Paper 66

©Civil-Comp Press, 2012
Proceedings of the Eighth International Conference
on Engineering Computational Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Optimization of Pipeline Routes using an AIS/Adaptive Penalty Method

R.R. de Lucena[1], B.S.L.P. de Lima[1], B.P. Jacob[1] and D.M. Rocha[2]
[1] COPPE/UFRJ, Post-Graduate Institute of the Federal University of Rio de Janeiro, Civil Engineering Department, Rio de Janeiro, Brazil
[2] PETROBRAS Research & Development Center CENPES, Rio de Janeiro, Brazil

Abstract

A computational tool for the synthesis and optimization of submarine pipeline routes has been developed as a result of previous research work. Such tool must rely on the accurate representation of the constraints associated to the design practice; the treatment of these constraints had been performed by a standard static penalty technique.

This work describes the implementation of an adaptive penalty method (APM), associated to the artificial immune systems meta-heuristic implemented on the route optimization tool. The performance of the APM is assessed with the results of a case study, indicating that this penalty-based approach is very efficient when compared to a static penalty technique for the treatment of constraints.

Keywords: optimization, nature-inspired algorithms, artificial immune systems, constraint-handling methodologies, adaptive penalty method, submarine pipeline routes.

1 Introduction

The discovery of offshore oil reserves on deep and ultradeep waters has increased the challenge for designers and engineers due to the need for increasingly complex structures subjected to extremely harsh conditions. In such scenarios, submarine pipelines are used to convey oil or gas. The high cost of installation, recovery and repair, associated with high risk of environmental damage, requires that design of these structures to be performed safely according with current standards.

The choice of the best pipeline route in offshore environments is one of the stages in a submarine pipeline project. The challenge is to find a feasible route, with the smallest length, leading to lower cost of materials and of future interventions. Other factors that affect the route performance must also be considered, including for instance geophysical/geotechnical data obtained from the seabed bathymetry and sonography; these data define the obstacles and regions that should be avoided, leading to the number of free spans along the route that should be mitigated.

Traditionally, the selection of a pipeline route has been performed manually by engineers and experts, through an inspection of the bathymetry and available information regarding obstacles. This stage of the project is very complex, governed by many variables and constraints, and should abide by standards such as DNV-OS-F101 [1]. Eventually the evaluation of some aspects of a given route could be performed using analysis tools, but in any case the process is highly dependent on the expertise of the engineer. Therefore, it should be recognized that the selection a submarine pipeline route with good performance and low cost must indeed be formally described and treated as a synthesis and optimization problem.

Optimization methods seek to find an optimal solution to a given engineering problem within a set of solutions, usually subject to constraints. Nature-inspired algorithms (NIAs) such as clonalg [2], which is based on Artificial Immune Systems (AIS), have been shown to be very useful for the search and optimization of solutions for offshore engineering problems [[3][4][5]].

In this context, a previous work [6] described the initial steps taken towards the development and implementation of a computational tool for the synthesis and optimization of submarine pipeline routes based on NIAs. In this tool, candidate routes are randomly generated and evaluated by an objective function that relates the length of the route and the violation degree of each constraint in order to find viable routes that minimize material and installation costs.

This work focuses on aspects related to how NIAs, originally designed to optimize unconstrained problems, deal with the violation degree of each constraint. In this context, the adaptive penalty method, known as APM [7], will be used to assist clonalg on choosing the “optimal solution”. The performance of this penalty-based approach will be compared with results obtained by the constraint-handling method previously used in the optimization tool.

The remainder of the paper begins with a brief description of the clonalg optimization method that has been implemented on the route optimization tool; proceeds with the parameterization and codification of a given pipeline route, and then presents the objective function and the penalty terms. These are responsible to represent a set of constraints present in an actual project. Then penalty-method approaches will be briefly discussed, with special attention to APM. The final sections present the results of a case study and the concluding remarks.

2 Artificial immune system

Different types of NIAs could be considered for the implementation of the optimization tool described in this work. For the preliminary implementation phase [6], genetic algorithm (GA) was used.

In this work clonalg was chosen to optimize a pipeline route. In this methodology, randomly generated candidate routes form an AIS population of antibodies (or cells). Each cell represents a candidate solution (or route) and its quality is based on its efficiency (or fitness). The evaluation is performed by means of an objective function, taking into account the problem constraints, as will be described later in this work.

Each cell present in the initial population produces a user-defined quantity of clones. From this moment, a new population of clones will become part of the algorithm evolutionary process. This new population will suffer a mutation process, based on the somatic hypermutation principle, giving genuinely new features to each clone. The amount of mutation that a clone cell will receive is governed by the affinity of this clone to the best solution ever found by the optimization algorithm. Thus, cells that have a good fitness value will be muted with low rates (the process of exploitation), while those that have low fitness will suffer mutations at high rates (scanning process). The somatic hypermutation is responsible for allowing the search process in this methodology.

After the mutation process, each clone will be evaluated by the objective function. The clonal selection principle will select the best cells among the original antibodies and their respective clones to compose the next generation of antibodies.

This process ends when a pre-defined stopping criterion is reached, and the cell with the best fitness is then defined as the "optimal solution" (or optimal route). The algorithm is summarized schematically in Figure 1.

