

# Applying a New Parallelized Version of PSO Algorithm for Electrical Power Transmission

M Zemzami<sup>1,3</sup>, A Makhoulfi<sup>2</sup>, N Elhami<sup>3</sup>, A Elhami<sup>2</sup>, M Itmi<sup>1</sup> and N Hmina<sup>3</sup>

<sup>1</sup>LITIS, Normandy University, National Institute of Applied Sciences, Rouen, France

<sup>2</sup>LOFIMS, Normandy University, National Institute of Applied Sciences, Rouen, France

<sup>3</sup>LGS, Ibn Tofail University, National School of Applied Sciences, Kenitra, Morocco  
maria.zemzami@gmail.com

**Abstract.** In this paper, the optimization of an electric power transmission material is presented giving specific consideration on material configuration and characteristics. The nature of electric power transmission networks makes it hard to manage. Thus, giving need for optimization. So the problem of optimization of electric power transmission as considered in this paper is improving the performance and reliability of the electricity pylon; the objective is to maximize resistance to load while reducing material usage and cost. For this purpose, we suggest a new version of PSO algorithm that allows the amelioration of its performance by introducing its parallelization associated to the concept of evolutionary neighborhoods. According to the experimental results, the proposed method is effective and outperforms basic PSO in terms of solution quality, accuracy, constraint handling, and time consuming.

## 1. Introduction

Particle swarm optimization (briefed as PSO) is a nature-inspired algorithm that has shown outstanding performance in solving many realistic problems. It is now one of the most commonly used optimization techniques. However, premature convergence and computational costs are still a great constraints for PSO, as well as for all other metaheuristics, especially in complex optimization problems with time consuming objective functions.

In the standard PSO, all particles are directly updated by their offspring no matter whether they are improved. If a particle moves to a better position, it can be replaced by the updated. However if it moves to a worse position, it is still replaced by its offspring. In fact, the most particles fly to worse positions for most cases, therefore the whole swarm will converge to local optima. The exploitation and exploration of the search space represent two conflicting behaviors, which work together for solving the problem, and the right balance between them is integral to the performance of the PSO algorithm. The correlation of these behaviors with our approach is apparent, and this has been the inspiration behind the first part of our algorithm. The second part consists on minimizing computational costs by using parallel computation.

In order to test the proposed PSO algorithm, we designed and applied the developed optimization procedure to electricity pylon example. The application problem studied is weight minimization of bars truss, by finding optimal cross-sectional areas of the truss members. The results obtained solving this optimization problem using the proposed PSO algorithm is then compared with the results from ANSYS first order conventional optimization technique.

The present paper has six sections including the introduction. In the next section, a brief description the basic PSO is given. In section 3 our proposed approach based on PSO algorithm is discussed. In Section 4 a description of the problematic is given. Section 5 describes the numerical results. The chapter finally concludes with Section 6.



## 2. Overview of PSO

### 2.1. Optimization

Optimization plays an important role in various disciplines such as Engineering Design, Manufacturing System, Computer Science, Economics, Artificial Intelligence, Operational Research and related fields. It is the process of trying to find the best, or optimal, solution to an optimization problem within a reasonable amount of time. An optimization problem is basically defined as: finding values of the variables that minimize or maximize the objective function while satisfying the constraints. The Optimization problems are centered on three factors: (1) an objective function which is to be minimized or maximized. (2) A set of unknowns or variables that affect the objective function. (3) A set of constraints that allow the unknowns to take on certain values but exclude others.

In order to solve optimization problems, optimization algorithms are used. The classification of optimization algorithm can be carried out in many ways. A simple way is looking at the nature of the algorithms, and this divides the algorithms into two categories: deterministic algorithm, and stochastic algorithms [1]. Deterministic algorithms are those whose behavior can be completely predicted as they are presented with same set of input and algorithm does same computations and produce same set results every time.

For stochastic algorithms, in general we have two types: heuristic and metaheuristic. Heuristic algorithms find solutions in a reasonable amount of time but there is no guarantee that optimal solutions are reached. On the other hand metaheuristic uses certain tradeoff a randomization and local search, as randomization provides a good way to move away from local search to the search on global scale which means metaheuristic algorithms intend to be suitable for global optimization. Nature-inspired metaheuristic algorithms are becoming powerful in solving modern global optimization problems.

