A COMPARATIVE STUDY APPLIED TO RISERS OPTIMIZATION USING BIO-INSPIRED ALGORITHMS

Ian Nascimento Vieira¹, Angelo José Moreira Silva¹, Beatriz S.L.P de Lima¹², Breno Jacob¹

ianvieira@yahoo.com.br, angelo.gemini@gmail.com, bia@coc.ufrj.br, breno@coc.ufrj.br
1COPPE/ Federal University of Rio de Janeiro
P.O. Box 68506, 21941-972 – Rio de Janeiro – RJ – Brazil
Carl Horst Albrecht², carl@ deg.ee.ufrj.br
2Polytechnic School, Federal University of Rio de Janeiro

Abstract. This work presents a comparative study of three different bio-inspired optimization methodologies applied to the optimization of a Steel Catenary Riser (SCR) for floating oil production systems. This problem arises from oil production activities that reach deep and ultra-deep waters. The optimization of such a system requires a time-consuming objective function evaluation, for this reason, we adopted a simplified evaluation method that uses a catenary analytic formulation. Finally the effectiveness of the employed algorithms, Artificial Immune System (AIS), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), is measured by the number of the objective function calculations and the respective values achieved. Results indicate that AIS approach is more effective than GA and PSO, generating better solutions with a small number of evaluations.

Keywords: Artificial immune System, Genetic Algorithm, Particle Swarm Optimization, Steel Catenary Risers

1 - INTRODUCTION

The increasing development of petroleum activities in deep and ultra-deep water has encouraged the demands for floating production systems. Many aspects involve the design of those structures mainly security and cost savings. The design of production risers, to connect the wellheads at the seabottom with a floating platform at the sea surface, is one crucial aspect of these activities.

Floating production systems can use flexible or rigid risers for transporting the oil from the bottom of the sea. Usually it employs flexible risers, that are usually very expensive and present technical and economical feasibility limits to water depths beyond 1000 m. This fact is due to the restriction of the viable riser diameter, particularly when associated to high external pressures and temperatures, and to significant static offsets and heave motions, associated with reduced capability of sustaining harsh service conditions (Lima et al., 2005).

Lately, the SCR (Steel Catenary Riser) concept has been shown to be able to overcome such limitations. Previous studies (Jacob et al. 1999) demonstrated that the lazy-wave configuration, in which distributed floaters are installed in an intermediate section of the riser (Fig. 1), presents a favorable structural behavior which is better than the usual free-hanging catenary, under environmental loadings of wave, marine current, and the motions imposed by the platform.

Figure 1 – Lazy-wave Configuration

The main motivation of this work, is the choice of a riser configuration with a good structural performance and low cost described as an optimization problem. This fact had already been recognized in Lima et al. (2005), where GA were employed in the development of an optimization procedure for lazy-wave risers.

However, the evaluation of the riser configurations, needed for the calculation of the objective function in the optimization algorithms, requires structural analyses employing a extensively time consuming computations. Therefore, an ideal optimization approach should be able to find an optimum solution in the shortest time possible, indicating that efforts should be directed to minimize the number of analyses for the calculation of the fitness of each candidate configuration. So, the main objective of this work is to compare nature-inspired methodologies searching the minimum number of objective function evaluations. The employed optimization algorithms are: Artificial Immune System – AIS, Genetic Algorithms – GA
and Particle Swarm Optimization – PSO.

This paper is organized as follows. Section 2 describes the optimization methodologies applied in this work. Section 3 presents the formulation of the riser optimization problem. Section 4 describes the comparative study among PSO, GA and AIS aiming to find the algorithm that achieve the best fitness with the lower possible number of evaluations. Finally, Section 5 presents the final remarks and conclusions.

2 - OPTIMIZATION METHODOLOGY

2.1 - Engineering Optimization Problems

Classical optimization approaches involves gradient information that can be difficult to apply to most of the real-world engineering optimization problems. Certainly, the evolutionary algorithms like AIS, GA and PSO have the distinct advantage of being able to solve multi-objective and multi-constraint problems that other gradient type of optimizers have failed to meet.

