EngOpt 2008 - International Conference on Engineering Optimization
Rio de Janeiro, Brazil, 01 - 05 June 2008.

Parameter Selection and Convergence Analysis of the PSO Algorithm Applied to the Design of Risers

Aline Aparecida de Pina [1]
Beatriz de Souza Leite Pires de Lima [1,2]
Carl Horst Albrecht [2]
Breno Pinheiro Jacob [1]

[1] Civil Engineering Program, COPPE/UFRJ – Post-Graduate Institute of the Federal University of Rio de Janeiro, Rio de Janeiro, RJ – Brazil
[2] Polytechnic School/UFRJ – Federal University of Rio de Janeiro, Rio de Janeiro, RJ – Brazil
alinepina@gmail.com, bia@coc.ufrj.br, carl@deg.ee.ufrj.br, breno@coc.ufrj.br

Abstract
The design of risers is a very important issue for the offshore exploration of oil and gas as they are essential components of floating production systems (FPS). Many aspects involve the design of those structures mainly security and cost savings. Those demands include the optimum design of the riser configuration.
The current tendency in the design of risers for FPSs considers the use of nonlinear, time-domain dynamic analysis tools, comprising either a coupled analysis strategy (where a hydrodynamic model of the floating system is coupled to a finite element structural and hydrodynamic model of the mooring lines and risers), or else an uncoupled strategy where the finite-element model of an individual line is analyzed, submitted to the motions of the platform. In either case, it is employed a time-domain dynamic simulation that requires a large amount of computational time. Thus, it is necessary the development of computational tools able to improve the efficiency in the design of risers through optimization methods. This work presents a study of the Particle Swarm Optimization method (PSO) applied to the design of risers in lazy wave configuration.
The PSO method has shown good efficiency but its performance greatly depends on its parameters settings. In order to improve the PSO performance, we analyze the behavior of its parameters through several experiments. The results show that some default values initially used were not the best ones to obtain a good performance of the PSO algorithm when applied to the design of risers. The best parameter selection will be included in the *OtimRiser* program – a computational tool that uses several optimization methods based on evolutionary concepts.
Keywords: Offshore Systems, Risers, Particle Swarm Optimization

1. Introduction

The increasing development of petroleum activities in deep and ultra-deep water has encouraged the demands for floating production systems (FPS). Many aspects involve the design of those structures mainly security and cost savings. Those demands include the optimum design of the steel catenary *riser* (SCR) configuration, since, for ultra-deep waters, flexible risers can frequently reach or exceed the technical and economical feasibility limits.
The current tendency in the design of risers for FPSs considers the use of non-linear, time-domain dynamic analysis tools, comprising either a coupled analysis strategy (where a hydrodynamic model of the floating system is coupled to a finite element structural and hydrodynamic model of the mooring lines and risers), or else an uncoupled strategy where the finite-element model of an individual line is analyzed, submitted to the motions of the platform. In either case, it is employed a time-domain dynamic simulation that requires a large amount of computational time. Thus, it is necessary the development of computational tools able to improve the efficiency in the design of risers through optimization methods.
Optimization methods based on evolutionary concepts, as the Particle Swarm Optimization method (PSO), can be used to find an optimum solution to the problem of design of risers. Vieira [1] presents an application of Genetic Algorithms (GAs) and Lima et al. [2] presents an application of a hybrid Fuzzy/GA algorithm applied to the o design of risers.
The Particle Swarm Optimization method (PSO) is a member of the wide category of Swarm Intelligence methods [3], for solving optimization problems. It was originally proposed by James Kennedy and Russel Eberhart as a simulation of social behavior, and it was initially introduced as an optimization method in 1995 [4, 5]. PSO has attracted much attention from computer science and technology communities. It is related with artificial life, specifically to swarming theory, and also with evolutionary computing, especially evolution strategies and genetic algorithms [6].
PSO can solve a variety of difficult optimization problems and has shown a faster convergence rate than other evolutionary algorithms on some problems [7, 8]. Another advantage of PSO is that it has very few parameters to adjust, which makes it particularly easy to implement.
In spite of being recent, the PSO algorithm has been proved to be useful on diverse applications [9 – 26]. On offshore petroleum exploration research field, Albrecht [27] presents an application of the PSO algorithm to the design of mooring systems.
Although its efficiency, the PSO algorithm performance greatly depends on its parameters settings. A bad choice to the parameter values can cause a premature convergence (finding a local, but not global, optimum) or even cause an oscillatory behavior of the particles near to the optimum solution with slow convergence. Trelea [28] presents a theoretical analysis of the PSO algorithm, where it defines some standard ideas to make a good choice in the parameters values. Even though, it is necessary an analysis for each application of the PSO algorithm, because certain values can be good to achieve convergence of one function, but not so good to achieve convergence of other functions.