Many modern metaheuristic algorithms were developed based on swarm intelligence in nature like PSO, this algorithm is based on the natural behavior and movement of insects, birds and fish.

### 2.2. Basic PSO

PSO is a population-based stochastic approach for solving continuous and discrete optimization problems [2]. As described by the inventors James Kennedy and Russell Eberhart in 1995 [2], "particle swarm algorithm imitates human (or insects) social behavior. Individuals interact with one another while learning from their own experience, and gradually the population members move into better regions of the problem space".

#### 2.2.1. Formulation

Each particle is a point in the N-dimensional search space. A particle is represented by its current position  $X_i = (x_{i1}, x_{i2}, \dots, x_{ij})$ , and its current velocity  $V_i = (v_{i1}, v_{i2}, \dots, v_{ij})$   $i=1,2,\dots,q$ ,  $j=1,2,\dots,n$ , where  $q$  is the swarm size. PSO tries to find the optimal solution to the problem by moving the particles and evaluating the fitness of the new position. Each particle searches for better positions in the search space by changing its velocity and position with every iteration, updating velocity depends on three factors; the particle's personal best position, the swarm's best position and the particle's previous velocity. The velocity and position of particles are calculated respectively as follows (1) and (2):

$$V(t+1) = V(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t)) \quad (1)$$

$$X(t+1) = X(t) + V(t+1) \quad (2)$$

$D$  is the dimension of the problem, i.e. the number of parameters of the function being optimized,

$C1$  and  $C2$  are acceleration coefficients, i.e. cognitive and social parameters,

$r1$  and  $r2$  are random numbers between 0 and 1,

$X_iD(t)$  is the position of particle  $i$  at time step  $t$ ,

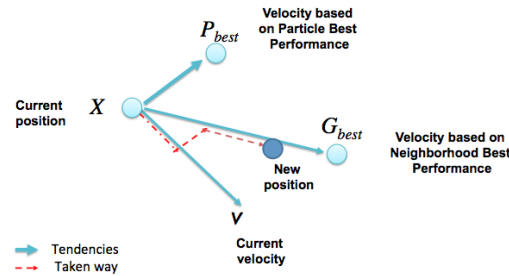
$V_iD(t)$  is the velocity of particle  $i$  at time step  $t$ ,

$PbD(t)$  is the best "remembered" individual particle position, at time step  $t$ ,

$PgD(t)$  is the best "remembered" swarm position, at time step  $t$ ,

PSO has been the focus of many research efforts. Many improvements have been suggested and many more are still possible. Even in its current state, PSO is considered a fast and robust method for optimization of continuous nonlinear functions. In a PSO system, particles change their positions by

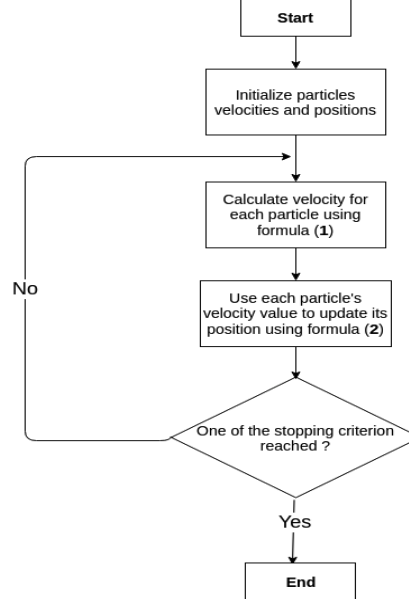
flying around in a multidimensional search space until reaching the convergence criteria. Each particle adjusts its travelling speed dynamically corresponding to the flying experiences of itself and its colleagues [3].



