IMPROVING PSO BASED ALGORITHMS WITH THE DOMAIN-SHRINKING TECHNIQUE FOR ELECTROMAGNETIC DEVICES OPTIMIZATION

Laurentiu DUCA[1], Cornel POPESCU[2]

In this paper, the authors present PSO based algorithms used to optimize the electromagnetic device of an international test problem. Several variants of PSO based algorithms are implemented. These are improved variants of Standard PSO (SPSO), Discrete PSO (DPSO) and Quantum PSO (QPSO). A domain-shrinking technique is used to improve the results of these algorithms when optimizing the electromagnetic device of the TEAM22 benchmark problem.

Keywords: optimization, SPSO, DPSO, QPSO, domain-shrinking, TEAM22, electromagnetic device

1. Introduction

Because of the real-life interest in solving many optimization problems, it was developed a wide scientific literature on the subject. Google Scholar reports for the term "optimization" about four million scientific papers, from which about one million in the past ten years, which proves the importance and the topicality of the subject. In present, the optimizations are considered part of Math, Computer Science and Operational Research Science. No matter the field, improvement of a solution supposes solving of an optimization problem. All high-tech products are the result of an optimization action.

The optimization problems are classified in two major categories: without restrictions [1] and with restrictions [2]. These may be convex (with a single minimum [3]) or not (with multiple local minimum). In the convex case, the objective function is smooth and differentiable, case that is exploited by the solving methods. In the non-convex case, the function is not smooth which makes the optimization problem far more difficult.

The solving methods for this kind of problems are classified in two major categories: deterministic and stochastic. The deterministic methods may be with or without gradient use [4] or use of another higher order derivatives of the objective function. These methods cannot solve problems with multiple local minima, because these incline to a local minimum. For determining global

[1] Teaching assistant, Automatics and Computers Faculty, University POLITEHNICA of Bucharest, e-mail: laurentiu.duca@cs.pub.ro

[2] Associate professor, Automatics and Computers Faculty, University POLITEHNICA of Bucharest, e-mail: cornel.popescu@cs.pub.ro

minimum are used different stochastic methods which apparently seem to chaotic explore the search space, but these give the chance to the algorithm to find the global minimum [5]. These methods are of metaheuristic type and do not guarantee to find the optimal solution but find a pseudo optimal one [6]. Most heuristics are inspired from nature and are classified in “single-solution” and “population based”. The second class is suitable to parallelization.

In many cases, the optimization of electromagnetic devices is an optimization of their geometric dimensions and of the position and the values of the electromagnetic field sources such that some objectives are satisfied [7] [8]. The difficult part is made by the complexity of the objective function which has many local minima and the fact that the evaluation of the objective function implies numerical solving of an electromagnetic field problem which requires important computing resources like processor and memory.

The main disadvantage of stochastic methods is a large number of objective function evaluations, especially when the cost of the objective function evaluation is significant. In this case, the running time of sequential implementation is too high, hence the need for algorithm parallelization. The technological evolution regarding transistor shrinking brought a limitation in the processors working frequency as a cause of the difficulty to extract the generated heat. The alternative is represented by parallel architectures grouped in multi-core clusters or GP-GPU devices which contain hundreds of cores if these are efficiently exploited by use of parallel algorithms [9] [10] [11].

The PSO (Particle Swarm Optimization) algorithms [12] are iterative stochastic optimization methods, which use a population of candidate solutions which evolves in time. These algorithms are independent of the problem to be solved and are appropriate to difficult optimization problems when the derivative of the objective function is unknown. The main goal of this paper is to propose innovative variants that improve performances of the PSO algorithms because these algorithms are well suited to parallelization.

2. The TEAM22 benchmark problem

SMES devices (Superconducting Magnetic Energy Storage) store the energy in magnetic fields and are made by solenoids manufactured from superconducting materials. The TEAM22 problem consists in optimization of such a SMES device where the solenoids are powered by a power converter switch. The switch is simultaneously opened with the shorting of the coils terminals and the current will flow through the coils without decreasing in time, cause of the superconducting resistance which is almost zero. These devices are used in energy systems to stabilize the power fluctuations [7].

In the TEAM22 problem, a SMES device (Fig. 1) must be optimized such that the following objectives are achieved [13]:

- the stored energy in the device is about 180 MJ;
- in the interior of the coils it must be met the condition of critical magnetic field which guarantees superconductivity;
- the stray field (measured at 10 meters distance from the device) should be as small as possible.