The computational cost of the objective function evaluation can be a serious dilemma to evolutionary algorithms applied to structural optimization problems. Some works present techniques that aim to reduce the computational effort of such algorithms.

All this nature-inspired algorithms have been applied to solve many complex engineering problems. Previous works presented applications in structural optimization. Gero et al. (2006) elaborated an elitist GA for design optimization of 3D steel structures and compared it to classical optimization techniques. The results showed that GA can present a higher computational cost compared to the classical methods; however it was able to find better solutions to the problem. A parameterless adaptive penalty scheme for genetic algorithms applied to constrained optimization problems was presented by Lemonge et al. (2004). The penalty parameters are automatically adjusted along the evolutionary process. The procedure is shown to be effective and robust when applied to structural engineering problems.

The first paper employing the immune system to solve structural optimization problems reported in the literature is the work of Yoo and Hajela (1999). Luh et al. (2004) proposed a multi-modal immune algorithm (MMIA) for finding optimal solution to multi-modal structure topology problems. The immune algorithm is used for improving the local search ability compared to GA. The methodology was tested within two benchmarks examples in structural topology optimization problems showing that the MMIA is an effective approach.

In Pina et al. (2008) a parametric study of PSO algorithm was done applied to riser optimization. PSO showed good performance but presented great dependence on the parameters adjustment. Albretch (2005) presented an application of PSO and GA to mooring lines synthesis and optimization. In this case, PSO method was more effective being able to accomplish a decrease in the tension level of the mooring system.

As can be seen, lots of previous work have been done using bio-inspired algorithm for structural optimization.

2.2 - Genetic Algorithms – GA

Genetic algorithms are mathematical algorithms inspired by the Darwin mechanisms of natural evolution, genetic recombination and mutation (Goldberg, 1989). According to C. Darwin, the fittest individuals had a bigger probability of reproduction and along with that their descendant keep the good genetic material in the species. This genetic material is represented by the chromosome, which constitute the identity of every individual.

In GA, every individual along with its unique genetic material represents a possible solution to the problem. Each optimization variable or parameter (x_n) is encoded by a gene using an appropriate representation, such as a real number or a string of bits. The corresponding genes for all parameters \( x_1, x_2, ..., x_n \) outline a chromosome, capable of describing an individual design solution. A set of chromosomes representing several individual solutions comprises a population. During the process of evolution, every individual is subjected to mate and reproduce. Matting is performed using crossover to combine genes from different parents to produce children (offspring). The children inherit features from each of the parents, and may be submitted to mutation, which confer some truly innovative features as well. The offspring are made to compete with each other, and possibly with their parents. Individuals are evaluated via the objective function that defines the problem. As a result of the evaluation, they are assigned a certain cost that discriminates between them. This value, named fitness value, represents the quality of the solution. By the end of a generation, only the fittest individuals are selected to continue in the population, and the other ones are rejected. Improvement in the population arises as a consequence of the repeated selection of the best parents, which are in turn more likely to produce good offspring, and the consequent elimination of low-performers.

The GA optimization algorithm can be summarized as follows:

1. Generate a random initial population of N individuals;
2. Compute the fitness values of the N individuals;
3. Select individuals to reproduction;
4. Apply crossover and mutation operator;
5. Compute the fitness values of the N individuals;
6. Select the N best individuals to compose the new population;
7. Repeat steps 3-6 until it reaches a pre-defined stopping criteria.

In the present work, the classical Genetic Algorithm is used, with binary codification, single point crossover, one individual elitism and roulette-wheel selection. As can be seen in section 3, there are 6 variables in the optimization process and the employed number of bits in the binary codification is 7, 6, 7, 3, 4 and 4 bits respectively.
2.3 - Particle Swarm Optimization – PSO

The Particle Swarm Optimization (PSO) method is a member of the wide category of Swarm Intelligence methods (Kennedy, Eberhart and Shi, 2001), for solving optimization problems. It was originally proposed as a simulation of social behavior, and it was initially introduced as an optimization method in 1995 (Kennedy and Eberhart, 1995).