**Figure 1.** The movement of the particle by basic PSO

### 2.2.2. Algorithm

The basic Particle Swarm Optimization algorithm is a stochastic process proposed by [2] and it can be summarized as follows: (1) Initialize a population of particles: positions and velocities. The positions of the particles must be randomly distributed in the updating parameter space. (2) Calculate the velocity for each particle in the swarm using equation 1. (3) Update the position of each particle using equation 2. (4) Repeat steps 2 and 3 until convergence (met the stopping criterion).



**Figure 2.** Basic PSO algorithm flow chart

### 2.2.3. Parameters

PSO has a number of parameters that determine not only its behavior but also its effectiveness. The establishment of PSO parameters affects greatly the PSO performance. Many specialists have already dealt with the improvement of PSO performance. From these studies, much effort has been invested to obtain a better understanding of the convergence properties of PSO. These studies based mostly on a better understanding of the basic PSO control parameters, namely the acceleration coefficients, inertia weight, velocity clamping, and swarm size [4] [5]. From these studies it can be concluded that the PSO is sensitive to control parameter choices, specifically the inertia weight, acceleration coefficients and velocity clamping. The wrong initialization of these parameters may lead to divergent or cyclic behavior.

In the initial version of the PSO, the values for  $C1$ ,  $C2$ , the swarm size, the number of iterations, the dimension of the problem, and the stopping criteria; have to be selected. This selection has an impact on the convergence speed and the ability of the algorithm to find the optimum. A lot of variations that

have been studied on the basic PSO equations to select a combination of values work well in a wide range of problems.

### 2.3. Neighborhood

The neighborhood constitutes the social network structure. It represents a particle with which each particle is able to communicate. There are two main types of neighborhoods:

The social neighborhood: this type of neighborhood represents the social proximity, neighborhoods are no longer the expression of the distance but the expression of the exchange of information, the neighbors are defined at the initialization and are not changed by the following. Once the network of social connections established, there is no need to refresh. It is a static neighborhood.

The geographical neighborhood: this type of neighborhood represents the geographical proximity, it is the most natural concept for the neighborhood particle swarm, neighbors are considered the closest particles. However, at each iteration, the new neighbors must be recalculated from a preset distance in the search space. It is a dynamic neighborhood that should be established and updated in each iteration. This is the kind of neighborhood that was retained in our approach.

The change of velocity of formula (1) is performed using a new term in the equation. It was introduced by [6], his picture appears in Figure 3, [7].

$$V(t+1) = V(t) + C1r1(Pb(t) - X(t)) + C2r2(Pg(t) - X(t)) + C3r3(Pn(t) - X(t))$$

With: Pn: the best position of the neighborhood; C3: the acceleration coefficient, also called social setting; r3: random number drawn from the interval [0,1].