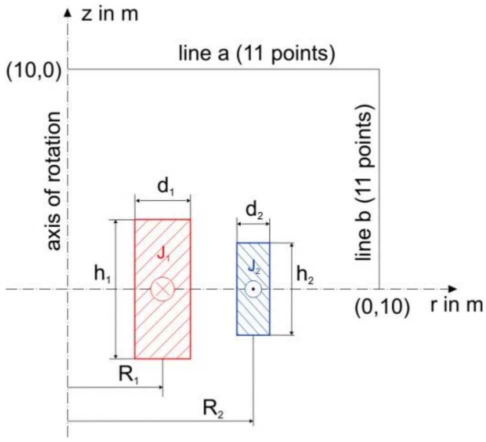

Fig. 1. SMES device with two solenoids [14]

The problem has 8 parameters (R1, R2, h1/2, h2/2, d1, d2, J1, J2) which have restrictions that are presented in table 1. These parameters must be computed by solving the optimization problem.

Table 1

Parameters restrictions for the TEAM22 problem

	R1 [m]	R2 [m]	h1/2 [m]	h2/2 [m]	d1 [m]	d2 [m]	J1 [MA/m^2]	J2 [MA/m^2]
min	1.0	1.8	0.1	0.1	0.1	0.1	10.0	-30.0
max	4.0	5.0	1.8	1.8	0.8	0.8	30.0	-10.0

The coils should not overlap each other, so a new design constraint must be met:

$$R_1 + \frac{d_1}{2} < R_2 - \frac{d_2}{2} \qquad (1)$$

The superconducting material must meet the quench condition which consists in a relation between the current density and the maximum value of magnetic flux density, condition shown in Fig. 2.

Equation (2) is an approximation of the curve in Fig. 2:

$$|\mathbf{J}| \leq (-6.4|\mathbf{B}| + 54.0) \quad \text{A/mm}^2 \qquad (2)$$

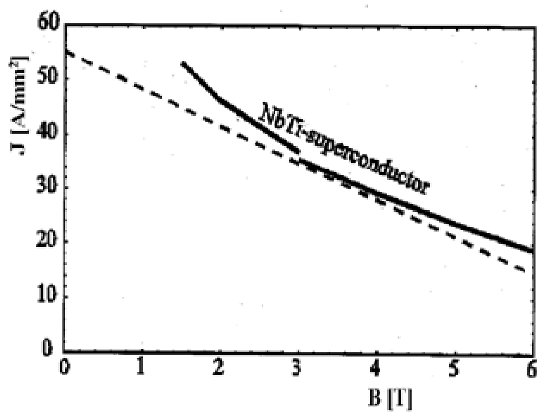

Fig. 2. Curve of the industrial superconductor [14]

The objective function of this problem must take into account the energy condition (the stored energy in the device is about 180 MJ) and the condition that the stray field should be as small as possible; thus, the problem is reduced to six parameters instead of eight. The objective function that must be minimized by solving the optimization problem is:

$$F = \frac{B_{stray}^2}{B_{norm}^2} + \frac{|E - E_{ref}|}{E_{ref}} \tag{3}$$

In equation (3), $E_{ref} = 180$ MJ and $B_{norm} = 2.0 \cdot 10^{-4}$T.

The B_{stray}^2 value is computed by evaluating the field in 22 equidistant points on line a and line b from Fig. 1 and has the expression:

$$B_{stray}^2 = \frac{\sum_{i=1}^{22} |B_{stray,i}|^2}{22} \tag{4}$$

For the objective function of the reference problem TEAM22, the minimum reported in the definition of the problem is 1.8E-03 [14]. Results close to this value were reported in [15], [16], [17]. In [15], the author uses a distributed evolutionary strategy which runs on a ten node network, each node computing 6000 objective function evaluations. In [16], it is implemented a version of the IPSO algorithm based on a *tabu* region (an interdicted region for the parameters) after keeping constant four parameters in the already computed minimum point obtained by [15] and varying two parameters. In [17], the author implements PSO and QPSO algorithms for a population of 30 particles and 200 iterations, a total of 6000 evaluations of the objective function. In this paper, initially the search domain it is considered unknown and all parameters vary conforming to the implemented algorithms.