In this work, we present a convergence analysis of the PSO algorithm applied to the design of risers, testing different parameter settings through several experiments. For the experimental tests, we will use an implementation of the PSO algorithm applied to the design of risers called *OtimRiser* – a computational tool that uses several optimization methods based on evolutionary concepts associated to the design of risers systems.

This paper is organized as follows: In section 2, we present the standard formulation of the PSO algorithm. In section 3, we show some variations of the PSO developed by different researchers to improve the performance of the method. The problem formulation of the design of risers and the implementation details of the PSO in *OtimRiser* are presented in section 4. In section 5, we show the algorithm analysis and our experimental results. In section 6, we present our conclusions.

2. The Particle Swarm Optimization method (PSO)

The Particle Swarm Optimization method (PSO) was developed by Kennedy and Eberhardt, based on observations of the social behavior of animals such as bird flocking and fish schooling [4]. In the PSO method, each individual, called particle, moves through cooperation and competition by successive iterations. The particles learn from their own past experiences and from their neighbors' experiences, by evaluating themselves, comparing their performance with others from the population and imitating only those individuals with more success than themselves. Those movements through the search space are guided by the best evaluations, with the population usually converging on a good problem solution. The quality of those solutions are measured by a predefined fitness function, which is problem-dependent.

Each particle of the swarm is represented by its current position in the search space and its current velocity, which means its change of position. It flies remembering the best search space position it has ever visited and towards the best individual of a topological neighborhood.

The search space is L-Dimensional, and the particle is represented by three L-Dimensional vectors: the current position vector x_i**,** the previous best position p_i and a velocity vector v_i.

To achieve a better performance, Shi and Eberhardt [29] proposed a modification in the standard algorithm introducing coefficients to control the influence of the global best, of the previous best and of the inertia in the particle movement behavior. Therefore, the fly of a particle i is expressed in Eq. (1) as a movement from position x at iteration t with velocity v:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{1}$$

The velocity vector is updated according to the best previously visited position of the i-th particle p_i and the best position found by all particles in the swarm, p_g:

$$v_i(t+1) = \omega * v_i(t) + C_1 * rnd * (p_g - x_i(t)) + C_2 * rnd * (p_i - x_i(t)) \tag{2}$$

where ω is the inertia weight introduced in order to support the balance between exploration and exploitation. The constants, C_1 and C_2, are employed to determine the balance between the influence of the particles set (C_1) and that of the individual's knowledge (C_2). Those coefficients are called social and cognitive parameters, respectively, as they can weigh the social influence (second term on the right-hand side of Eq. (2), or, the cognitive learning (last term of Eq. (2)). The values of *rnd* are random numbers from uniform distributions in the range [0,1]. The terms $(p_g - x_i)$ e $(p_i - x_i)$ are called terms of *acceleration by distance* [4], because they control the velocity variation.

The current position x_i can be considered as a set of coordinates describing a point in the space. In each iteration of the algorithm, the current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, p_i. The value of the best function result so far is stored in a variable that can be called $pbest_i$ (for "previous best"), for comparison on later iterations. The objective, of course, is to keep finding better positions and updating p_i and $pbest_i$. New points are chosen by adding v_i coordinates to x_i, and the algorithm operates by adjusting v_i, which can effectively be seen as a step size [30].

The PSO algorithm is constructed as follows:

1. Initialize a population array of particles with random positions and velocities on L dimensions in the search space.
2. For each particle, evaluate the desired optimization fitness function in L variables.
3. Compare particle's fitness evaluation with its $pbest_i$. If current value is better than $pbest_i$, then set $pbest_i$ equal to the current value, and p_i equal to the current location x_i in L-dimensional space.
4. Identify the particle in the population with the best success so far, and assign its index to the variable g (updating p_g).
5. Update the velocity and the position of each particle according to Eq. (2) and Eq. (1).
6. Repeat steps 2 to 5 until the stopping criteria is met (usually a sufficiently good fitness or a maximum number of iterations)

To reduce the possibility of particles flying out of the search space, Kennedy and Eberhardt [5] put forward a clamping scheme that limited the speed of each particle to a range $[-v_{max}, +v_{max}]$ with v_{max} calculated as shown in Eq. (3).

$$v_{max} = x_{max} - x_{min} \tag{3}$$

where x_{max} and x_{min} are the lower and upper bounds of the search space, respectively.

3. PSO Variants

Many tweaks and adjustments have been made to the standard PSO algorithm over the past decade. Some have resulted in improved general performance, and some have improved performance on particular kinds of problems. We present here some of these variants.