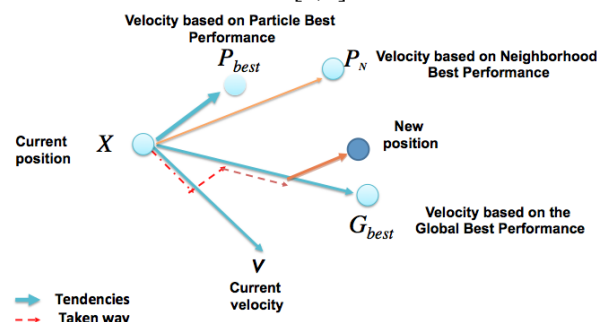


Figure 3. The movement of the particle by PSO Modified

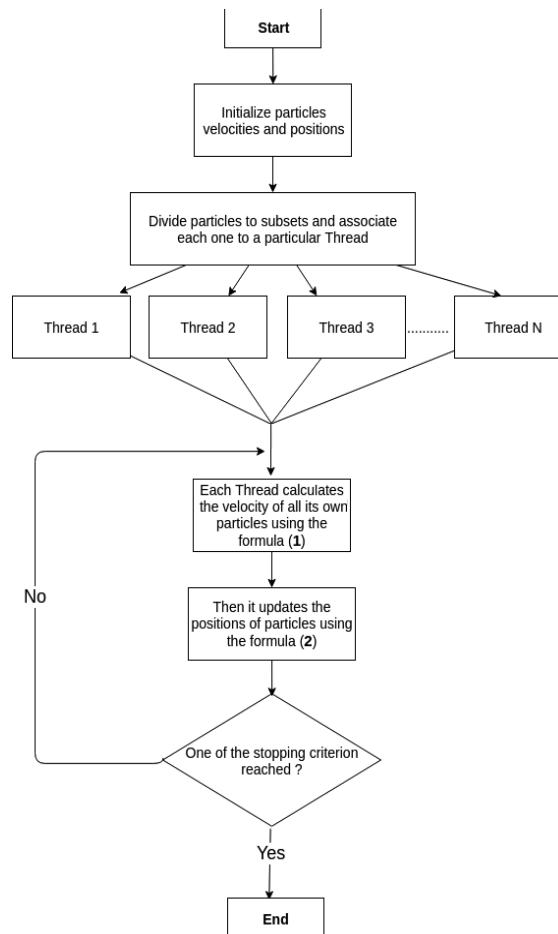
### 3. The proposed approach

Nowadays engineering optimization problems often impose large computational demands, resulting in long solution times even on a modern high-end processor. The performance of serial PSO is affected when complex engineering optimizations are considered; this motivated the development of parallel optimization.

In the implementation of the classical PSO algorithm, all computations are done sequentially, that is where we get the idea of parallelization to improve the performance of the algorithm. Several authors have performed analysis to prove the performance of parallel PSO [8]-[12], the one we have adopted for our implementation, allows parallelizing computations. In fact processes are to perform calculations on set of particles located in different neighborhoods.

The threads, a sort of processes are executed in parallel for each iteration of the algorithm. Each thread executes the processing of an iteration of its particles' set, and waits for the other threads to finish their processing in order to update the neighborhoods and start a new iteration. This scenario repeats itself until a satisfactory solution is obtained: "achievement of stopping criterion". Our neighborhoods have the form of spheres, which are updated at each iteration: their centers evolve and the radius changes according to conditions relating to the number of neighborhoods.

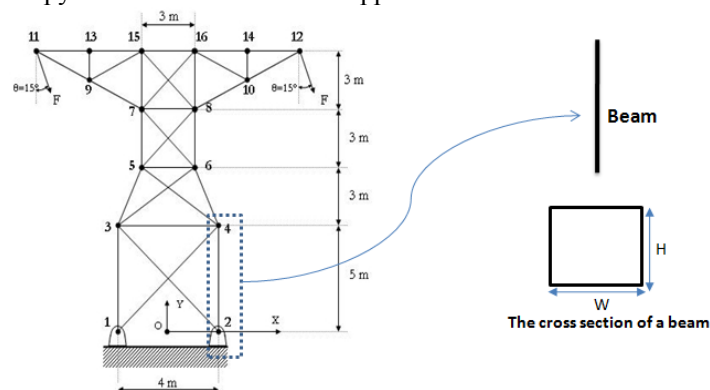
The particularity of the approach consists to take advantage of the robustness of the PSO algorithm in choosing the right parameter setting, particularly the concept of dynamic neighborhood, in order to create diversity in research and in the sharing of information for a more optimal convergence. Besides the parallel computation that accelerates the calculations in order to have an optimal solution in an optimized computation time. The Figure 4 is a flow chart of the suggested approach. The reader is referred to [13] for a more details about this model.



**Figure 4.** Flow chart of the proposed approach

#### 4. Applying our proposed approach for electrical power transmission

In order to test the proposed PSO algorithm, we designed and applied the developed optimization procedure to electricity pylon example. The application problem studied is weight minimization of bars truss, by finding optimal cross-sectional areas of the truss members. The results obtained solving this optimization problem using the proposed PSO algorithm is then compared with the results from ANSYS first order conventional optimization technique. The static scheme of the problem is presented in Figure 5. Node positions and bars connections are fixed, only the sectional areas being the subject of the optimization. A pylon has a force of 18KN applied as shown below.