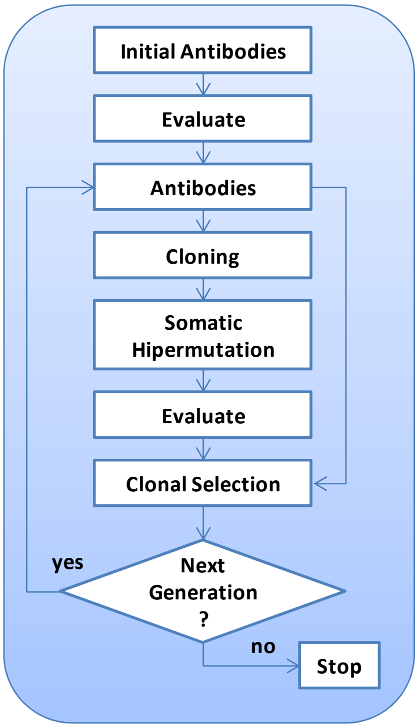


Figure 1. Clonalg flowchart

3 Route parameterization

The optimization process by NIAs requires a codification that uniquely represents each possible candidate solution (or pipeline route). This section describes the parameterization employed for the geometric representation of pipeline routes in the optimization tool.

A given route is defined by its endpoints A and B, and by a set of straight and curved sections. Curves, which are represented by intersection points (PIs), are necessary for a route to avoid constraints present on the seabed. The position of the set of PIs is associated with base points distributed evenly along a straight line connecting the endpoints A and B. The position of each PI relative to its base point is defined in terms of polar coordinates (the Radial (δ) and Angular (α)). This representation is shown in Figure 2. It has been proved in a previous work [6] that the straight lines and curves that comprise a route between points A and B can be completely described by a set of the following parameters for each curve: a) Its curvature radius R and b) The polar coordinates δ and α of the corresponding inflexion point (PI).

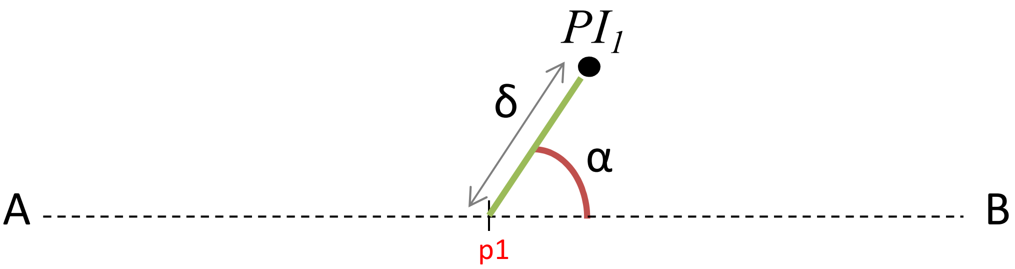


Figure 2. The polar coordinates from PI to its base point

Additionally, there is a fourth gene, associated with an activation key (*A*), that indicates the status of each PI. This allows the number of PIs to vary along the optimization process, beginning with a maximum number specified by user. Therefore, depending on the complexity of the problem, the geometric representation of the route can be simplified by the algorithm, by disabling some PIs (or curves), that is, considering that the corresponding section of the route is straight. The activation key is a binary value 0 (indicating that the PI is inactive and its parameters should be ignored) or 1 (indicating that the PI generates a curve).

The codification of each cell is then comprised by a chromosome with N sets of genes, each set associated with a PI. Then the full codification of a chromosome can then be written as Figure 3:

$$A_1\delta_1\alpha_1R_1A_2\delta_2\alpha_2R_2\ldots A_N\delta_N\alpha_NR_N$$

Figure 3. The chromosome codification

Where $A_1\delta_1\alpha_1R_1$ are parameters corresponding to the first PI (or curve), and N is the maximum number of PIs. Therefore, along the evolution process, the optimization algorithm will define candidate solutions by selecting the number of active points (PIs), the values for the polar coordinates that define their position, and the associated curvature radius.

Figure 4 illustrates an example of activation/deactivation process for a route with three PIs.