The development of PSO was based on observations of the social behavior of animals such as bird flocking and fish schooling. Several modifications on the original paradigm showed that the algorithm could optimize complex functions based in a concept of swarm theory (Kennedy and Eberhart, 1995). Like GAs, PSO consist of populations of individuals representing candidate solutions to a problem, but instead of an evolutionary and selection pressure, swarm searches are guided by a social pressure. 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 is 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 two \( L \)-Dimensional vectors: a position vector \( X \) and a velocity vector \( V \).

The fly of a particle \( i \) is expressed in Eq. (1) as a movement from position \( x \) at iteration \( t \) with velocity \( v \):

\[
X_{i}(t+1) = X_{i}(t) + v_{i}(t+1) \cdot \Delta t
\]

Where \( \Delta t \) is the time step, usually equal to 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 any member of its neighborhood \( P_{g} \):

\[
v_{i}(t+1) = \omega \cdot v_{i}(t) + \phi_{1} \cdot \text{rand} \cdot (P_{i} - x_{i}(t)) + \phi_{2} \cdot \text{rand} \cdot (P_{g} - x_{i}(t))
\]

Where \( \omega \) is the inertia weight introduced in order to support with the balance between exploration and exploitation. The constants, \( \phi_{1} \) and \( \phi_{2} \), are employed to determine the balance between the influence of the individual’s knowledge (\( \phi_{1} \)) and that of the neighborhood (\( \phi_{2} \)). Those coefficients are called cognitive and social parameters, respectively, as they can weigh the cognitive learning. The values of \( \text{rand}() \) are random numbers from uniform distributions in the range \([0,1]\).

The PSO optimization algorithm can be written as follows:

1. Generate a random initial swarm of particles, assigning each one a random position and velocity;
2. Compute the fitness values of the \( N \) particles;
3. Update the values of the best position of each particle (\( P_{i} \)) and the best position found by the swarm (\( P_{g} \));
4. Update the position and the velocity of every particle according to Eq. (1) and (2);
5. Repeat steps 2-4 until it reaches a pre-defined stopping criteria.

Some adjustments have been made to the standard PSO algorithm over the past decade (Pina et al. 2008). Here it is presented these improvements, specifically the ones we used.

Albrecht (2005) proposed a term called social attraction correlated to passive congregation force, representing the influence of the group on the individual. This term involves the center of mass of the swarm. The new term added is defined next in Eq.(3).

\[
\phi_{3} \cdot \text{rand}() \cdot (C_{m} - x_{i}(t))
\]

Where \( \phi_{3} \) is the passive congregation coefficient; \( C_{m} \) is the center of mass of the swarm, where the mass of the particle is represented by its fitness value.

\[
C_{m} = \frac{\sum_{i=1}^{N} F_{i} \cdot x_{i}}{\sum_{i=1}^{N} F_{i}}
\]

Where \( F_{i} \) is the fitness value for the \( i \)-th individual; \( N \) is the number of individuals; \( x_{i} \) is the position of the particle.

These modifications results in some modification in the velocity equation (Eq.(2)).

\[
v_{i}(t+1) = \omega \cdot v_{i}(t) + \phi_{1} \cdot \text{rand} \cdot (P_{i} - x_{i}(t)) + \phi_{2} \cdot \text{rand} \cdot (P_{g} - x_{i}(t)) + \phi_{3} \cdot \text{rand} \cdot (C_{m} - x_{i}(t))
\]

The inertia coefficient \( \omega \) that was initially a constant value was modified by Albrecht (2005) and transformed it into a non-linear variation function \( \omega(t) \).

\[
\omega(t) = K \cdot \frac{(t-1)^{n}}{N}
\]

Where \( K \) is the non-linear variation control element; \( n \) is the non-linearity exponent and \( t \) is the iteration of the evolution process.

