Optimization of FPSO spread mooring systems with a surrogate-assisted Differential Evolution algorithm

Vinicius G. do Prado[1], Bruno da F. Monteiro[2], Beatriz S. L. P. de Lima[2], Breno P. Jacob[2]

[1] *Petrobras – Petróleo Brasileiro S/A*
Av. República do Chile, 330, 20031-170, Rio de Janeiro/RJ, Brazil
vgprado@petrobras.com.br
[2] *LAMCSO – Laboratory of Computer Methods and Offshore Systems, COPPE/UFRJ*
Avenida Pedro Calmon, S/N, 21941-596, Rio de Janeiro/RJ, Brazil
bruno.monteiro@lamcso.coppe.ufrj.br, bia@coc.ufrj.br, breno@lamcso.coppe.ufrj.br

Abstract. The design of mooring systems is a complex and time-consuming task that must be thoroughly addressed in every Oil & Gas upstream project. Up to now, the task is performed mostly based on the expertise and engineering judgment of the analysts, with little to no optimization ever pursued. This article presents a method that employs the well-known ε-Constrained Differential Evolution algorithm to design the mooring system and makes use of Artificial Neural Networks to evaluate its performance, thus eliminating the constraints imposed by the limited capabilities of the human mind and providing feasible systems with reduced costs.

Keywords: mooring, FPSO, optimization, differential evolution, neural networks.

1 Introduction

FPSOs (Floating Production Storage Offloading units) moored in deep water fields embody the current trend of the Oil and Gas industry in Brazil and many other regions around the world, with a recent spike in projects entailed by increased oil prices. The mooring systems of these units are sized to guarantee that their maximum offsets will not exceed the design conditions of the risers, the dynamic sections of pipelines that interconnect the floating units and the subsea wells, while also complying with the structural requirements presented in the rules of Classification Societies and regulations like ISO 19901-07 [1] and API RP 2SK [2].

The mooring system analyses in time-domain constitute one of the most complex, time-consuming, and computer-intensive tasks involved in the subsea layout definition for upstream projects. The input data is plenty and the conclusions, hard to draw from the torrents of raw data available from the simulations. The interplay between the many design variables of the problem is hard to grasp. Because of that, up to now, the design of the mooring systems has relied mostly on the expertise and engineering judgment of the analysts in charge of the projects, with little to no optimization ever performed.

A different approach for the design and optimization of mooring systems is presented in this article. The method employs the well-known ε-Constrained Differential Evolution algorithm to optimize the mooring system and makes use of a metamodel (or surrogate model), by Artificial Neural Networks (ANNs), to evaluate the performance of the candidate solutions in extreme environmental conditions. The single objective of the algorithm is to minimize the cost of the mooring system and the constraints are basically related to the maximum offsets of the floating unit and maximum tensions on the mooring lines.

The design of cost-effective mooring systems has already been formally described as an optimization problem in many previous works and the use of evolutionary algorithms to tackle the issue is not new [3 to 6]. These algorithms perform very well on a variety of problems but require a great number of candidate evaluations to achieve convergence. That being so, the methods adopted for estimating the performance of the candidate solutions varies in these works, with some being more precise than others and posing a clear trade-off between precision of the results and the computational efforts required to deliver them. In this context, the use of metamodels (also known as surrogate models), to approximate the results from time-domain simulations, provides both reliable estimations and reduced processing times. We hope that these features will render the proposed method attractive as a design tool for the upstream projects still to come.

This article is organized as follows: the mooring system optimization problem is defined, the design methodology is described together with the metamodel and the optimization algorithm, then the case of a typical spread-moored FPSO in the Pre-Salt area is discussed to demonstrate the auspicious results that were obtained.

2 Mooring system optimization problem

The mooring system optimization problem is defined as follows. The cost of the system installed offshore is the objective function to be minimized. The maximum tensions acting on the mooring lines, in line with ISO and API requirements, and the maximum FPSO offsets defined by the operator are taken as constraints. In addition, polyester segments should not touch the seabed. Therefore, the mooring system optimization problem addressed in this work takes the form:

$$Minimize\ CAPEX(x) \tag{1}$$

Subject to:

$$\frac{\max\ Offset}{MAO} \leq 1 \tag{2}$$

$$\frac{\max\ T_{top}}{MBS} \leq \mathrm{SF} \tag{3}$$

$$\frac{\max\ T_{bottom}}{MAHP} \leq \mathrm{SF} \tag{4}$$

$$\frac{\max\ L_{seabed}}{L_{bottom\ chain}} \leq 1 \tag{5}$$

Where:
x = mooring system configuration
CAPEX = Capital Expenditure of the mooring system (cost of materials and installation)
MAO = Maximum allowable Offset
MBS = Minimum Breaking Strength (corroded)
MAHP = Maximum Anchor Holding Power
SF = safety factor, defined by ISO and API regulations
T_{top} = tension on the line at the FPSO fairlead level
T_{bottom} = tension on the line at the mudline penetration point
L_{seabed} = length of mooring line that lays on the seabed
$L_{bottom\ chain}$ = length of bottom chain