3.1. Passive Congregation and Social Attraction

The PSO algorithm is inspired by social behaviors such as spatial order, more specially, aggregation such as bird flocking, fish schooling, or swarming of insects. Each of these cases has stable spatio-temporal integrities of the group of organisms: the group moves persistently as a whole without losing the shape and density [31].

For each of these groups, different biological forces are essential for preserving the group's integrity: the aggregation and the congregation forces. These forces can be active or passive.

The aggregation force refers to a grouping of the organisms by non-social, external, physical forces. Active aggregation is a grouping by attractive resource, such as food, with each member of the group recruited to a specific location actively. The passive aggregation is a passive grouping by physical processes.

Different from aggregation, congregation is a grouping by social forces, that is, the attractive resource is the group itself. Passive congregation is an attraction of an individual to other group members but where there is no display of social behavior. Active congregation, also known as social congregation, usually happens in a group where the members are related (sometimes highly related).

We can correlate with active aggregation the terms $C_1 * rnd * (p_g - x_i(t))$ and $C_2 * rnd * (p_i - x_i(t))$ in Eq. (2), if we consider p_g and p_i as attractive resources for the group members.

He et al. [33] proposed the introduction of a term in PSO related to the passive congregation force, in other words, a term that represents the attraction of an individual to other group members but where there is no display of social behavior. This involves a particle selected randomly from the swarm and a passive congregation coefficient. Albrecht [27] proposed a similar approach for the influence of the group on the individual denominated *Social Attraction*, where the passive congregation term involves the center of mass of the swarm, not a particle selected randomly from the swarm.

The new term added is defined in Eq. (4).

$$C_3 * rnd * (C_m - x_i(t)) \tag{4}$$

where C_3 is the passive congregation coefficient and the particle C_m is the center of mass of the swarm, where the mass of the particle is represented by the fitness value of the particle, as following (Eq. (5)).

$$C_m = \frac{\sum_{j=1}^{N} f_j x_j}{\sum_{j=1}^{N} f_j} \tag{5}$$

Adding this new term in Eq.(2), the PSO algorithm is now given by Eq.(6).

$$\begin{aligned} v_i(t+1) &= \omega * v_i(t) + C_1 * rnd * (p_g - x_i(t)) + C_2 * rnd * (p_i - x_i(t)) + C_3 * rnd * (C_m - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \tag{6}$$

3.2. Linear Variation of the Inertia Coefficient

The PSO algorithm presents some difficult to converge to the global optimum in the last iterations. It is very efficient at the beginning of the search but it tends to waste time (or some iterations) when close to the optimum. The inertia weight plays an important task in this convergence behavior of the algorithm. In [29] several experiments were driven to assess the influence of the inertia weight. They concluded that for smaller values of inertia weight, the PSO behaves like a local search algorithm and for larger inertia weights the PSO presents a good global exploration always trying to explore new areas.

Eberhardt and Shi [32] proposed that the inertia coefficient ω should be linearly variant, instead of being constant. Therefore, the inertia coefficient is given by Eq. (7).

$$\omega(t) = \left(\omega_{ini} - \omega_{fin}\right) \frac{(N - t)}{N} + \omega_{fin} \tag{7}$$

where N is the maximum number of iterations. ω_{ini} and ω_{fin} are the initial and the final inertia weights, respectively.

3.3. Non-Linear Variation of the Inertia Coefficient

To improve the PSO performance in the last iterations, Chatterjee e Siarry [33] proposed a non-linearly variation of the inertia coefficient, given by Eq. (8).

$$\omega(t) = \left\{ \frac{(N-t)^n}{N^n} \right\} \left(\omega_{ini} - \omega_{fin}\right) + \omega_{fin} \tag{8}$$

where n is the non-linearity exponent.

With this formulation, the inertia influence can be controlled. The initial and final values of the inertia coefficient are pre-determined, but its behavior during the iterations depends on the value of n.

Albrecht [27] proposed a modification in Eq.(8), as following.

$$\omega(t) = K - \frac{(t-1)^n}{N^n} \tag{9}$$

where n and K are the elements of control of the non-linear variation.

Using Eq. (9), the inertia weight ω will be K at the beginning of the process, and K–1, at the end.

3.4. Linear Variation of the Aggregation and Congregation Coefficients

Following the idea the linearly variant inertia weight, Ratnaweera and Halgamuge [34] proposed the linear variation of the aggregation coefficients (C_1 and C_2). The equations for these variations are given in Eq. (10) and Eq. (11).

$$C_1(t) = \left(C_{1fin} - C_{1ini}\right) \frac{t}{N} + C_{1ini} \tag{10}$$

$$C_2(t) = \left(C_{2fin} - C_{2ini}\right) \frac{t}{N} + C_{2ini} \tag{11}$$

where C_{1ini}, C_{1fin} are the initial and final values of C_1 and C_{2ini}, C_{2fin} are the initial and final values of C_2, respectively.