Another modification used was a linear variation of the aggregation and congregation coefficients (Ratnaweera et al. 2004). The equations for these variations are given in Eq.(7) to (9).

\[
\phi_{i}(t) = (\phi_{i,0} - \omega_{i}) \cdot \frac{t}{N} + \omega_{i}
\]
Where $\phi_1\text{ini}$, $\phi_2\text{ini}$, $\phi_3\text{ini}$ are the initial values of the coefficients, and $\phi_1\text{fin}$, $\phi_2\text{fin}$, $\phi_3\text{fin}$ are its final values.

In this algorithm a real codification was used for the variables representation.

### 2.4 - Artificial Immune System – CLONALG

Artificial immune systems appeared due to attempts of engineers and computer scientists to simulate particular immunological mechanisms with the objective of creating artificial systems to solve engineering problems. These attempts appeared with the existing analogies between the patterns recognition of the immunological system and the patterns recognition in computation, detection and virus elimination in the organism and network invasion detection.

AIS follows ideas taken from immunology in order to develop systems capable of performing different tasks in various areas of research like pattern recognition, detection of flaws and anomalies, computational security, optimization, control, robotics, scheduling, data analysis and machine learning. The clonal selection principle that is able to explain the basic features of an immune response to an antigen stimulus inspired the development of powerful computational tools (Castro et al., 2001). This principle is associated to the basic features of an adaptive immune response to an antigenic stimulus. It establishes that only the cell that is able to recognize a certain antigenic stimulus will proliferate. This immune response is specific to each antigen. After this recognition the cell will proliferate by cloning. Some of the new cloned cells will be differentiated into plasma cells and memory cells. The plasma cells are the most active antibodies secretors and will suffer mutation at high rates (hypermutation) that will promote their genetic variation. The memory cells are responsible for the immunologic response to future antigen invasion. Next, the selection mechanism will keep the cells with the best affinity to the antigens in the next population.

The AIS considered in this work uses the Clonal Selection Algorithm – CLONALG (Castro et al., 2001) to implement the optimization procedure. This algorithm was initially proposed to carry out machine-learning and pattern recognition tasks, and then adapted to solve optimization problems.

The basic CLONALG optimization algorithm may be written as follows:

1. Generate a random initial population of antibodies (Ab) of size $N$ that represents the candidate solution of the problem;
2. Evaluate affinity values of the Ab population;
3. Generate $N_c$ clones by cloning all $N$ cells in the Ab population;
4. Mutate the clone population to produce a mature clone population;
5. Evaluate affinity values of the $N_c$ clones population;
6. Select the $N$ best Ab within each family to compose the new Ab population;
7. Repeat steps 3-6 until it reaches a pre-defined stopping criteria.

The reproduction is made by cloning all antibodies of the population, giving rise to a temporary population of clones.

In this work, fitness and affinity have the same meaning; both measure the solution quality, however, they are presented with different names to respect the biologic inspiration of each method.

The somatic hypermutation is the next event in the maturation of the immune cells, after the gene rearrangement, which can improve the affinity of the antibodies. The mutation rate applied to every immune cell is inversely proportional to its antigenic affinity. The mutation changes all attributes of an antibody vector according to the following expression (Castro et al., 2001):

$$m' = m + \exp(-\rho \cdot D^*)$$  \hspace{1cm} (10)

Where $m = \{m_1, m_2, ..., m_I\}$ is the attribute string, $m'$ its mutated version, $\rho$ is a parameter that control the smoothness of the inverse exponential, $D^*$ is the normalized affinity, that can be determined by $D^* = D/D_{\text{max}}$.

In step 5, the best antibody is selected deterministically between the cells of each antibody family, by comparing each original antibody only with its clones. This mechanism can guarantee a good diversity during the evolution, allowing the algorithm to simultaneously explore different points of the search space. Since we are interested in reducing the population size in order to reduce the number of function evaluations, this method allows the use of a very small number of antibodies without losing diversity.

This characteristic, along with the affinity proportional mutation rate, provides the ability to search around every “family” according to its affinity. It is expected that this mechanism contributes to the aforementioned objective of its study: find the optimization method that is capable of reaching the optimum solution within a few fitness evaluations.