The candidate solution *x* may comprise every design variable that define the mooring system, e.g., the number of lines, the azimuths, the radii and the pre-tension levels of each cluster as well as the composition of the lines, in terms of size and strength of materials and lengths of segments. In our case study, however, only number of lines, radii and pre-tension levels were controlled. Pre-tension was adjusted by changing the lengths of the top polyester segments, while all other segments were maintained fixed.

3 Design methodology

3.1 Overview

A number of works have already been published in this line of research, with different objective functions and different approaches to the evaluation of the candidate solutions. In Monteiro et al. [3], the Particle Swarm Optimization (PSO) algorithm was applied and the candidates were evaluated by fully coupled nonlinear time-domain Finite Element simulations. The objective function is the maximization of the vessel offsets inside the feasibility limits of the riser system. In Carbono et al. [4], the Genetic Algorithm (GA) was used to minimize the vessel offsets, calculated by static analyses. Shafieefar and Rezvani [5] also used GA to minimize the vessel offsets, but evaluated the candidate solutions via time-domain analyses. Monteiro et al. [6] compared the performances of the PSO and DE algorithms applied to mooring optimization problems and concluded that DE presents faster convergence. In this study, the constrained problem was converted into an unconstrained problem by application of penalties in the objective function. Pina et al. [7] demonstrated that ANNs can approximate results from mooring analyses in time-domain with reasonable precision.

Inspired by Garcia et al. [8], which used metamodels in constrained optimization problems, we employ the ε-Constrained Differential Evolution algorithm coupled with ANNs to optimize mooring systems. Each candidate solution is assessed in terms of violation of constraints by the ANNs, trained from a dataset comprising 3000 random solutions evaluated by quasi-dynamic simulations. The algorithm evolution proceeds until no constraint violation is detected in the best individual of the generation and no additional decrease in the cost of the system is attained in a pre-defined number of generations. The best individual is then evaluated by fully coupled nonlinear time-domain Finite Element analyses to confirm its feasibility.

3.2 Artificial Neural Networks

ANNs have been studied since the 1940s and are one of the most widely known Artificial Intelligence techniques, in great part due to its flexibility and quality of estimations provided. ANNs are composed by processing units, the neurons, disposed in layers. Each neuron takes a linear combination of its weighted inputs and processes it via a user-defined activation function that may take many forms. The neurons of a layer are connected to all neurons of the previous and next layers. Besides the input and output layers, most applications are satisfied with one intermediate (or hidden) layer [11]. See Fig. 1.

The computer is able to learn complex non-linear relationships between inputs and outputs by adjusting the weights of the net. To achieve this goal, it is necessary to feed the ANN with vast amounts of known input-output pairs, the training data. The predictions of the net are compared to the real outputs and the error is used, by means of gradient-based methods, to adjust the weights of the net in an iterative process known as back-propagation.

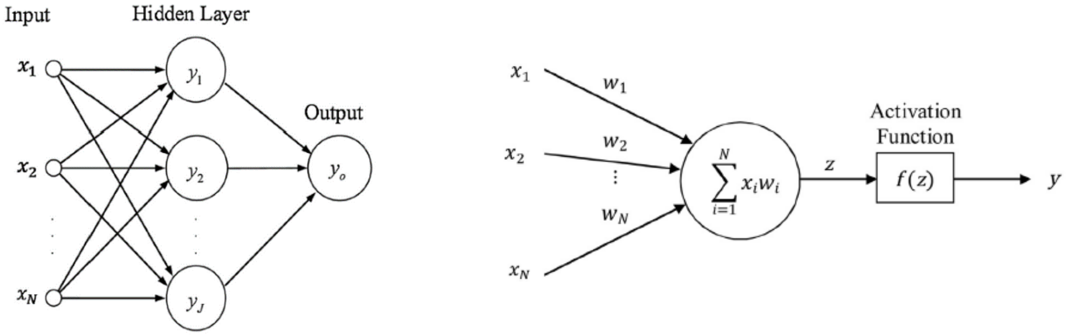

Figure 1. Neural network with one hidden layer (left) and artificial neuron (right)