3. PSO algorithms

Particle swarm optimization was invented by Kennedy and Eberhart and, at the beginning, the scope was to simulate the social behavior as a representation of the movement of a flock of birds or school of fish [8]. PSO is a stochastic optimization algorithm and does not require the objective function of the test problem to be differentiable. The algorithm consists of a population of individuals (of size *S*) which are named particles. These particles fly (move) in the search space (domain of definition of size *D*), of rectangular form or classified in a cuboid with the limits x_{min} and x_{max} of the objective function. Each particle is defined by its position in the search space, denoted by x_i and speed, denoted by v_i. At each iteration of the algorithm, the particles modify their position and speed based on two values: the personal best position (p_i) and the global best position (g) from the history of the swarm. The following equations describe the expression of speed and position for a given particle at the next iteration time.

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_p \cdot r_p \cdot (p_{i,j}(t) - x_{i,j}(t)) + c_g \cdot r_g \cdot (g_j(t) - x_{i,j}(t)) \quad (5)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t) \quad (6)$$

The three weights (inertial weight w, personal weight c_p and social weight c_g) used in the speed equation describe the behavior of the individuals and of the hole swarm which aspire to go to the best solution. The parameters w, c_p, c_g are selected by the algorithm developer and controls the behavior and effectiveness of the PSO method. An example of values for these parameters, for the function $f = 3 + x_0^2 + x_1^2$ is w=0.73, c_p=1.49, c_g=1.49 [18]. The variables r_p and r_g are randomly distributed in the range (0,1).

The classic PSO algorithm for different types of problems has the tendency to get stuck, by agglomerating whole swarm, in a local minimum zone. Different variants of the PSO algorithm try to avoid premature convergence [19] [20] [21].

Until today, the PSC (Particle Swarm Central) authority has made public three variants of SPSO algorithm: SPSO 2006, SPSO 2007 and SPSO 2011 [22]. Each standard is a better version of the previous one.

SPSO keeps the ideea of the particle swarm and the basis from PSO but considers that particles are connected, each connection representing a link between two different particles. A connection has an informed and an informing particle, the first particle knowing the personal best and the position of the second particle. Thus, each informed particle has a set of informing particles called neighborhood.

SPSO uses a random topology which changes the connections graph at each unsuccessful iteration (when the global best solution is not improved). The graph of links between particles is created this way: each particle informs three randomly chosen particles; so, a particle will be informed by a number from 1 to *S* particles. The velocity formula introduces a new term, the center of gravity, for obtaining “exploration” and “exploitation”. The center of gravity depends on three terms: the current position, a term relative to the previous best p_i, and a term relative to the previous best in the neighborhood l_i.

Let $N_i(t)$ be the set of neighbors of the particle i at the moment t. Initialization in SPSO 2006 and SPSO 2007 is done like PSO with the addition that initialization of l_i is made:

$$l_i\ (0) = argmin_{j \in Ni(0)}(f(p_j(0))) \tag{7}$$

In SPSO 2011, initialization is the same, except the speed initialization and it is avoided that the particles leave the search space when the dimension D of the definition domain is big [22]. The speed new formula is:

$$v_{i,j}(0) = random\ (x_{min,j} - x_{i,j}(0),\ x_{max,j} - x_{i,j}(0)) \tag{8}$$

The equations which update the speed and positions from SPSO 2006 and SPSO 2007 have problems in finding the minimum when this is placed on an axis, on a diagonal or in the center of the coordinate system. In SPSO 2011, the speed is modified in a way that does not depend by the coordinate system. Let G_i being the center of gravity for the particle i. The equation of G_i is:

$$G_i = x_i + c(p_i+l_i-2x_i)/3 \tag{9}$$

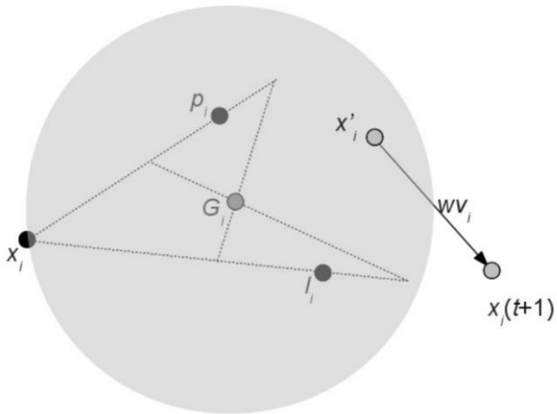

Fig. 3. Computing new position in SPSO 2011 [22]

Let $H_i(G_i, ||G_i\text{-}x_i||)$ be the hypersphere of center G_i and radius $||G_i\text{-}x_i||$, and x_i' a random point in this hypersphere. The update equations of speed and position are:

$$v_i(t+1) = w \cdot v_i(t) + x_i'(t) - x_i(t) \tag{10}$$

$$x_i(t+1) = w \cdot v_i(t) + x_i'(t) \tag{11}$$

The scope of the SPSO algorithm authors was to keep the principles of the classic PSO algorithm with minimum dependencies in the implementation.