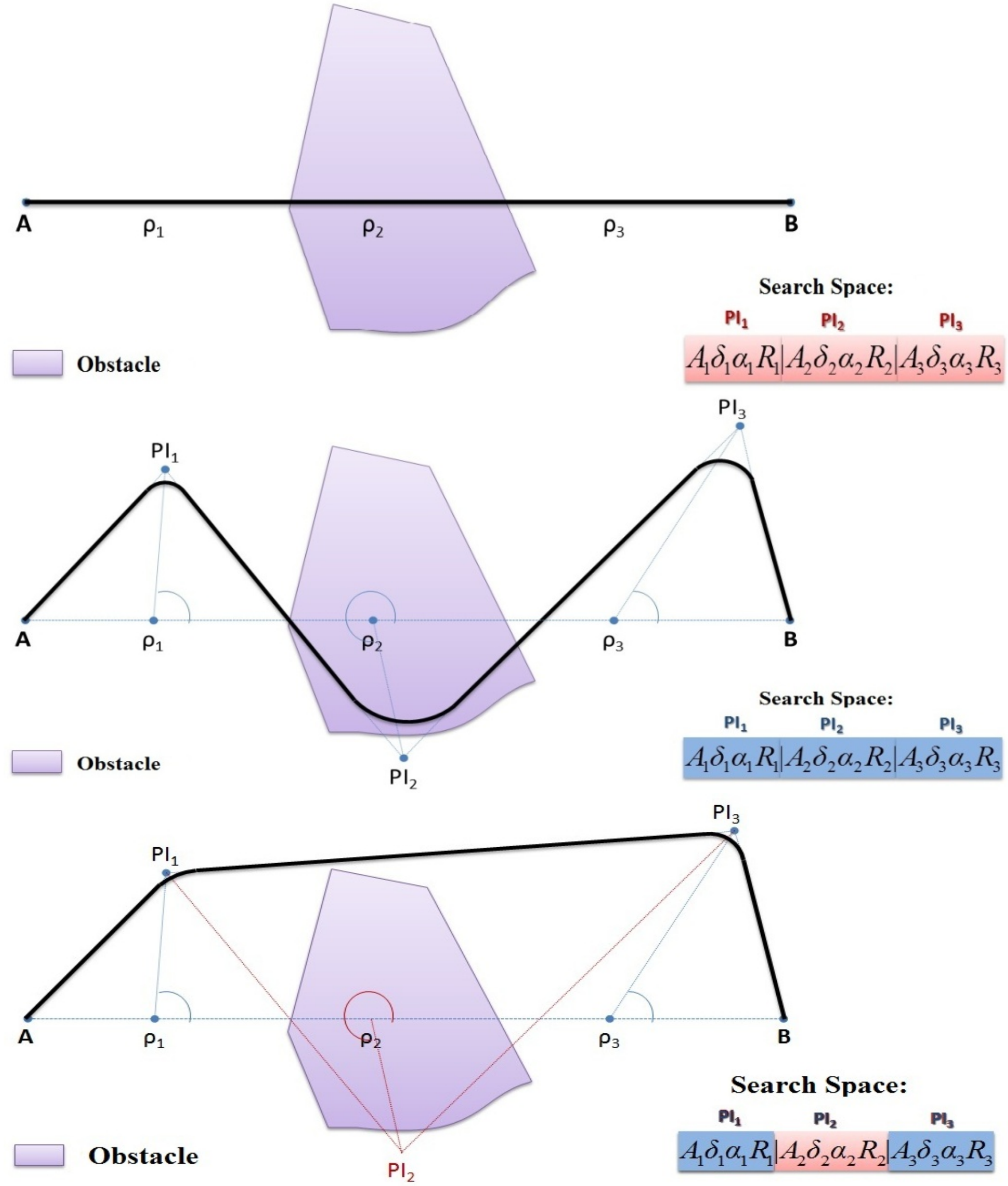


Figure 4. An example of PIs activation/deactivation process

The route codification, which is represented by parameters of each PI, is described for each solution. In this example the parameters values (R, δ and α) for each PIs are the same for all three represented routes. In the first one, all PIs are deactivated (A_1, A_2 and A_3 equals zero), thus, the information regarding its parameters are ignored. It can be seen that despite having the smallest possible length, this solution (or route) could not avoid the obstacle. In the second solution all PIs are active (A_1, A_2 and A_3 equals one), so the parameters of each PI (or curve) are considered. This is also not a good solution because it cannot avoid the obstacle. In the third solution, the route was able to avoid the obstacle only by disabling the information of the second PI.

It is important to emphasize that each PI adds four new parameters to the route representation, increasing the complexity of the problem and hampering the search process performed by NIAs. However, a small amount of PIs may not be sufficient for a route to avoid all constraints present in a given seabed. For this reason it is extremely important to correctly dimension the amount of PIs (or curves) that will represent a route in a particular project.

4 Objective function

After describing the codification in an optimization tool, it is necessary to define an objective function to evaluate the suitability of each candidate route. The objective function should reflect the route quality, taking into account relevant aspects concerning the choice of the best route.

One of the most important factors involved in a route evaluation is its total length. Obviously, the pipeline length should be minimized, in order to reduce material and intervention costs. Therefore, the objective function proposed in this work is defined simply as follows:

$$f(x) = \frac{LenghtRoute}{LenghtAB} \tag{1}$$

Where *LenghtRoute* represents the length of a given solution and *LenghtAB* represents the straight line length that connect the endpoints A and B.

The main goal of an optimization algorithm is to minimize the *f(x)* value, thus it is evident that if all constraints were ignored the best route would be trivially defined by the straight line connecting points A and B.

5 Constraints

Of course, besides the length there are several other factors that influence the cost and safety of a pipeline route, including physical, geometric and structural constraints, related for instance to geographical/topographical issues associated with the sea bottom bathymetry, interference with obstacles, slope and so on.

Thus, the computational tool must consider these complex limitations present in an actual submarine pipeline project. Penalty functions are responsible to quantify the violation degree of each constraint. Table 1 shows constraints summary considered in the optimization tool, their penalty function type, when each constraint is active and each penalty function limits. The description of each constraint may be found in [8].