3.3 ε-Constrained Differential Evolution Algorithm

The optimization algorithm used in this work was proposed by Takahama and Sakai [9]. Individuals are ranked according to the ε-comparison, a routine that compares the sets of values (f,φ) assigned to each individual in such a way that the minimization of the constraint violation φ precedes the minimization of the objective function f. The ε-level represents the maximum allowable constraint violation and is defined as a function of the number of generations, slowly decreasing to zero. The algorithm also employs gradient-based mutation and elitism to enhance feasibility of the population throughout the generations. The parameters that need to be set are the size of the population, the number of generations, the amplification factor F and the crossover rate CR. The pseudo-code of the algorithm is presented below.

```
Generate initial population
Do
  For each individual x in the population
  Choose 3 individuals a, b and c from the population different from x
  Set v = a + F.(b-c)
  Define z by exponential crossover between x and v with probability CR
  Gradient-based mutation of the new individual
  Replace x with new individual based on ε-comparison
  Set ε-level
Until termination condition is achieved
```

Figure 2. Pseudo-code of the ε-Constrained Differential Evolution algorithm

In the exponential crossover, a point *k* is chosen from all dimensions of the vector *x* and is used as a starting point for the exchange of elements between vectors *x* and *v*. The elements from the vector *x* are replaced by elements from the vector *v* with an exponentially decreasing probability given by CR, thus defining the vector *z*.

Gradient-based mutation is used as a local search operator to improve feasibility of the new individuals. It is applied over the vector *z* with a 1% probability if the parent vector *x* is not ε-feasible. The operator is applied up to 3 times or until *z* becomes ε-feasible. The increment to *z*, Δ*z*, and the mutated vector *z'* are then given by:

$$\Delta z = -\nabla C(z)^{-1} \Delta C(z) \tag{6}$$

$$z' = z + \Delta z \tag{7}$$

Where $\Delta C(z) = \max_i\, (0, g_i(z))$ is a vector of constraint violations, with $g_i(z)$ representing each inequality constraint applicable to the problem, and $\nabla C(z)^{-1}$ is the pseudo-inverse of the gradient matrix, which is calculated by finite differences considering small perturbations over the vector *z*.

4 Case Study

4.1 Performance of the metamodel – ANN

To gather the training data required by the ANN for our case study, candidate solutions were randomly generated within the bounds of the design variables and evaluated by time-domain analyses of the critical environmental conditions found for a base case system defined *a priori*. A total of 3000 candidates were evaluated and their results were assembled into a dataset, regardless of whether they fulfilled the design constraints given in (2) through (5).

As is common practice, 70% of the set was used for the training set and 30% for the test set. Table 2 shows the Mean Absolute Error (MAE) and the Coefficient of Determination (R^2) obtained after training the ANN for the maximum FPSO offsets and maximum tensions on the mooring lines, demonstrating that the results of the mooring analyses can be reliably estimated by surrogate models.

Table 2. Precision measures of the ANN predictions for the test set

Variable	MAE	1-R
Offset/Max Offset	7.44E-02	2.38E-02
Max T_{top}/MBS	6.85E-02	1.61E-02
Max T_{bottom}/MAHP	3.61E-02	1.92E-02
Max L_{seabed}/L_{bottom_chain}	6.43E-02	4.50E-02

Figure 3 shows the comparison of the real outputs in the dataset (y_real) to the estimations of the ANNs (y_hat). Only the maximum FPSO offset is presented here for brevity. Blue dots represent the training set and orange dots represent the test set. Very good agreement between predictions and real data was achieved.