In order to enhance the performance of AIS, it was studied the relation between the normalized affinity (fitness) and the mutation rate ($\exp(-\rho \cdot D^*)$) depicted in the next figure.
In Figure 2, it can be seen that each value of $\rho$ results in a different mutation rate represented as $[\exp(-\rho \cdot D^*)]$. A smaller value of $\rho$ presents a higher level of variation. For example, when $\rho = 1$, even the best individual ($D^* = 1.0$) will show a high mutation rate (0.3679) resulting in a bad local search. For greater values of $\rho$, the local search sensitivity enhances, but the global search capability is almost lost. In such case, it was decided to connect the $\rho$ value to the quality of each individual, defining when it should behave like a local search or a global search. So it is proposed a linear variation of $\rho$. The formulation is depicted next.

$$m^* = m + \exp(-\rho_{Ad} \cdot D^*) \quad (11)$$

Where $\rho_{Ad}$ is the adaptive value of $\rho$. It is a function of the affinity value of each antibody.

$$\rho_{Ad} = \left[ \frac{\rho_{\text{max}} - \rho_{\text{min}}}{F_{\text{max}} - F_{\text{min}}} \right] F + \left[ \frac{\rho_{\text{max}} - \rho_{\text{min}}}{F_{\text{max}} - F_{\text{min}}} \right] \left( \frac{\rho_{\text{max}} - \rho_{\text{min}}}{F_{\text{max}} - F_{\text{min}}} \right) \quad (12)$$

Where $\rho_{\text{min}}$ is the minimum value and $\rho_{\text{max}}$ is the maximum value for $\rho_{Ad}$ both chosen by the user, $F$ is the affinity value for each antibody, $F_{\text{min}}$ is the minimum value of affinity among all antibodies and $F_{\text{max}}$ is its maximum value. In this way, the antibody with minimum affinity will receive the smallest value of $\rho_{Ad}$ and the one with maximum affinity will receive the highest value of $\rho_{Ad}$.

This modification tends to give more specific search ability to all antibodies, allowing the best ones to be responsible for the local search and the worst ones for the global search.

In this algorithm a real codification was used for the variables representation.

3 - PROBLEM REPRESENTATION

As mentioned in the introduction, the objective of this work is to compare three different nature-inspired optimization methodologies aiming to find the one capable of finding the best solution with the smallest number of fitness evaluation. This section describes the variables that are considered for the optimization process, employing real-valued shape-space.

Figure 3 presents a schematic model showing the parameters that define a lazy-wave riser system. The geometric riser parameters are:

- $L_1 =$ length of lower riser segment;
- $L_2 =$ length of segments with distributed floaters;
- $L_3 =$ length of top segment of the riser;
- $\alpha =$ the “top angle”, or the angle of the riser axis with the vertical direction at the connection with the platform, measured in the neutral equilibrium configuration;
- $Z =$ the depth of the connection, and
- $P =$ the horizontal projection.

Since the horizontal projection $P$ and the depth $Z$ are dictated by the characteristics of the platform and well connections, and the angle $\alpha$ is related to the projection $P$ and the total length ($L_1 + L_2 + L_3$), only these latter geometric parameters need to be considered in the optimization process.

There are also the parameters related to the buoys, which are:

- $L_f =$ buoy length;
- $H_{df} =$ buoy diameter;
- $\text{ESP} =$ spacing between buoys.

Other parameters such as the specific weight and other mechanical properties of the buoys could be considered; however in this work only the geometric parameters $L_f$, $H_{df}$ and ESP will be optimized.

Therefore, there are six parameters to be optimized, in order to determine a riser configuration that complies with all technical standards and design criteria, and presents the lowest construction cost. For this purpose, the following cost function is used:

$$f = \frac{1}{J_{\text{max}}} \left[ \sum_{i=1}^{n} C_{i} \cdot L_{i} \right] + \left( C_{\text{buoy}} \cdot \text{ESP} \right) \quad (13)$$

Where $i = 1..n$ represents the number of segments of the riser; $C_i$ is the cost index associated to each segment; $L_i$ is the segment length; $C_{\text{buoy}}$ represents the cost index associated to the volume of the buoy; is the volume of the buoy and is the maximum value of the objective function in order to normalize the fitness value.