Constraints	Penalty Function Type	Penalized when:	Limits
Self Crossing	Exponential	Number of intersections > 0	[0 , ∞)
Interference with Obstacles	Exponential	Number of intersections > 0	[0 , ∞)
Minimum length between curves	Linear	Measured length < 500 m	[1 , 0]
Minimum Length of Straight Sections	Linear	Measured length < 500 m	[1 , 0]
Minimum Radius of Curvature	Linear	Measured Readius < 1,681 m	[1 , 0]
Longitudinal Declivity	Exponential	Measured declivity > 3^0	[0 , ∞)
Stability Criteria	Linear	Measured safety factor < 1,1	[1,1 , 0]
Attractor	Linear	Distance to attractor > attractor radius	[0 , ∞)

Table 1. Constraints and their Penalty Functions

5.1 Working with constraints

NIAs were designed to deal with unconstrained search spaces. This was the main motivation to add constrained-handling techniques aiming to guide the search to feasible regions.

Descriptions and examples of several constraint-handling techniques which have had a relatively impact in this research area can be found in [9]. Some of them are known as: i) feasibility rules, ii) stochastic ranking; iii) ε-constrained method; iv) penalty functions; among others.

In this work penalty functions method, which provided very competitive results [9], is used. In this methodology, a constrained optimization problem is transformed into an unconstrained problem by introducing penalty terms to objective function, whenever a given constraint is violated. Thus, the suitability or fitness of each solution generated by the NIA has adopted penalty functions, whose general formula can be written as Equation (2):

$$F(x) = f(x) + \sum k_i P_i(x) \quad \text{where} \quad \begin{array}{ll} P(x) = 0 & \text{if feasible} \\ P(x) > 0 & \text{otherwise} \end{array} \tag{2}$$

Where $P_i(x)$ is the penalty function of each constraint and k_i is its positive constant called “penalty factor”.

In Equation (2), the penalty value is added to infeasible solution fitness because low fitness values are preferred as expected in a minimization problem. As can be noted, the aim is to increase the infeasible solutions fitness in other to favour the selection of feasible one.

Even though their implementation is quite simple, penalty-based approaches require a careful fine-tuning of each factor value (k_i) in order to determine the severity degree to be applied to its penalty function. The penalty factor is intended to amplify its constraint penalty value in detriment of others.

The static penalty approach, which factor values (k_i) remain fixed during all the algorithm evolution, is actually used in the computational tool. Thus, the penalty factors tuning is part of a design problem and may require a high computational cost caused by "trial and error" process. By properly tuning, the relative importance of the different factors should be correct. The main drawback of keeping penalty factor values fixed is the generalization of such type of approach, i.e., the values that may be suitable for one scenario are normally unsuitable for another one.

In this paper another penalty-based approach, where penalty factors are treated dynamically, is described in next section and will be compared to static method in the case study.

5.2 Adaptive penalty method

In this work a method without any type of user defined penalty factors, known as APM [7], was implemented in the computational tool to assist clonalg handling constraints. This adaptive scheme uses information from the population, such as the objective function average and the violation level of each constraint during the evolution, in order to define different penalties factors (k_i) for different constraints. The idea is that the penalty factors values should be distributed in a way that those constraints which are more difficult to be satisfied should have a relatively higher factor. One indication of such difficulty is the number of elements violating a given constraint and the amount of violation. In order to achieve the desired distribution, the j_{th} coefficient is made proportional to the average of the violation of the j_{th} constraint by the elements of the current population. In this method, the fitness function value for each candidate solution is obtained by means of Equation (3):

$$F(x) = \begin{cases} f(x) & \text{if feasible} \\ \overline{f}(x) + \sum_{j=1}^{m} k_j v_j(x) & \text{otherwise} \end{cases} \tag{3}$$

Where,

$$\overline{f}(x) = \begin{cases} f(x) & \text{if } f(x) > \langle f(x) \rangle \\ \langle f(x) \rangle & \text{otherwise} \end{cases} \tag{4}$$

The penalty factor (k_j) is defined at each generation by

$$k_j = |\langle f(x)\rangle| \frac{\langle v_j(x)\rangle}{\sum_{l=1}^{m}[\langle v_l(x)\rangle]^2} \tag{5}$$

Where $\langle f(x)\rangle$ is the average of the objective function of all solutions in the current iteration, $v_j(x)$ is the violation value j and k_j is the penalty factor that is computed adaptively according to Equation (5).

Figure 5 illustrates how it is obtained the function $\overline{f}(x)$ used in Equation (3) to each unfeasible solution. The goal of this example is to minimize the fitness value. In this figure feasible as well as infeasible solutions are shown. It can be seen that the points (x = 1, 2, 3, 4, 5 or 6) highlighted in the graph represent infeasible solutions because they are not in a feasible region of search space thus become necessary to calculate the value of $\overline{f}(x)$ for each solution according to Equation (4).