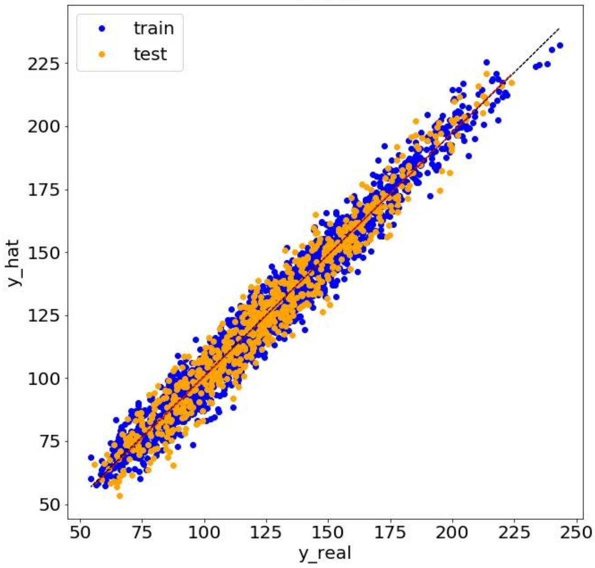

Figure 3. Comparison between maximum offsets calculated by FEM and estimations via metamodel

4.2 Performance of the ε-Constrained DE

The performance of the optimization algorithm was verified with some of the test functions given in the specialized literature [10]. Table 1 shows the comparison of the results obtained by our implementation and the minima of these functions. The size of the population, the number of generations, the amplification factor F and the crossover rate CR, were taken as 40, 1000, 0.7 and 0.9 respectively.

Table 1. Performance of the optimization algorithm

Test Function	Minima	ε-Constrained DE	Difference (%)
G01	-15.00000	-15.00000	0.00%
G04	-30665.53867	-30665.53867	0.00%
G06	-6961.81388	-6961.81388	0.00%
G05	5126.49671	5127.97007	0.03%
G03	-1.00050	-1.00000	0.05%
G11	0.74990	0.75000	0.01%
G15	961.71502	961.71517	0.00%

4.3 Application to mooring system optimization

The methodology was applied to the case of a typical VLCC-sized spread-moored FPSO in the Pre-Salt area, offshore Brazil, in a water depth of 2000 m. The unit is provided with 4 mooring clusters, one in each corner of the hull, and 32 risers connected to the riser balcony on the portside of the unit. The mooring lines are composed of segments of 120 mm R4 studless chains and polyester ropes of SWL 1250 t. The adopted maximum allowable FPSO offset and tensions at top and bottom of the mooring lines are 164 m, 6000 kN and 5400 kN respectively.

Applying the ε-Constrained DE algorithm to the problem and evaluating the candidate solutions via metamodels, a solution for the mooring system was found. Table 3 presents the best mooring system configuration found in 20 runs of the algorithm. Table 4 presents the performance of the best solution under 100-yr environmental conditions, calculated by fully coupled nonlinear time-domain Finite Element analyses.

Table 3. Optimized mooring system configuration

Mooring Cluster	Number of Lines	Radius/Depth	Pre-Tension [kN]
SE	4	1.02	2470
NE	4	1.16	1300
NW	4	1.11	2010
SW	5	1.05	2500

Table 4. Performance of the optimized system

Mooring Cluster	Max Top Tension [kN]	Max Bottom Tension [kN]	Maximum FPSO Offset [m]
SE	4823	4065	156
NE	4660	3905	
NW	5087	4350	
SW	4793	3895	

Figure 4 shows the fitness and constraint violation φ of the best solution of each generation along 100 generations of one run of the optimization algorithm. After finding a feasible region inside the search space, the fitness is improved, exactly as devised by the ε-Constrained DE algorithm.

While human experience would steer the new design towards solutions that are similar to what has already been done in the past, the algorithm tests a great number of alternatives, gathering valuable information about the problem at each iteration, to finally yield solutions with noticeable reduction in costs. Table 5 shows the mean, best and worst solutions found in these runs in terms of fitness (CAPEX in USD), demonstrating that the methodology is both stable and robust.