These modifications in the calculation of the coefficients improved the algorithm performance, as shown in [34, 35].

In the *OtimRiser* implementation of the PSO algorithm, the congregation coefficient (C_3) is also linearly variant. The equation for the variation is given in Eq. (12).

$$C_3(t) = \left(C_{3fin} - C_{3ini}\right)\frac{t}{N} + C_{3ini} \tag{12}$$

where C_{3ini}, C_{3fin} are the initial and final values of C_3, respectively.

4. The PSO Algorithm Applied to the Design of *Risers*

The PSO algorithm with the variations mentioned above was implemented in the *OtimRiser* program. The *OtimRiser* is a computational tool that uses several optimization methods based on evolutionary concepts and is still being developed by a group of researchers and students of the Civil Engineering Program/COPPE/UFRJ.

A detailed explanation about the implementation of some evolutionary algorithms (Genetic Algorithm, Micro-Genetic Algorithm, PSO and Evolutive Strategy) applied to the design of mooring systems can be seen in [27]. In this work, we focus only on the results of the Particle Swarm Optimization method applied to the design of *risers*.

4.1. The Problem of Designing *Risers*

Petroleum industries around the world have been faced with the permanent challenge of developing oil production activities in deep and ultra-deep waters. One particular aspect of these activities regards the optimum design of the steel catenary *riser* (SCR) configuration, since, for ultra-deep waters, flexible risers can frequently reach or exceed the technical and economical feasibility limits.

The authors have previously studied the use of SCRs installed in FPSs [36, 37]. In these studies several different alternative configurations of SCRs were considered, including the free-hanging catenary and other configurations with flexing and/or floating elements such as the lazy wave, steep-wave, lazy-S and steep-S configurations (Fig. 1). The behavior of these configurations was simulated by several non-linear time-domain dynamic analyses. These studies have shown, as expected, that configurations with floating elements such as the lazy wave configuration present a structural behavior more favorable than the free-hanging catenary, both to extreme conditions and to operation/fatigue conditions. Although the free-hanging catenary presents smaller costs and the feasibility may be proven in some particular situations, the lazy wave configuration was selected as a base case, and more detailed studies were performed in [38].

The Lazy wave configuration (Fig. 1f) may be defined as a double catenary arrangement, with an intermediate segment comprised of distributed buoys. This segment alleviates the weight supported by the floating unit, and contributes with restoring moments when submitted to lateral loads.

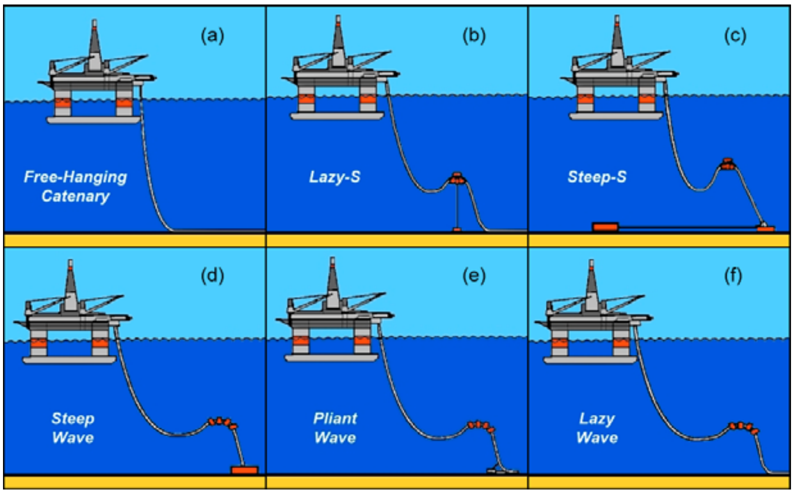

Figure 1. Some catenary configurations assumed by the *risers*.

The exhaustive parametric studies of Jacob *et al.* [36 – 38] comprised a hard task, involving the ‘manual’ variation of different geometric parameters and environmental data. Thus, in order to find a configuration with minimum effort, this work presents a study of the Particle Swarm Optimization method (PSO) applied to the design of risers in lazy wave configuration.