Since this is a minimization problem with constraints, the fitness function will be defined as:

$$\text{affinity/fitness} = \left( f + \sum P_j \right)^{-1} \quad (14)$$

Where $\sum P_j$ is the sum of all penalties.

The structural behavior constraints are determined from the
results of structural analyses. At this point, it is interesting to note that, although the final optimization procedure must rely in a full non-linear, time-domain dynamic Finite Element solver, in the current application the evaluations are performed using an analytical catenary solver, which is much faster to compute and provides results that are at least representative of the actual FE (Finite Element) solution. This is because the main objective of this work is to compare three nature inspired methodologies in the search of the best performance in reducing the number of evaluations. In actual design applications, the evaluations will be performed by nonlinear dynamic FE analyses.

In any case, the structural behavior constraints are (Vieira, 2008):

- The maximum tension at the riser top (also dictated by the design of the flex-joint); and
- The minimum tension at the riser bottom (to avoid buckling and collapse of a riser section);
- The maximum angle between the riser axis and the vertical direction at the connection with the platform (dictated by installation requirements);
- The maximum variation of the "built-in" angle, measured at the top riser axis, between the neutral equilibrium configuration and any configuration acquired by the riser during the application of the environmental loadings and the platform motions (dictated by the design of the flex-joint that provides an articulated connection of the riser with the platform);
- The maximum equivalent Von Mises stress acting on the riser sections (to assure the structural integrity of the pipe);

Therefore, there are five constraints, and the violation of any one of them results in a penalty given by the following equation:

\[ P_j = \begin{cases} k \cdot (1 - x), & \text{if } x < 1 \\ 0, & \text{if } x \geq 1 \end{cases} \]  

(16)

Where \( P_j \) is the penalty value of the \( j \)-ith constraint criteria, \( x \) is the ratio between \( j \)-ith constraint limit value and its calculated value and \( k \) is a factor that allow the emergence of non constrained solutions. In some cases, the best solutions in the generation are the penalized ones, so, this coefficient \( (k) \) can increase the penalization of those solutions. In this work, \( k \) value is augmented in 10% every time the best solution violates any structural behavior constraint. The employed initial value of \( k \) is equal to 1.

The optimization was made given lazy-wave riser configuration, to be installed at a sea depth of 1290m, and considering a horizontal projection of 2000m. Specific data related to the riser modeling are depicted in Table 1. In this table, the cost ratio \( C_2/C_1 \) means that the segment with floaters costs two times more than the regular riser segments.

### Table 1 – Riser modeling data

<table>
<thead>
<tr>
<th>MATERIAL</th>
<th>GEOMETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>Thickness</td>
</tr>
<tr>
<td>7800 kg/m³</td>
<td>0.01905 m</td>
</tr>
<tr>
<td>Specific weight</td>
<td>External diameter</td>
</tr>
<tr>
<td>77 kN/m³</td>
<td>0.21908 m</td>
</tr>
<tr>
<td>Yield stress</td>
<td>Internal diameter</td>
</tr>
<tr>
<td>413 MPa</td>
<td>0.18098 m</td>
</tr>
<tr>
<td>Allowable stress</td>
<td>Floater weight</td>
</tr>
<tr>
<td>277 MPa</td>
<td>0.162 ton/m</td>
</tr>
<tr>
<td>Elastic modulus</td>
<td>Floater buoyancy</td>
</tr>
<tr>
<td>207800 MPa</td>
<td>0.3175 ton/m</td>
</tr>
<tr>
<td>Cost ratio C2/C1</td>
<td>Floater external diameter</td>
</tr>
<tr>
<td>2.0</td>
<td>0.568 m</td>
</tr>
</tbody>
</table>

The user-defined bounds for the riser structural behavior constraints are displayed in Table 2.