**Figure 5.** Geometric representation of the electricity pylon

The objective of this problem is to design a bars Steel truss structure of minimum weight (volume) that can with stand applied loads within the limits of allowable displacement. The material properties

are Young's modulus  $E = 200$  GPa, Poisson's ratio  $\nu = 0.3$  and weight density  $\rho$  of  $7500 \text{ kg/m}^3$ . The lengths of the truss members are fixed. The design variables or the input variables for the optimization problem are the cross sectional areas of the truss members. The cross sectional areas of the truss members are allowed to vary between  $5$  and  $100 \text{ mm}^2$ . The displacement limits is  $9 \text{ mm}$  in all direction. The geometry of the problem is shown in Figure 5. The structure is hinged at nodes (1, 2). ANSYS LINK1 element is used to model and simulate the structure for analyses [14]. The truss members are divided into four element groups as shown in Figure 6.

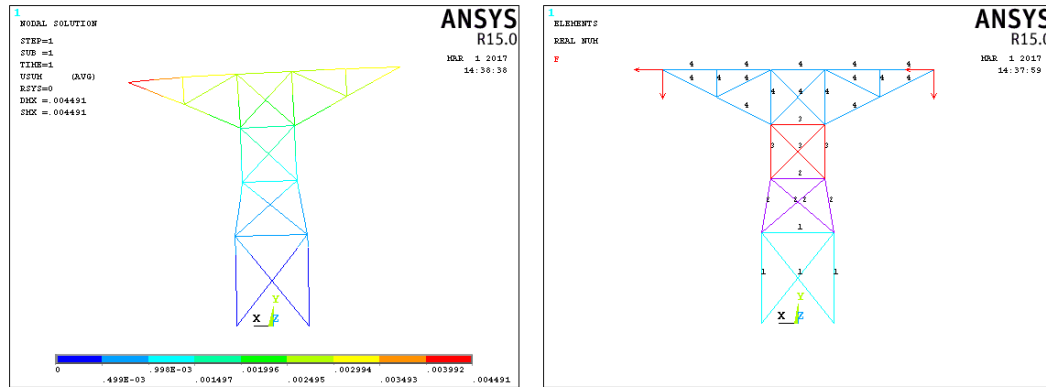


Figure 6. Element groups for the pylon

The four element groups signify four design variables. The design space in optimization can be considered as a four dimensional space where each design variable (cross-sectional area) represents four coordinates of the design space. The mathematical model can be expressed as:

$$A = [A_1, A_2, A_3, A_4]^T$$

$$f = \min W = \sum_{i=1}^4 \rho_i \cdot A_i \cdot L_i$$

Subject to:  $g(A) = u_{\max} - u_c \leq 0$

$$A_{\min} \leq A \leq A_{\max}$$

Where  $A = [A_1, A_2, A_3, A_4]^T$  are design variables for the cross section,  $f$  is the objective function.  $W$  is the total weight of the structure,  $\rho_i$ ,  $A_i$  and  $L_i$  were the density, section area, bar length of the  $i$ th group bars, respectively.  $g(A)$  is the displacement constraints,  $u_{\max}$  et  $u_c$  are the maximum displacement and the displacement limit of the  $i$ th group bars under various conditions.  $A_{\min}$  and  $A_{\max}$  are the minimum and maximum section size, respectively.

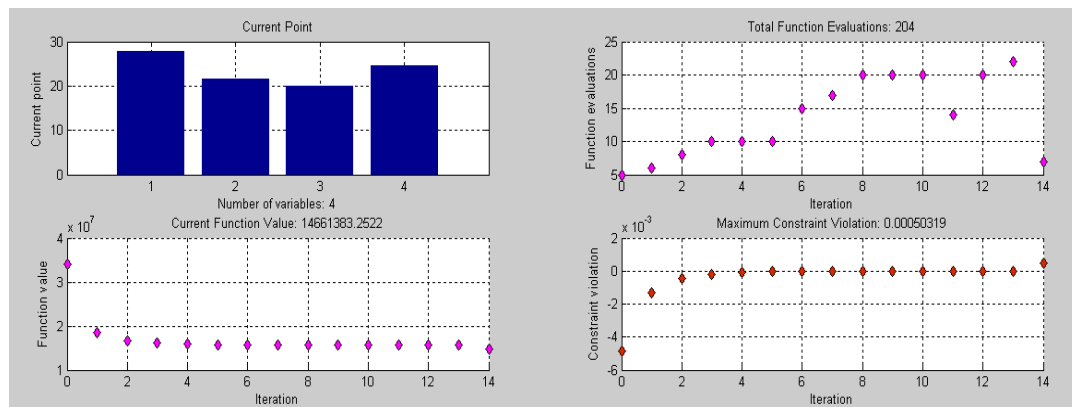
## 5. Numerical Results

The results obtained solving these optimization problems using the proposed PSO algorithm is then compared with the results from ANSYS first order conventional optimization technique.