The SPSO equations cannot be used to generate binary/discrete values because the positions are vectors of real values. In order to obtain discrete (binary) values, the resulted values of each iteration must be rounded.

Pan et al. [23] has presented a discrete PSO optimization algorithm (DPSO) which can be used for continuous problems if a finite precision for the optimization variables is imposed. In this algorithm the positions of the particles are given by the following equations:

$$x_i(t+1) = p_{GB} \oplus CR(p_{PB} \oplus CR(x_{iM}(t), p_i(t)), g) \tag{12}$$

$$x_{iM}(t) = p_v \oplus M(x_i(t)) \tag{13}$$

M is a mutation operator applied with the probability p_v. It is generated a random number r in (0,1) and if $r > p_v$ then it will be applied the mutation operator on $x_i(t)$: it is chosen a coordinate and it will be generated a random value for it. CR is a crossover operator and it is applied twice. First time is applied with probability p_{PB}: it is generated a random number r in (0,1) and if $r > p_{PB}$, for the variables $x_{iM}(t)$ and $p_i(t)$ it is chosen a coordinate j for which it will be interchanged the values of the two variables. Second time, the operation is similar, but with probability p_{GB}.

The algorithm developed by Pan et al. has an innovation degree by setting a link between PSO and genetic algorithms.

The main disadvantage of PSO is that it cannot guarantee its global convergence [24]. In order to face this problem, a method that increase the probability to obtain global convergence named Quantum PSO (QPSO) has been developed and published in [25].

Unlike PSO and SPSO algorithms, where the particle trajectories are according to Newton's mechanic laws, QPSO is a quantum system. In QPSO, the particle position does not depend on speed, like PSO. The particles move conforming to the following equations:

$$x_i(t+1) = pl_i(t) \pm \beta \cdot |m(t) - x_i(t)| \cdot \ln(1/u) \tag{14}$$

$$m(t) = \frac{1}{S}\sum_{i=1}^{S} p_i(t) \tag{15}$$

$$pl_i(t) = \varphi \cdot p_i(t) + (1-\varphi) \cdot g(t) \tag{16}$$

The parameter β is the contraction-expansion factor which adjusts the convergence speed of the algorithm. The parameters u and φ are random numbers uniformly distributed in (0,1). In the x_i equation, there is a hidden parameter k which is random in (0,1) and if $k > 0.5$ the operation is add, else the operation is subtract.

While in the PSO algorithm the particles converge to the solution through the global best position, in QPSO the particles exert a greater influence on each other through an average of the personal best positions, so the probability to get stuck in a local minimum is smaller.

4. Results

We have chosen the SPSO, QPSO and DPSO algorithms for the TEAM22 problem which were implemented in [26] for different sizes of the swarm (32, 64 and 128). For a relevant statistical study, it were made 30 runs for each algorithm and swarm size, at each time, the initialization being made with different values of the populations randomly generated in the search space. The algorithms were stopped when it was reached the maximum number of iterations, a number equivalent to 2560 function evaluations.

Table 2

The objective function values and standard deviation of the PSO algorithms applied to the TEAM22 problem [26]

Algorithm	Swarm size	Min	Max	Mean	Standard deviation
SPSO	32	2.98 E-3	15.99 E-3	5.82 E-3	2.58 E-3
	64	3.41 E-3	7.95 E-3	**5.41 E-3**	**1.16 E-3**
	128	3.42 E-3	9.50 E-3	6.41 E-3	1.42 E-3
QPSO	32	**2.23 E-3**	27.10 E-3	8.11 E-3	6.46 E-3
	64	2.49 E-3	26.50 E-3	5.95 E-3	4.80 E-3
	128	2.92 E-3	14.30 E-3	7.07 E-3	3.05 E-3
DPSO	32	6.70 E-3	37.30 E-3	17.32 E-3	9.27 E-3
	64	3.60 E-3	34.02 E-3	15.02 E-3	9.31 E-3
	128	4.80 E-3	52.07 E-3	17.62 E-3	13.19 E-3