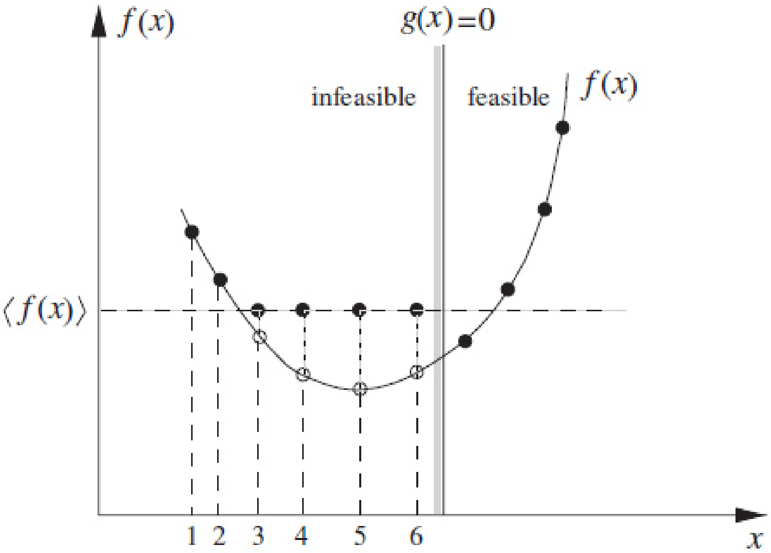


Figure 5. Example of APM.

Among these six infeasible solutions, the individuals #3, #4, #5 and #6 have their objective function values (represented by opened circles), less than the population average objective function and, according to the proposed method, have $\overline{f}(x)$ given by $\langle f(x)\rangle$. The solutions #1 and #2 have objective function values which are worst than the population average objective function and thus have $\overline{f}(x)=f(x)$

This method treats the penalty of equality and inequality constraints, does not demand explicit knowledge of the constraints, is free of parameters to be defined by the user and computational implementation is easy.

6 Case study

To assess the relative performance of the two penalty-based approaches discussed above (i.e. static and dynamic penalty), we have conducted an experimental study in which the methodologies are coupled to clonalg and standard GA optimization methodologies for the treatment of the pipeline route constraints.

The scenario of this study is shown in Figure 6. The main goal is to find the best pipeline route that connects points A and B. The minimum distance between these locations, represented by the straight purple line, is approximately 12,474.00 meters. The bathymetry and the seabed obstacles, with different levels of severity (green, yellow and red lines) are included on this figure.

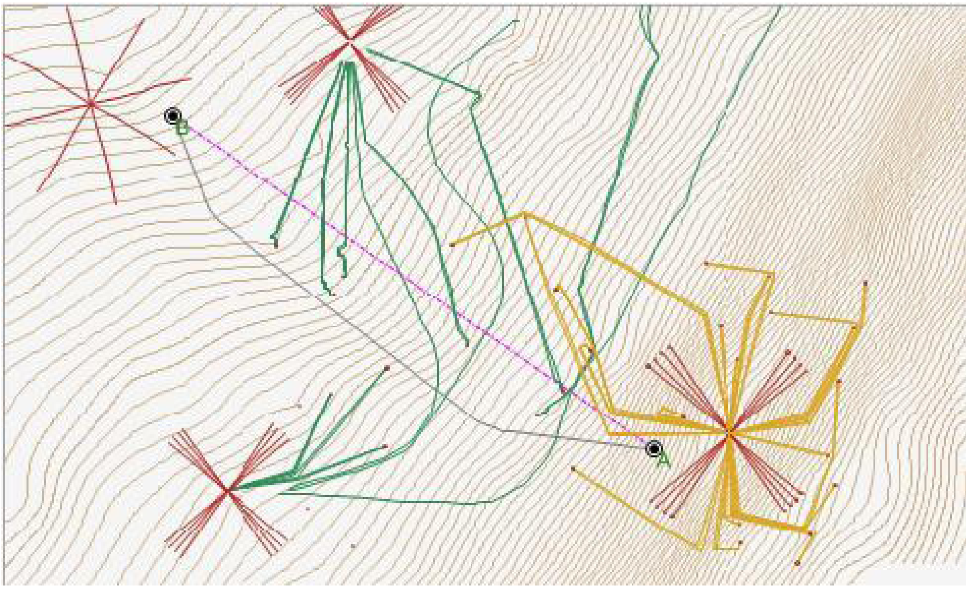


Figure 6: Case study scenario

6.1 Configuration of the experiments

For this experiment, the maximum number of PIs was set to four. Thus, the coding of each solution (or route) generated by the optimization algorithms are represented in Figure 7.

$$A_1\delta_1\alpha_1R_1 \mid A_2\delta_2\alpha_2R_2 \mid A_3\delta_3\alpha_3R_3 \mid A_4\delta_4\alpha_4R_4$$

Figure 7: Solution code to this case study

It is interesting to note that all possible combination values for these twelve parameters define the problem search space that is subject to constraints described in section 3. The limit values for the parameter of each PI are described in Table 2.

Parameter	Limits
Activation (A)	[0 , 1]
Angular (α)	[0^o , 360^o]
Radial (δ)	[0 , 6.237]
Radius (R)	[500 , 5.000]

Table 2: Limit values for the Parameters of each PI

The threshold of activating factor (*A*), which defines the activation/deactivation probability of each PI (or curve) in a route was set to 0.75. Thus, the parameters that represent each PI will be considered when their activation factor value (*A*) is greater than or equal to 0.25.

Table 3 shows the static factors tuned for this scenario, employed on the test with static penalty approach. This tuning was obtained after several tests and analyses performed by experienced specialists in this type of project.