The ANSYS first order optimization is a conventional optimization method where the true functions (Objective and Constraints) are used for the optimization. An APDL file that defines the pre-processing, solution, post-processing and the optimization phase is used for the conventional ANSYS optimization. In the optimization phase, the algorithm for optimization, the number of iterations, the upper and lower bounds of the design variables and the limits on constraints are specified. The process converges within the specified iteration if a minimum value of objective function has been found that obeys the given constraints. If it does not converge a second iteration has to be setup with a new starting point. This goes on until a convergence is met within the specified constraints.

The same problem was also solved using the ANSYS First order optimization method. The maximum number of iterations was set as 20 for the ANSYS First order optimization process. The optimization problem converged to a minimum value of objective function given by  $W = 1.46 \text{ E7 Kg}$ . The optimal design variables were  $A_1 = 26.86 \text{ mm}^2$ ,  $A_2 = 25.95 \text{ mm}^2$ ,  $A_3 = 24.86 \text{ mm}^2$  and  $A_4 = 24.54 \text{ mm}^2$ . The convergence plot for the ANSYS First order optimization and the optimum values of the design variables are shown in Figure 7.

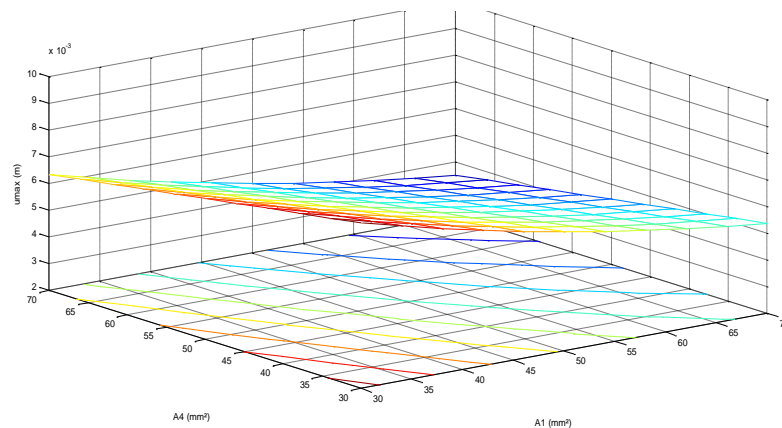




**Figure 7.** Function convergence plot

For the PSO algorithm, the procedure for optimization can be summarized in the following steps.

- Read the ANSYS APDL.
- Create response surface using second order polynomial based regression analysis using the candidate designs and the true responses. The generated response surfaces for displacement constraint are shown in Figure 8.
- Combine the PSO algorithm directly to the response surface and evaluate the optimal design solution.



**Figure 8.** Response plot for maximum displacement

The results obtained for this problem using the Modified PSO Algorithm is given in Table 1.

**Table 1.** Design point from PSO algorithm

Parameter	Initial point	ANSYS First order	MPSO algorithm
A1 (mm)	60	26.85	27.05
A2 (mm)	55	25.94	25.34
A3 (mm)	55	24.86	24.14
A4 (mm)	50	24.54	24.91
Umax (m)	0.0025	0.0085	0.0089
Fobj (Kg)	3.402E7	1.466E7	1.451e+007

It is observed here that the results for the objective function of the MPSO optimization process compare well with those of ANSYS First order optimization process and the optimal solution is obtained in an optimized computation time.