The results of the performance study from [26] are presented in table 2 which contain the mean of the minimum values obtained at each from the 30 program executions. Min and Max represent the minimum and maximum of the minimum values obtained from the 30 program executions. The QPSO and SPSO have better results than DPSO. The smallest values of the objective function were obtained by the QPSO algorithm while the smallest mean and standard deviation were obtained by SPSO. The optimum swarm size for QPSO and SPSO is between 32 and 64 particles.

The solution improvements from our study consist in shrinking the domain of the parameters after the algorithm has run a number of iterations and obtained values considerably better than the initial moment. The algorithms selected for this type of improvement are SPSO, QPSO and DPSO, and the variants where the domain was shrank are 1, which means no shrinking, 1/4 – the variant to which it was made domain shrinking only the variables R_1 and R_2, and 1/64 – the variant to which it was made domain shrinking for all the parameters (domain shrinking is made for all the six variables after 30% of iterations).

In Table 3 is shown the improvement applied to the SPSO algorithm for populations of 32, 64 and 128 particles.

Table 3

Results of the improved SPSO algorithms by domain-shrinking for the TEAM22 problem

Algorithm	Swarm size	Min	Max	Mean	Standard deviation
SPSO 1/4	32	2.54E-3	12.34E-3	4.32E-3	2.21E-3
SPSO 1/4	64	3.01E-3	6.53E-3	3.92E-3	0.95E-3
SPSO 1/4	128	3.12E-3	7.42E-3	4.28E-3	0.99E-3
SPSO 1/64	32	2.33E-3	9.19E-3	3.75E-3	1.24E-3
SPSO 1/64	64	2.82E-3	5.70E-3	3.69E-3	0.64E-3
SPSO 1/64	128	2.87E-3	5.90E-3	3.93E-3	0.68E-3

In Table 4 is shown the improvement applied to the QPSO algorithm for populations of 32, 64 and 128 particles.

Table 4

Results of the improved QPSO algorithms by domain-shrinking for the TEAM22 problem

Algorithm	Swarm size	Min	Max	Mean	Standard deviation
QPSO 1/4	32	2.13E-3	8.82E-3	4.54E-3	1.64E-3
QPSO 1/4	64	2.48E-3	14.67E-3	5.02E-3	2.55E-3
QPSO 1/4	128	2.67E-3	15.79E-3	6.03E-3	3.20E-3
QPSO 1/64	32	2.02E-3	3.27E-3	2.49E-3	0.30E-3
QPSO 1/64	64	2.09E-3	3.96E-3	2.77E-3	0.51E-3
QPSO 1/64	128	2.11E-3	4.08E-3	2.83E-3	0.52E-3

In Table 5 is shown the improvement applied to the DPSO algorithm for populations of 32, 64 and 128 particles.

Table 5

Results of the improved DPSO algorithms by domain-shrinking for the TEAM22 problem

Algorithm	Swarm size	Min	Max	Mean	Standard deviation
DPSO 1/4	32	2.98E-3	37.09E-3	11.49E-3	8.43E-3
DPSO 1/4	64	3.80E-3	30.32E-3	10.22E-3	7.11E-3
DPSO 1/4	128	3.24E-3	29.96E-3	9.11E-3	6.34E-3
DPSO 1/64	32	2.62E-3	26.10E-3	10.00E-3	5.75E-3
DPSO 1/64	64	3.31E-3	27.16E-3	6.51E-3	2.34E-3
DPSO 1/64	128	3.16E-3	11.71E-3	6.38E-3	2.38E-3

The domain-shrinking solutions improvements are proven to be satisfying for all algorithms SPSO, QPSO and DPSO. Best results are obtained, in order, by the variants QPSO 1/64 (minimum of 2.02E-3 in the point given by R_1=1.305600m, d_1=0.520453m, h_1=1.1410211·2m, R_2=1.800000m,

d_2=0.203861m, h_2=1.552848·2m), SPSO 1/64, QPSO 1/4, SPSO 1/4, DPSO 1/64, DPSO 1/4. The DPSO algorithms give the weakest performances from the three categories (SPSO, QPSO and DPSO).