Constraint	Penalty factor value (k)
Self Crossing	10
Interference with Obstacles	5
Minimum length between curves	1
Minimum Length of Straight Sections	1
Minimum Radius of Curvature	1
Longitudinal Declivity	2
Stability	1
Attractor	1

Table 3: Penalty factor values for the static penalty approach

The experiment employed the clonalg and GA algorithms with real representation. Other parameter values to the clonalg, such as population size, number of clones, somatic hypermutation and the value of rho, are show in Table 4.

Parameter	Value
Number of antibodies	5
Number of clones	5
Somatic hypermutation	2 variáveis
Rho	14

Table 4: Clonalg settings

The GA was set with one individual elitism and non-uniform mutation. Other parameter values to this algorithm, such as population size and crossover rate, are show in Table 5.

Parameter	Value
Number of Individuals	50
Crossover Rate	0.6
Non-Uniform Mutation (b)	5

Table 5. Genetic algorithm settings

6.2 Results

In order to compare the performance/robustness of the two penalty-based approaches, this experiment was evaluated by thirty independent runs. All runs were terminated after three hundred (300) generations.

The results obtained by the APM and the static penalty approaches are presented in Table 6 to 9 respectively, in terms of the best, worst, mean and standard deviation for length and two penalties (obstacle and slope) that kept activated at the end of all runs. All other constraints present in studied scenario could be avoided. The best values obtained in all runs are highlighted in bold.

Parameters	Best	Worst	Mean	Standard Deviation
Route lenght (m)	**12.953,29**	**13.417,33**	**13.169,95**	**98,27**
Interf. with Obstacles	**0,82**	**8,02**	**1,34**	**1,50**
Longitudinal Declivity	0,27	**0,69**	**0,46**	**0,10**

Table 6. Results of clonalg with APM

Parameters	Best	Worst	Mean	Standard Deviation
Route lenght (m)	13.150,98	16.837,45	13.747,52	759,25
Interf. with Obstacles	**0,82**	8,10	1,93	2,09
Longitudinal Declivity	**0,23**	1,17	0,60	0,21

Table 7. Results of clonalg with static penalty

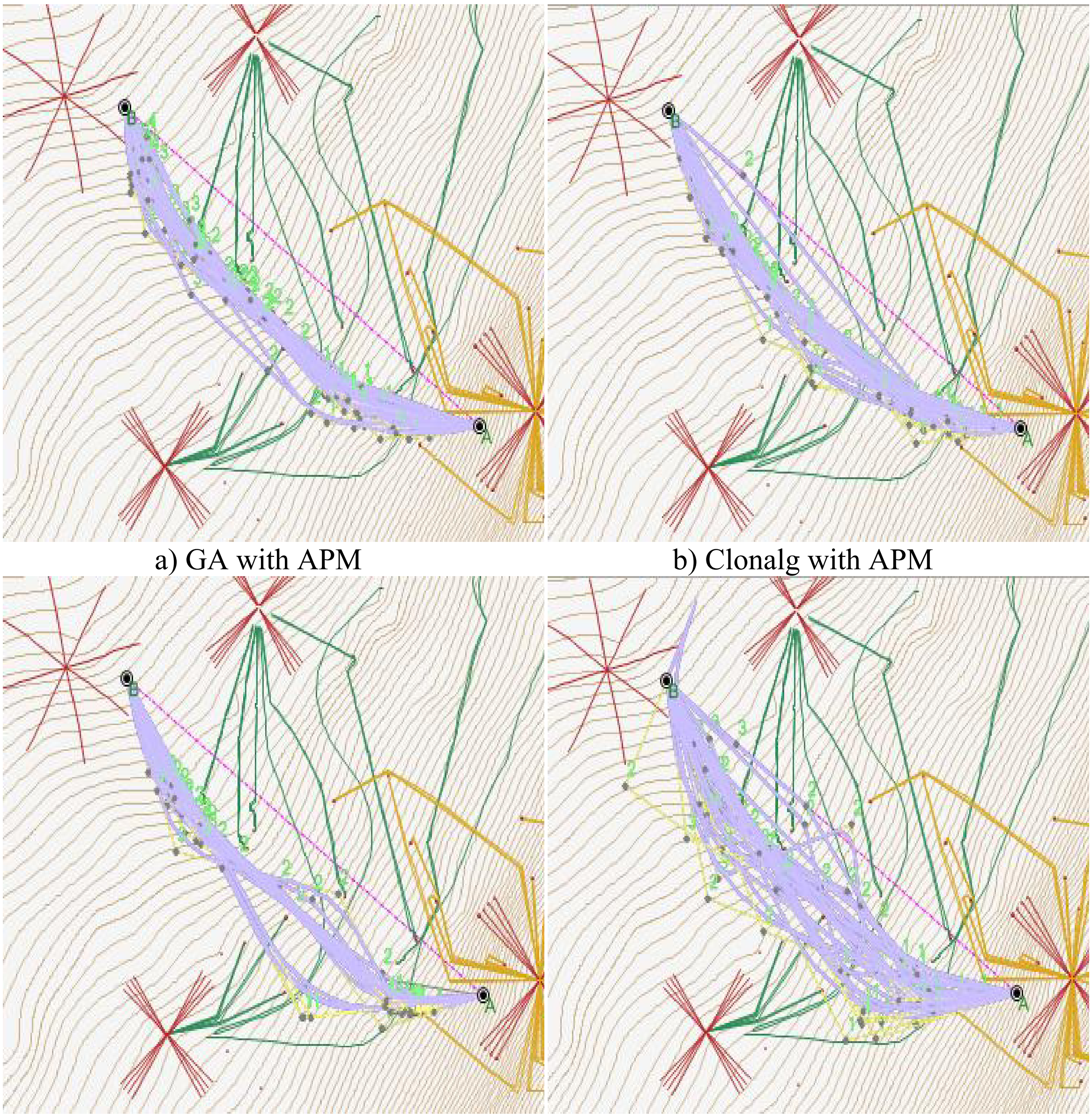
Parameters	Best	Worst	Mean	Standard Deviation
Route lenght (m)	12965,18	13912,71	13277,55	204,53
Interf. with Obstacles	**0,82**	1,72	**0,91**	0,22
Longitudinal Declivity	0,24	0,68	**0,46**	0,11