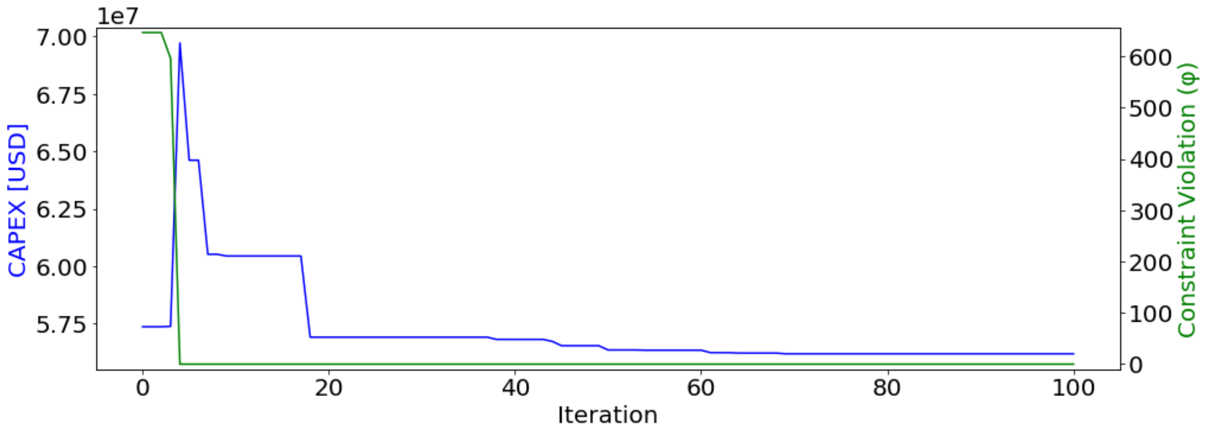

Figure 4. Optimization of the mooring system by the ε-Constrained DE algorithm

Table 5. Mean, best and worst solutions in 20 runs [CAPEX in MM USD]

Mean	Best	Worst	Std
56.43	52.10	60.25	2.72

5 Conclusions

The proposed methodology provides promising results, achieving a feasible solution with reduced cost in our case study. The ε-Constrained DE algorithm was devised for much more complex search spaces, therefore converges easily to feasible solutions with reduced objective function when applied to our problem. In addition, the gradient-based mutation routine handles feasibility issues very deftly.

Once the ANN is trained, the generation of new solutions is a matter of a few hours, mostly spent on running time-domain analyses to confirm the feasibility of the solution found by the algorithm. Since the generation of data for the ANNs can be de-coupled from the critical path of new projects, we believe that our proposal offers many advantages for practical applications in the industry. The expansion of the training dataset to encompass different scenarios (water depths, vessel headings, number of risers, etc) has the potential to transform the methodology into a design tool that can be coupled to subsea arrangement optimization routines.

In future works, we will study a method for identifying the critical environmental conditions of each candidate solution for the generation of the training dataset of the ANN, thus rendering the algorithm more precise. Data from fatigue analyses will also be incorporated to the problem, to enable a complete evaluation of the mooring system.

Acknowledgements. The authors would like to express gratitude towards SOFEC Mooring Solutions, which provided all the information required and lots of support for the mooring analyses in time-domain.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] ISO 19901-7, "Stationkeeping systems for floating offshore structures and mobile offshore units", International Organization for Standardization, 2013.

[2] API RP 2SK, "Design and Analysis of Stationkeeping Systems for Floating Structures", American Petroleum Institute, 2005.

[3] B.F. Monteiro, J.S. Baioco, C.H. Albrecht, B.S.L.P. de Lima, B.P.Jacob, "Optimization of mooring systems in the context of an integrated design methodology", *Marine Structures 75*, 102874, 2021.

[4] A.J.J. Carbono, I.F.M. Menezes, L.F. Martha, "Mooring Pattern Optimization Using Genetic Algorithms", *6th World Congress of Structural and Multidisciplinary Optimization*, 2005.

[5] M. Shafieefar, A. Rezvani, "Mooring Optimization of Floating Platforms Using a Genetic Algorithm", *Ocean Engineering, 34*, pp. 1413 - 1421, 2007.

[6] B.F. Monteiro, A.A. de Pina, J.S. Baioco, C.H. Albrecht, B.S.L.P. de Lima, B.P.Jacob, "Towards a Methodology for the Optimal Design of Mooring Systems for Floating Offshore Platforms Using Evolutionary Algorithms", *Marine Systems & Ocean Technology 11*, pp. 55 - 67, 2016.

[7] A.A. de Pina, B.F. Monteiro, C.H. Albrecht, B.S.L.P. de Lima, B.P. Jacob, "Artificial Neural Networks for the Analysis of Spread Mooring Configurations for Floating Production Systems", *Applied Ocean Research 59*, pp. 254 - 264, 2016.

[8] R.P. Garcia, B.S.L.P. de Lima, A.C.C. Lemonge, A Surrogate Assisted Differential Evolution to Solve Constrained Optimization Problems, *IEEE Congress on Computational Intelligence*, 2017.

[9] T. Takahama, S. Sakai, "Constrained Optimization by the ε-Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites", *IEEE Congress on Evolutionary Computation*, 2006.

[10] J.J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C.A.C. Coello, K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization", 2006.

[11] S. Haykin. *Neural Networks and Learning Machines*. Prentice-Hall, 2009.