### Table 2 – Riser optimization parameters

<table>
<thead>
<tr>
<th>DESIGN LIMITS (meters)</th>
<th>MIN</th>
<th>MAX</th>
<th>CONSTRAINTS</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Riser segment (L1)</td>
<td>800</td>
<td>2000</td>
<td>Von Mises Stress</td>
<td>415.5MPa</td>
</tr>
<tr>
<td>Riser segment (L2)</td>
<td>400</td>
<td>800</td>
<td>Maximum top angle</td>
<td>18°</td>
</tr>
<tr>
<td>Riser segment (L3)</td>
<td>800</td>
<td>2000</td>
<td>Minimum top angle</td>
<td>5°</td>
</tr>
<tr>
<td>Buoy diameter (HDf)</td>
<td>0.5</td>
<td>2</td>
<td>&quot;Built-in&quot; angle variation</td>
<td>5°</td>
</tr>
<tr>
<td>Buoy length (Lf)</td>
<td>0.5</td>
<td>2</td>
<td>Maximum top Stress</td>
<td>1500 kN</td>
</tr>
<tr>
<td>Spacing between buoy (Esp)</td>
<td>0.8</td>
<td>1.5</td>
<td>Minimum Stress</td>
<td>300 kN</td>
</tr>
</tbody>
</table>

### 4 - COMPARATIVE STUDY

#### 4.1 - Optimization parameters

In this section it will be presented the parameters used for each optimization methodology for the comparative tests. The stopping criteria used was a fixed number of fitness evaluations. This criterion was used in order to compare the efficiency of the different optimization methodologies. It was chosen 5 numbers of fitness calculations: 1200, 1440, 1680, 2400 and 4800. These numbers were chosen by previous experience of the authors in analyzing this study case with GA (Vieira, 2008b).

For tests conducted with AIS, it was used 4 antibodies, 9 clones. In the first case study it was employed a fixed value of \( \rho = 2 \) (Eq. (10)), and in the second case, a linear variation of \( \rho \) with ranges between 1 and 3 (according to the affinity value).

The parameters employed in tests with GA were: 100 individuals, crossover probability equal to 0.80, and mutation rate equal to 0.05.

The employed PSO parameters were: 25 particles, \( \varphi_{1 ini} = 1.0, \varphi_{2 ini} = 1.0, \varphi_{lim} = 1.0 \) and \( K = 1.2 \) (detailed in Eq.(6)).

We can summarize the algorithms configurations in the next table.
Table 3 – Algorithms parameters

<table>
<thead>
<tr>
<th>NAME</th>
<th>ALGORITHMS</th>
<th>INDIVIDUALS</th>
<th>CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIS_1</td>
<td>AIS</td>
<td>40 indiv = 4 Ab, 9 clones</td>
<td>( \rho = 2 )</td>
</tr>
<tr>
<td>AIS_2</td>
<td>AIS</td>
<td>40 indiv = 4 Ab, 9 clones</td>
<td>( \rho = 1 \text{ to } 3 ) (linear Variation)</td>
</tr>
<tr>
<td>GA_1</td>
<td>AG</td>
<td>100 indiv</td>
<td>( P_{\text{cross}} = 0.80, P_{\text{mut}} = 0.05 )</td>
</tr>
<tr>
<td>PSO_1</td>
<td>PSO</td>
<td>25 indiv</td>
<td>( \Phi_{\text{ini}} = 1.0, \Phi_{\text{fin}} = 2.0, K=1.2 )</td>
</tr>
</tbody>
</table>

These values were chosen due to previous studies employing PSO (Pina et al., 2008 and Albrecht, 2005), GA (Vieira, 2008b) and AIS (Vieira et al., 2008a).

4.2 - Results

The results of the tests are depicted in the next table. All the algorithms were executed 10 times, and the results are compared in terms of the mean fitness value of the fittest individual at the last evaluation.

Table 4 shows the results for all the algorithms. As it can be seen, the mean value of fitness is growing as the number of evaluation increases. The best mean fitness achieved with 1200 evaluations was 1.585 (PSO_1), followed by 1.567 (AIS_2). From 1440 to 2400 evaluations, AIS_2 is leading followed closely by PSO_1. Comparing the standard deviation, we can see that AIS_2 is better than AIS_1 and PSO_1 has its smallest value. The GA_1 showed to be the worst of the algorithms, presenting the lowest mean fitness value.