Table 8. Results of GA with APM

Parameters	Best	Worst	Mean	Standard Deviation
Route lenght (m)	13224,17	13984,34	13434,11	229,12
Interf. with Obstacles	0,82	1,72	0,94	0,31
Longitudinal Declivity	0,29	0,75	0,48	0,07

Table 9. Results of GA with static penalty

Figure 8 shows a comparison between the thirty solutions (or routes) obtained by each penalty-based approach applied to each optimization algorithm. Analyzing these results it is evident that the APM obtained more robust results, when compared to static penalty approach, presenting shorter routes and lower mean and standard deviation values for the length and each activated constraint (slope and obstacle).



a) GA with APM

b) Clonalg with APM

c) GA with static penalty

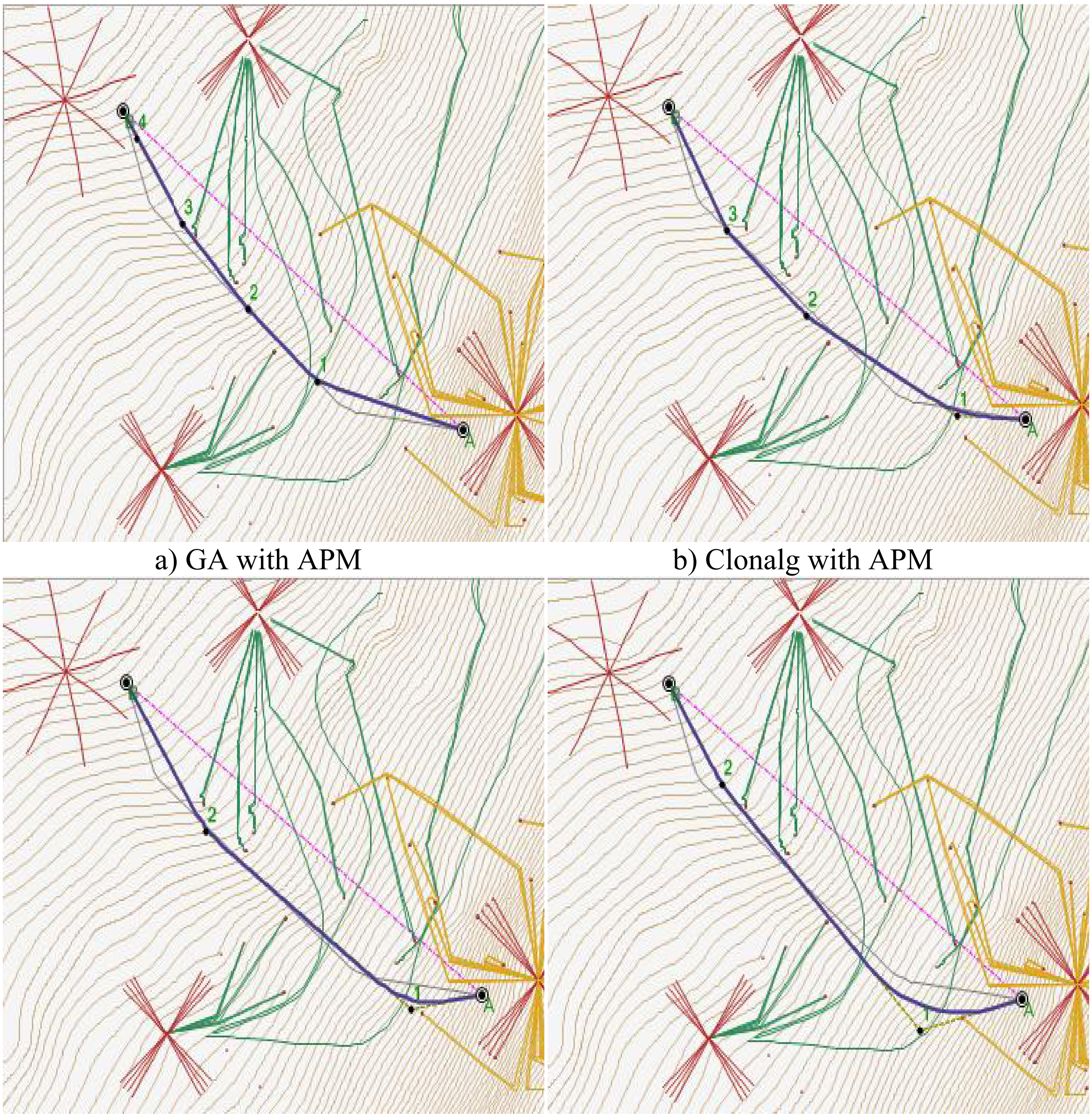
d) Clonalg with static penalty

Figure 8. Graphical representation of the thirty solutions

Table 10 shows the characteristics of the "optimal route" obtained by each approach. These routes are shown in Figure 9.