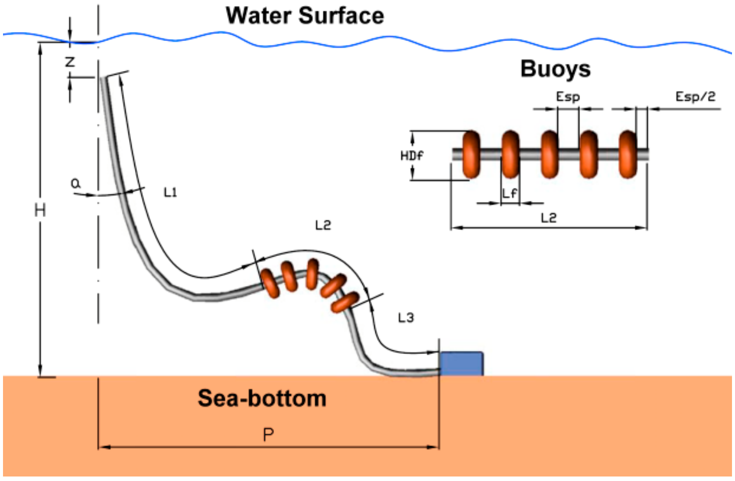

Figure 2. The project variables.

The project variables of the optimization process are the global geometric values presented in Figure 2:

- L1 - Length of top segment of the riser
- L2 - Length of segment with distributed buoys
- L3 - Length of lower riser segment
- α - Top angle (angle between the top riser axis and the vertical direction)
- H - Still water level
- z - Connection depth
- P - Horizontal projection
- Buoys characteristics:
    - HDf - External diameter
    - Lf - Length
    - Esp - Spacing between buoys
- Other variables:
    - Rhof - Weight of the buoys
    - Mechanic and Geometric section properties

To simplify the optimization problem, the main project variables were chosen as search variables: L1, L2, L3, HDf, Lf and Esp. The values of the other project variables are calculated from these search parameters or are given by fixed input values (as defined in [1]). In *OtimRiser* it is possible to select and control the PSO parameters, as well as defining the upper and lower bounds to each search variables of the problem. Table 1 presents the length limit values that describe the search space used in our tests.

Table 1. Limits of the search variables.

Variable	**Min. Value**	**Max. Value**
L1	800	2000
L2	400	800
L3	800	2000
HDf	0.5	2
Lf	0.5	2
Esp	0.8	1.5

4.2. Objective Function and Project Constraints

In engineering optimization problems, the objective function usually involves the lowest construction cost. In this work, the length and the cost of each segment as well as the cost of the buoys are taken as the cost function.

Therefore, the objective function of the problem is given by Eq. (13).

$$f = \frac{\left(\sum_{i=1}^{n} IC_i \times L_i\right) + \left(V_{flut} \times IC_{flut}\right)}{f_{\max}} \qquad (13)$$

The value of $f_{\max}$ is given by Eq. (14).

$$f_{\max} = \left(\sum_{i=1}^{n} IC_i \times ParMax_i\right) + \left(VMax_{flut} \times IC_{flut}\right) \qquad (14)$$

where:

L_i – length of segment i

IC_i – cost weight of segment i

V_{flut} – buoys volume

IC_{flut} – cost weight of buoys

$ParMax_i$ – maximal value of parameter i

$VMax_{flut}$ – maximal volume of buoys

The constraints of the present optimization problem involve the *risers* structural behavior and are presented as following:

- Von Mises stress acting on riser sections
- Top angle - angle between the riser axis and the vertical direction at the connection with the platform (dictated by installation requirements):
  - Upper bound (set to 5°)
  - Lower bound (set to 18°)
- "Built-in" angle variation - measured at the top riser axis, between the neutral equilibrium configuration and any configuration (set to 5°)
- Maximal tension at the top (set to 1500 kN)
- Minimal tension at the riser bottom (set to 300 kN)

Therefore, the penalty function is defined as follows:

$$P = \begin{cases} k \times (1 - x^3), & \text{if } x < 1 \\ 0, & \text{if } x \geq 1 \end{cases} \quad (15)$$

where:

$$x = \frac{Limit\ Value}{Calculated\ Value} \quad (16)$$

and k attends to force the arising of non restricted solutions.
Then, the objective function with penalties is given by Eq. (17).

$$fitness = \frac{1}{f + \sum P_j} \quad (17)$$

where P_j is the penalty related to violations of the j-ith constraint criteria.

Therefore, the maximal *fitness* value occurs when f is minimal and there are no violations in the behavior constraints.

5. The *OtimRiser* Analysis

An optimization algorithm must be robust, to deal with functions that describe problems, and accurate, to find an optimum solution. In the case of *risers* optimization, which is a process of high computational cost, the algorithm also needs to be efficient, in other words, it needs to achieve a good result with the lesser number of evaluations of the objective function. The parameters of the PSO algorithm in Eq. (6) that will be evaluated are the inertia coefficient (ω) and the coefficients C_1, C_2 and C_3. The behavior of the variation of the coefficients C_1, C_2 and C_3 will be set as increasing or decreasing, depending on their initial and final values in the Equations (10), (11) e (12). Therefore, we can have 8 different types of configuration for C_1, C_2 and C_3.