Table 4 – Summary of Fitness values

<table>
<thead>
<tr>
<th>NAME</th>
<th>NUMBER OF EVALUATIONS</th>
<th>1200</th>
<th>1440</th>
<th>1680</th>
<th>2400</th>
<th>4800</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIS_1</td>
<td>MAX</td>
<td>1.624</td>
<td>1.634</td>
<td>1.624</td>
<td>1.631</td>
<td>1.627</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
<td>1.567</td>
<td>1.576</td>
<td>1.594</td>
<td>1.606</td>
<td>1.615</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>1.453</td>
<td>1.454</td>
<td>1.552</td>
<td>1.566</td>
<td>1.595</td>
</tr>
<tr>
<td></td>
<td>STD DEV</td>
<td>0.0523</td>
<td>0.0554</td>
<td>0.0244</td>
<td>0.0242</td>
<td>0.0113</td>
</tr>
<tr>
<td>AIS_2</td>
<td>MAX</td>
<td>1.624</td>
<td>1.626</td>
<td>1.629</td>
<td>1.630</td>
<td>1.632</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
<td>1.567</td>
<td>1.589</td>
<td>1.600</td>
<td>1.618</td>
<td>1.621</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>1.453</td>
<td>1.538</td>
<td>1.543</td>
<td>1.607</td>
<td>1.608</td>
</tr>
<tr>
<td></td>
<td>STD DEV</td>
<td>0.0236</td>
<td>0.0342</td>
<td>0.0316</td>
<td>0.0057</td>
<td>0.0069</td>
</tr>
<tr>
<td>GA_1</td>
<td>MAX</td>
<td>1.526</td>
<td>1.494</td>
<td>1.572</td>
<td>1.538</td>
<td>1.606</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
<td>1.395</td>
<td>1.423</td>
<td>1.424</td>
<td>1.456</td>
<td>1.409</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>1.296</td>
<td>1.336</td>
<td>1.286</td>
<td>1.361</td>
<td>1.178</td>
</tr>
<tr>
<td></td>
<td>STD DEV</td>
<td>0.0630</td>
<td>0.0402</td>
<td>0.0861</td>
<td>0.0537</td>
<td>0.1238</td>
</tr>
<tr>
<td>PSO_1</td>
<td>MAX</td>
<td>1.623</td>
<td>1.631</td>
<td>1.606</td>
<td>1.631</td>
<td>1.630</td>
</tr>
<tr>
<td></td>
<td>MEAN</td>
<td>1.585</td>
<td>1.583</td>
<td>1.595</td>
<td>1.612</td>
<td>1.610</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>1.530</td>
<td>1.538</td>
<td>1.565</td>
<td>1.590</td>
<td>1.585</td>
</tr>
<tr>
<td></td>
<td>STD DEV</td>
<td>0.0287</td>
<td>0.0245</td>
<td>0.0119</td>
<td>0.0136</td>
<td>0.0138</td>
</tr>
</tbody>
</table>

Analyzing the previous table, one can see that AIS and PSO are able to achieve a high performance with few evaluations if compared to GA. Both methods achieved good results with small standard deviation.

5 - FINAL REMARKS

This work intends to compare three different bio-inspired optimization methodologies applied to a structural optimization problem, that is, definition of an optimal lazy-wave SCR riser configuration for deepwater oil exploitation activities. This problem requires cost-effective solutions where both the cost and the complexity of the structures tend to increase, therefore motivating studies on optimization procedures.

The evaluation of the behavior of riser configurations requires Finite Element structural analyses employing a non-linear time-domain dynamic solver which are extensively time consuming. So, the main objective of this work is to obtain the best results with the minimum number of objective function calculations.

The comparison between the bio-inspired optimization algorithms showed that AIS and PSO performance are very close and present the best results. GA exhibited the worst results. AIS despite showing a bigger standard deviation compared to PSO, achieved higher mean values, endorsing its good optimization capacities.

Acknowledgements

The authors are grateful to the Brazilian research agency CNPq for the financial support of this research project.

REFERENCES