## 6. Conclusion

In this paper, a parallelized version of the Particle Swarm Optimization algorithm is presented. A new concept of evolutionary neighborhoods is associated to the parallel model in order to improve the PSO performance. PSO is a stochastic metaheuristic based on population solutions. It searches for optimal

solutions based on the concepts of cooperation and neighborhoods. Many variations and improvements of classical PSO version have been suggested by adapting its parameters, but good algorithm acceleration is required with a parallelization approach.

In order to test the proposed PSO algorithm, we designed and applied the developed optimization procedure to electricity pylon example. The application problem studied is weight minimization of bars truss, by finding optimal cross-sectional areas of the truss members. The results obtained solving this optimization problem using the proposed PSO algorithm is then compared with the results from ANSYS first order conventional optimization technique.

The PSO algorithm was implemented sequentially, and then in parallel, based on the exchange between the subswarms. By testing the parallel version, and through the results, we find that our model shows remarkable efficiency in terms of reduced time and optimality convergence.

Finally, in the future we intend to study other variants of the suggested parallel model and also to test the program on other various real optimization problems.

## 7. References

- [1] M. Pardalos, E. Romeija, H. Tuyb: Optimization and Nonlinear Equations Recent developments and trends in global optimization. In: Journal of Computational and Applied Mathematics, Numerical Analysis 2000 Vol 124, pp. 209-228, 2000.
- [2] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," In: Proceedings of the IEEE International Joint Conference on Neural Networks, IEEE Press, vol. 8, no. 3, pp. 1943–1948, 1995.
- [3] Y. Cooren, "Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire. Applications en génie médical et en électronique". Doctorat thesis, University of Paris 12 Val de Marne, France, 2008.
- [4] K. E. Parsopoulos and M. N. Vrahatis: "Recent approaches to global optimization problems through particle swarm optimization". In: Natural Computing: an international journal, 1(2-3), pp.235-306, 2002.
- [5] M.E. Hyass and P. Hyass: "Good Parameters for Particle Swarm Optimization". In : Laboratories Technical Report no. HL1001, 2010.
- [6] B. Bochnek et P. Fory's, "Structural optimization for post buckling behavior using particle swarms". Struct Multidisc Optim. Pages 521-531, 2006.
- [7] N. Elhami, R. Ellaia, M. Itmi: "Hybrid Evolutionary Optimization Algorithm MPSO-SA", In: International Journal of Simulation and Multidisciplinary Optimimisation, Vol. 4, pp. 27- 32, 2010.
- [8] P. Rabanal, I. Rodríguez and F. Rubio: "Parallelizing Particle Swarm Optimization in a Functional Programming Environment ». In Algorithms2014 : vol. 7, pp. 554–581, 2014.
- [9] K. Byung-I and G. Alan, "Parallel asynchronous particle swarm optimization," International Journal For Numerical Methods In Engineering, vol. 67, pp. 578-595, 2006.
- [10] J. Chang, S. Chu, J. Roddick and J. Pan : « A Parallel Particle Swarm Optimization Algorithm With Communication Strategies ». In :Journal of Information Science and Engineering, 2005.
- [11] M.Zenzami, N.Elhami, M.Itmi and N.Hmina, "Parallélisation de la Méthode PSO: Découpage de l'espace et traitement par lot des particules". In: International Workshop on New Services and Networks (WNSN'16). Khouribga. Morocco. 2016.
- [12] M.Zenzami, N.Elhami, M.Itmi and N.Hmina, "A New Parallel Approach For The Exploitation Of The Search Space Based On PSO Algorithm". In: International Colloquium in Information Science and Technology (CIST'16). Tangier. Morocco. Indexed in Scopus 2016.
- [13] M.Zenzami, N.Elhami, M.Itmi and N.Hmina, "Parallelization of the PSO algorithm on evolutionary neighborhoods". In: International Conference on Modeling, Optimization and Simulation (MOSIM'16). Montréal. Canada. 2016.
- [14] ANSYS Guide 2015, ANSYS Structural Analysis Guide, 2015.

## Acknowledgments

This research is supported by « XTERM »: Complex Systems, Territories Intelligence and Mobility, co-financed by the European Union with the European regional development fund (ERDF) and Normandy Region.