We will analyze the behavior of the algorithm according to the type of variation applied to the inertia coefficient (ω) and to the configuration of C_1, C_2 and C_3. The initial and final values used for each coefficient in each configuration can be seen in Table 2. The configuration 1 is the one used initially in *OtimRiser*. We will verify if this configuration is really the one that returns the best results in PSO.

Table 2. Initial and final values of the coefficients C_1, C_2 and C_3 in each configuration.

	C_1		C_2		C_3	
Config	**Initial**	**Final**	**Initial**	**Final**	**Initial**	**Final**
1	1	2	1	2	1	2
2	2	1	1	2	2	1
3	1	2	2	1	2	1
4	1	2	2	1	1	2
5	2	1	1	2	1	2
6	1	2	1	2	2	1
7	2	1	2	1	2	1
8	2	1	2	1	1	2

To decrease the number of tests necessary to analyze the algorithm, two types of analysis for each varied parameter had been made. The first type was the execution with the same random seed, that is, the initial random population is always the same one. In the second type of analysis, 30 simulations had been made and the average value of the analyses was used as comparison.

The invariant data in the tests were:

- Size of the population: 10 individuals.
- Maximum number of generations: 100.
- Stopping criterion: maximum number of generations.

The parameters chosen for comparison were:

- Fitness – Maximum value of the objective function reached by the method;
- Number of Evaluations – The total number of evaluations of the objective function;
- Gain – Percentile relation between the best value of the objective function in the first generation (randomly generated) and the best value in the last generation. The gain indicates how much the method was capable to improve the initial population;
- Efficiency – Relation between the gain and the number of evaluations.

An input variable k plays different roles in function of the different types of inertia coefficient variation. In the equations referring to each type of variation of inertia, the variable k exerts the following functions:

- Fixed inertia: $k = \omega$;
- Inertia with linear variation: $k = \omega_{ini}$ e $\omega_{fin} = 0$;
- Inertia with nonlinear variation: $k = n$ and K = 1.

5.1. Experimental Results for the Inertia Weight Variation with Fixed Random Seed

We test the algorithm with all 8 configurations of the coefficients C_1, C_2 and C_3 as defined previously. For each configuration, we made k = 0.8 (fixed). The *fitness* value of the seed is 0.694 where L1 = 1196.85, L2 = 634.92, L3 = 1026.77, HDf = 1.80, Lf = 0.60 and Esp = 1.00.

Table 2 presents the results of the experimental tests of each configuration of the coefficients C_1, C_2 and C_3, using three different types of inertia weight variation (fixed, linear and nonlinear).

Table 3. Results of the experimental tests with fixed random seed

Config	Inertia Weight					
	Fixed		Linear Variation		Nonlinear Variation	
	Fitness	N. Eval	Fitness	N. Eval	Fitness	N. Eval
1	1.894	916	1.745	753	1.851	929
2	1.921	956	1.928	901	1.921	846
3	1.944	956	1.884	792	1.911	929
4	1.923	935	1.727	886	1.932	894
5	1.948	931	1.903	859	1.928	916
6	1.928	808	1.753	708	1.784	860
7	1.856	930	1.859	685	1.859	603
8	1.896	976	1.743	778	1.934	765

We can see that the method with fixed inertia weight presented the best *fitness* results, followed by the method with nonlinear variation of the inertia. However, the method with nonlinear variation of the inertia presented a greater efficiency. To obtain more meaningful results, we need to make statistics tests, running the PSO algorithm many times to each configuration.

5.2. Statistical Experimental Results for the Inertia Weight Variation

In the tests of variation of the inertia coefficient, we will use only configurations 1 (for being the originally implemented in the program), 2, 3, 4 and 5.

In the experiments carried out in this work, we set the final value of the inertia equal to zero. The input value k varies according to the following Eq. (18).

$$k = 0.4 \times j, \quad \text{were} \quad j = 1, \ldots, 6 \tag{18}$$

The algorithm was executed 30 times for each configuration with each value of k. The *fitness* average values for each tested configuration and the values of k that had generated such results can be seen in Table 4.

Table 4. Best input values of k for each type of inertia weight variation.

Config	Inertia Weight					
	Fixed		Linear Variation		Nonlinear Variation	
	k	Fitness	k	Fitness	k	Fitness
1	0.8	1.86030	1.2	1.86030	0.8	1.86013
2	0.8	1.83500	0.8	1.82250	1.2	1.84573
3	0.8	1.85427	0.8	1.87043	1.2	1.86550
4	0.8	1.83907	1.6	1.84713	2.4	1.86450
5	1.2	1.87640	1.2	1.87843	2.0	1.88417

From Table 3 we can see that configuration 5 got the best results of *fitness* average for any type inertia weight variation. Therefore, we will start to use the configuration 5 in the implementation of the *OtimRiser* and the best values of *k* for each type of inertia weight variation can be used as default values of the program.