For all algorithms, the variants 1/64 are proven to be better than 1/4. In Fig. 4 it is shown the evolution of the mean obtained by the best configurations of the SPSO, QPSO and DPSO algorithms with the domain-shrinking technique.

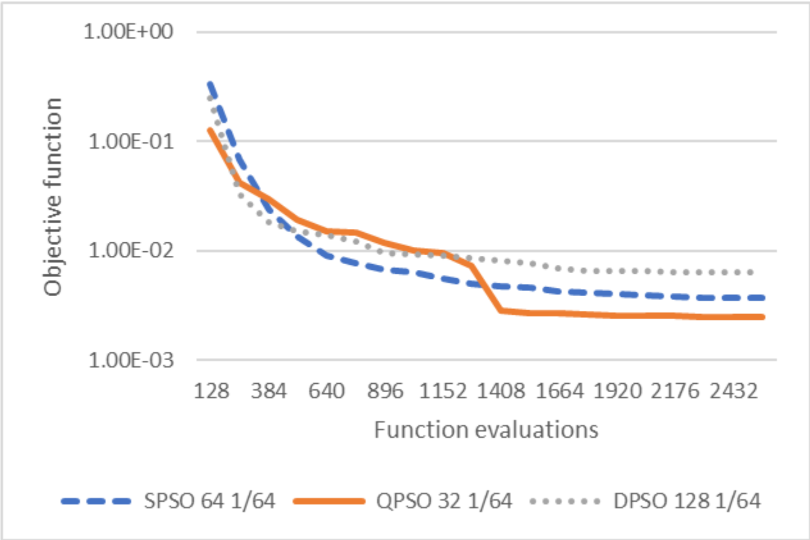

Fig. 4. Evolution of the mean obtained by the best 1/64 configurations of SPSO, QPSO and DPSO applied to the TEAM22 problem

5. Conclusion

In this paper were presented PSO based algorithms and domain-shrinking technique used for electromagnetic devices optimization. This technique was applied to the PSO based algorithms SPSO, DPSO and QPSO which were used to optimize the electromagnetic device of an international test problem, namely TEAM22.

The domain-shrinking technique significantly improves the performance of all the three algorithms. SPSO 1/4, QPSO 1/4 and DPSO 1/4 have better results than SPSO, QPSO and respectively DPSO. Going further, SPSO 1/64, QPSO 1/64 and DPSO 1/64 have better performances than SPSO 1/4, QPSO 1/4 and DPSO 1/4. The best result was obtained for QPSO 1/64 which is close to the minimum point of the TEAM22 objective function.

Acknowledgement

This work was funded by the Romania – Belgium (RO-BE) bilateral project, EchoMEMS, PN-III-CEI-BIM-PBE, 2017 (id 98 BM).

R E F E R E N C E S

[1] *R. Horst, H. Tuy*, “Global Optimization: Deterministic Approaches”, Springer, 1996.

[2] *D. P. Bertsekas*, “Constrained Optimization and Lagrange Multiplier Methods”, Academic Press, 2014.

[3] *S. Boyd, L. Vandenberghe*, “Convex Optimization”, Cambridge University, 2004.

[4] *A. R. Conn, K. Scheinberg, L.N. Vicente*, “Introduction to Derivative-Free Optimization”, MPS-SIAM, 2009.

[5] *J.C. Spall*, “Introduction to Stochastic Search and Optimization”, Wiley, ISBN 0-471-33052-3, 2003.

[6] *K. Sörensen*, M. Sevaux, F. Glover, "A History of Metaheuristics", University of Colorado, 2015.

[7] *G. Ciuprina*, “Studiul câmpului electromagnetic în medii neliniare - contribuții privind optimizarea dispozitivelor electromagnetice” (Study of the electromagnetic field in non-linear medium – contributions regarding electromagnetic devices optimization), PhD thesis, UPB, 1999.

[8] *A. Duca, M. Rebican, L. Janousek, M. Smetana, and T. Strapacova*, “PSO based techniques for NDT-ECT inverse problems”, in Studies in Applied Electromagnetics and Mechanics, vol. 39, IOS Press, pp. 323-330, 2014.

[9] *A. Duca, and F.M.G. Tomescu*, “A Distributed Hybrid Optimization System for NDET Inverse Problems”, in The Proceedings of the International Symposium of Nonlinear Theory and its Applications (NOLTA), pp. 1059-1062. Bologna, Italy, 2006.

