design optimization', *Engineering Optimization*, Vol. 34, No. 2, pp.141–153.
- Saldam, A., Ahmad, I. and Al-Madani, S. (2002) 'Particle swarm optimization for task assignment problem', *Microprocessors and Microsystems*, Vol. 26, pp.363–371.
- Sarvas, J. (1987) 'Basic mathematical and electromagnetic concepts of the biomagnetic problem', *Physics in Medicine and Biology*, Vol. 32, pp.11–22.
- Skokos, C., Parsopoulos, K.E., Patsis, P.A. and Vrahatis, M.N. (2005) 'Particle swarm optimization: an efficient method for tracing periodic orbits in 3D galactic potentials', *Monthly Notices of the Royal Astronomical Society*, Vol. 359, pp.251–260.
- Suganthan, P.N. (1999) 'Particle swarm optimizer with neighborhood operator', in *Proc. IEEE Congr. Evol. Comput.*, pp.1958–1961, Washington, D.C., USA.
- Trelea, I.C. (2003) 'The particle swarm optimization algorithm: convergence analysis and parameter selection', *Information Processing Letters*, Vol. 85, pp.317–325.
- Wachowiak, M.P., Smolikova, R., Zheng, Y., Zurada, J.M. and Elmaghraby, A.S. (2004) 'An approach to multimodal biomedical image registration utilising particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.289–301.
- Xie, L. and Jiang, L. (2005) 'Global optimal ICA and its application in brain MEG data analysis', in *Proceedings of the 2005 Int. Conf. Neural Networks and Brain*, pp.353–357.
- Xu, C., Zhang, Q., Wang, B. and Zhang, R. (2008) 'Research on the DNA sequence design based on GA/PSO algorithms', in *Proc. 2nd Int. Conf. Bioinformatics and Biomedical Engineering (ICBBE)*, pp.816–819.
- Yang, C-S., Chuang, L-Y., Li, J-C. and Yang, C-H. (2008) 'A novel BPSO approach for gene selection and classification of microarray data', in *Proc. IEEE 2008 Int. Joint Conf. Neural Networks (IJCNN)*.
- Zhang, X. and Li, T. (2007) 'Improved particle swarm optimization algorithm for 2D protein folding prediction', in *Proc. 1st Int. Conf. Bioinformatics and Biomedical Engineering (ICBBE 2007)*, pp.53–56.