Although the maximum value of the objective function is not known, the highest *fitness* value found in all the tests carried through was 1.953.

6. Conclusion

The objective of this work was to find values for the parameters of the PSO algorithm, so that, applied to the project of *risers*, we got a good result with the smallest number of evaluations of the objective function. Our experiments had shown that the configuration initially implemented in the *OtimRiser* did not get the best results. The *OtimRiser* program is currently under construction and being updated, so the results of the analysis presented here will be used as default input values for the PSO parameters. With the alteration of those values, we expect to get better results in a static or even dynamic analysis of the problem.

Acknowledgements

The authors would like to express their gratitude to CAPES for the financial support provided during the course of this present research.

References

1. L.T. Vieira. Projeto Ótimo de *Risers* via Algoritmos Genéticos. D.Sc. Qualify, COPPE/UFRJ, Rio de Janeiro, 2004.
2. B.S.L.P. Lima, B.P. Jacob and N.F.F. Ebecken. A hybrid fuzzy/genetic algorithm for the design of offshore oil production *risers*. International Journal For Numerical Methods In Engineering. Int. J. Numer. Meth. Engng 2005; 64:1459–1482.
3. J. Kennedy, R.C. Eberhardt and Y. Shi. Swarm Intelligence. The Morgan Kaufmann Series in Evolutionary Computation, 2001, Academic Press, San Francisco.
4. J. Kennedy and R. Eberhardt. Particle Swarm Optimization. Proc. IEEE Conference on Neural Networks, 1995, p. 1942–1948.
5. J. Kennedy and R. Eberhardt. A New Optimizer Using Particle Swarm Theory. Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, p. 39–43.
6. K.E. Parsopoulos and M.N. Vrahatis, Recent approaches to global optimization problems through Particle Swarm Optimization, Natural Computing 1: 235–306, 2002.
7. X. Hu, R.C. Eberhardt and Y. Shi. Engineering Optimization with Particle Swarm. In IEEE Swarm Intelligence Symposium, 2003, Indianapolis, p. 53–57.
8. R. Hassan, B. Cohanim, O. de Weck and G. Venter. A Comparison of Particle Swarm Optimization and the Genetic Algorithm. In 1st AIAA Multidisciplinary Design Optimization Specialist Conference, 2005, Austin, TX, No. AIAA-2005-1897.
9. A.R. Cockshott, B.E. Hartman. Improving the fermentation médium for Echinocandin B production part II: Particle swarm optimization. Process Biochemistry, 2001, Vol. 36, p. 661–669.
10. C.O. Ourique, E.C. Biscaia Jr. and J.C. Pinto. The Use of Particle Swarm Optimization for Dynamical Analysis in Chemical Processes. Computers and Chemical Engineering, 2002, Vol. 26, p. 1783–1793.
11. E.F. Costa Jr., P.L.C. Lage and E.C. Biscaia Jr. On the numerical solution and optimization of styrene polimerization in tubular reactors. Computers and Chemical Engineering, 2003, Vol. 27, p. 2591–1604.
12. S. Kannan, S.M.R. Slochanal, P. Subbaraj and P.P. Narayana. Aplication of Particle Swarm Optimization Technique and its Variants to Generation Expansion Planning Problem. Electric Power Systems Research, 2004, Vol. 70, p. 203–210, Elsevier.
13. M. Wetter and J. Wright. A comparison of Deterministic and Probabilistic Optimization Algorithms for Nonsmooth Simulation-Based Optimization. Building and Evironment, 2004, p. 989–999, Elsevier.
14. K.E. Parsopoulos and M.N. Vrahatis. On the Computation of All Global Minimizers Through Particle Swarm Optimization. Transaction on Evolutionary Computation, IEEE, 2004, Vol. 8, No. 3.
15. R.C. Eberhardt and Y. Shi. Guest Editorial: Special Number on Particle Swarm Optimization. Transaction on Evolutionary Computation, IEEE, 2004, Vol. 8, No. 3.
16. J.A.C.C. Medeiros. Enxame de Partículas como Ferramenta de Otimização em Problemas Complexos de Engenharia Nuclear. D.Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, 2005.
17. N.N. Siqueira. Otimização Aplicada ao Planejamento de Políticas de Testes em Sistemas Nucleares por Enxame de Partículas. M.Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, 2006.
18. R.E. Perez and K. Behdinan. Particle Swarm Aproach for Structural Design Optimization. Computers & Structures, 2007, 85. Available online at www.sciencedirect.com.
19. C. Coello and E. Luna. Use of Particle Swarm Optimization to Design Combinatorial Logic Circuits. In: A. Tyrell, P. Haddow and J. Torrensen editors. 5th International Conference on Envolvable Systems: from Biology to Hardware, ICES 2003. Lecture Notes in Computer Science, Vol. 2606. Trondheim, Norway: Springer; 2003. p. 398-409.
20. Y. Zheng, L. Ma and J. Qian. Robust Pid Controller Design Using Particle Swarm Optimizer. In: IEEE International Symposium on Intelligence Control, 2003. p. 974–9.
21. R. Krohling, L. Coelho and Y. Shi. Cooperative Particle Swarm Optimization for Robust Control System Design. In: 7th Online World Conference on Soft Computing in Industrial Applications, 2002.
22. P. Oliveira, J. Cunha and J. Coelho. Design of Pid Controllers Using the Particle Swarm Algorithm. In: Twenty-first IASTED International Conference: Modelling, Identification and Control (MIC 2002), Innsbruck, Austria, 2002.
23. W. Zhang, M. Liu and Y. Clerc. An Adaptive PSO Algorithm for Reactive Power Optimization. In: Sixth International Conference on Advances in Power System Control, Operation and Management (APSCOM), Hong Kong, China, 2003. p. 302–7.
24. M. Abido. Optimal Design of Power System Stabilizers Using Particle Swarm Optimization. IEEE Trans Energy Convers 2002;17(3):406–13.