[10] *O.T. Altinoz, E. Akca, A.E. Yilmaz, A. Duca, G. Ciuprina*, Parallel Implementation of Desirability Function-Based Scalarization Approach for Multiobjective Optimization Problems. Informatica (Slovenia), vol. 39(2), pp. 115-123, 2015.

[11] *M. Rebican, L. Janousek, M. Smetana, A. Duca*, Diagnosis of Real Cracks from Eddy Current Testing Signals using Parallel Computation, in Journal of Nondestructive Testing and Evaluation, pp. 435-443, vol. 32(4), Taylor & Francis, 2017.

[12] *J. Kennedy, R. Eberhart*, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks IV, pp. 1942–1948, 1995.

[13] *C. Magele, G. Fuerntratt, I. Bardi, K.R. Richter, G. Schoenwetter, P. Alotto, G. Molinari, and M. Repetto*, "SMES Optimization Benchmark - Preliminary TEAM Workshop Problem 22", TEAM Workshop, Okayama, pages 38–48, 1996.

[14] *P.G. Alotto, U. Baumgartner, F. Freschi, M. Jaindl, A. K¨ostinger, Ch. Magele, W. Renhart, M. Repetto*, “SMES Optimization Benchmark: TEAM Workshop Problem 22“, International Compumag Society, http://www.compumag.org/jsite/images/stories/TEAM/problem22.pdf, 1996.

[15] *D. Ioan, G. Ciuprina, A. Szigeti*, “Embedded stochastic-deterministic optimization method with accuracy control”, IEEE Transactions on Magnetics, vol. 35, pp. 1702-1705, 1999.

[16] *R. Bărbulescu, A-S. Lup, G. Ciuprina, D. Ioan, A.E. Yilmaz*, “Intelligent Particle Swarm Optimization of Superconducting Magnetic Energy Storage Devices”, 2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE), DOI 10.1109/ISFEE.2014.7050607, 2014.

[17] *L.S. Coelho, P. Alotto*, “Global optimization of electromagnetic devices using an exponential quantum-behaved particle swarm optimizer”, IEEE Transactions on Magenetics 44, pp. 1074–1077, 2008.

[18] *J. McCaffrey*, "Artificial Intelligence - Particle Swarm Optimization", Microsoft magazine, http://msdn.microsoft.com/en-us/magazine/hh335067.aspx, 2011.

[19] *T. Altinoz, A.E. Yilmaz, A. Duca, and G. Ciuprina*, “Particle Swarm Optimization with social exclusion and its application in electromagnetics”, in the International Conference on Optimization of Electrical and Electronic Equipment (OPTIM 2014), Bran, Romania, pp. 105-110, 2014.

[20] *T. Altinoz, A.E. Yilmaz, A. Duca, and G. Ciuprina*, “Incorporating the Avoidance Behavior to the Standard Particle Swarm Optimization 2011”, Advances in electrical and computer engineering, vol. 15 (2), pp. 51-58, 2015.

[21] *G. Ciuprina, D. Ioan, I. Munteanu*, “Use of intelligent-particle swarm optimization in Electromagnetics”, IEEE Transactions on Magnetics, vol. 38 (2), pp. 1037-1040, 2002.

[22] *M. Clerc*, "Standard Particle Swarm Optimization", Open access archive HAL, https://hal.archives-ouvertes.fr/hal-00764996/document, 2012.

[23] *Q.-K. Pan, M.F. Tasgetiren, Y.-C. Liang*, "A discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem with makespan and total flowtime criteria", Computers and Operations Research, pp. 2807-2839, 2008.

[24] *F. Van den Bergh*, “An Analysis of Particle Swarm Optimizers”, Ph.D. Thesis, University of Pretoria, November 2001.

[25] *J. Sun, B. Feng, W.B. Xu*, "Particle swarm optimization with particles having quantum behavior", IEEE Proceedings of Congress on Evolutionary Computation, pp. 325–331, 2004.

[26] *A. Duca, L. Duca, G. Ciuprina, A.E. Yilmaz, T. Altinoz*, “PSO algorithms and GPGPU technique for electromagnetic problems”, International Journal of Applied Electromagnetics and Mechanics, vol. 53, DOI 10.3233/JAE-140166, pag. 249–259, 2017.