25. J. Peng, Y. Chen and R. Eberhart. Battery Pack State of Charge Estimator Design Using Computational Intelligence Approaches. In: Fifteenth Annual Battery Conference on Applications and Advances, 2000. p. 173–7.
26. S. He, E. Prempain and Q. H. Wu. An Improved Particle Swarm Optimizer For Mechanical Design Optimization Problems. Engineering Optimization, Vol. 36, No. 5, p. 585–605, October 2004.
27. C.H. Albrecht. Algoritmos Evolutivos Aplicados à Síntese e Otimização de Sistemas de Ancoragem. D.Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, 2005.
28. I.C. Trelea. The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection. Information Processing Letters, 2003, No. 85, p. 317–325.
29. Y. Shi and R.C. Eberhardt. A Modified Particle Swarm Optimizer. IEEE International Conference on Evolutionary Computation, 1998, Anchorage, p. 1945–1950, Alaska.
30. R. Poli, J. Kennedy and T. Blackwell. Particle Swarm Optimization: An Overview, Swarm Intell (2007) 1: 33–57, DOI 10.1007/s11721-007-0002-0
31. S. He, Q.H. Wu, J.Y. Wen, J.R. Saunders and R.C. Paton. A Particle Swarm Optimizer with Passive Congregation. Elsevier BioSystems, 2004, Vol. 78, p. 135 –147.
32. R.C. Eberhardt and Y. Shi. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. IEEE International Conference on Evolutionary Computation, 2000, San Diego, California, p. 84–88.
33. A. Chatterjee and P. Siarry. Nonlinear Inertia Weight Variation for Dynamic adaptation in Particle Swarm Optimization. Computers & Operations Research, Elsevier, 2004, Article in Press.
34. A. Ratnaweera and S.K. Halgamuge. Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients. Transactions On Evolutionary Computation, IEEE, 2004, Vol. 8, No. 3, June .
35. R.C. Eberhardt and Y. Shi. Empirical Study of Particle Swarm Optimization. IEEE proc. Conference on Evolutionary Computation, 1999, Washington DC, p. 1945 – 1949.
36. B.P. Jacob, B.S.L.P. Lima and M.C.T. Reyes. Studies on Alternative Configurations for Steel Catenary *Risers* Connected to FPSO’s (in Portuguese). LAMCSO/COPPE/Federal University of Rio de Janeiro-Petrobras Report ET-150822, October 1998.
37. B.P. Jacob, B.S.L.P. Lima, M.C.T. Reyes, A.L.F.L. Torres, M.M. Mourelle and R.M.C. Silva. Alternative Configurations for Steel Catenary *Risers* for Turret-Moored FPSO’s. Proceedings of the 9th International Offshore and Polar Engineering Conference, Brest-France, vol. II, 1999; 234–239.
38. B.P. Jacob, B.S.L.P. Lima and M.C.T. Reyes. Studies on the Lazy-Wave Configuration for Steel Catenary *Risers* Connected to FPSO’s (in Portuguese). LAMCSO/COPPE/Federal University of Rio de Janeiro-Petrobras Report PEC-361, January 2001.