Algorithm	Penalty Approach	Best Fitness	Route lenght (m)	Interf. with Obstacles	Slope
Clonalg	APM	**14,28**	13101,91	**0,82**	0,36
	Estática	14,62	13430,45	0,89	0,30
AG	APM	14,32	**12965,18**	0,88	0,47
	Estática	14,52	13224,17	0,82	0,48

Table 10: Best result by each approach



a) GA with APM

b) Clonalg with APM

c) GA with Static Penalty

d) Clonalg with Static Penalty

Figure 9. Representation of best pipeline route of each algorithm

During the tests, it could be observed that the use of APM to handling constraints took longer time to converge to a satisfactory solution. This is because the static penalty method had "ideal" tuning values for a set of penalty factors (k) that were already adjusted to this study scenario, facilitating the convergence of the algorithm. In spite of this, the use of APM for handling constraints generates shorter routes than the static method, noting that a small reduction in a pipeline route length may represent a significant cost savings in actual offshore engineering design.

Figure 10 shows the value evolution of each penalty factor generated by the APM associated to the clonalg in the treatment of constraints. It is seen that the constraints that cannot be avoided (obstacle and slope) have higher penalty factor values, because the APM distributes the values of ki, so that the restrictions more difficult to be avoided are severely penalized.

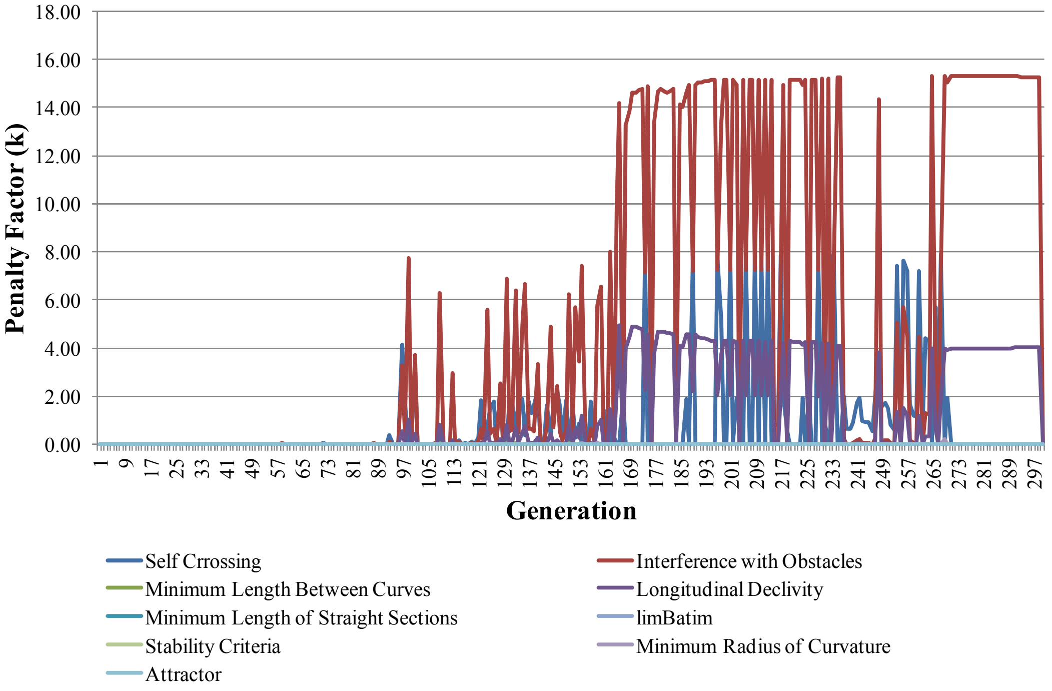


Figure 10. Factor penalty value (k) evolution for each constraint treated by APM

6 Final remarks

This work compared the performance of an adaptive penalty method, known as APM, with the static penalty method currently used in an optimization tool for submarine pipeline routes. Both methodologies are associated to artificial immune system/clonalg and to standard genetic algorithm optimization methodologies.

The optimization algorithms using the APM were shown to be more efficient and robust to provide the best solutions in terms of the higher mean values and lower standard deviations of fitness of the routes. Based on these results, it can be stated that APM is able to assist the optimization algorithms in actual pipeline projects, refining the analysis with respect to avoiding obstacles, seeking and finding lower slopes with shorter routes. Thus the methodology used in this study showed consistent results, satisfactory and with a reduced computational time.

As a result, it is expected that the application of the route optimization tool may reduce the design time needed to assess an optimal pipeline route, while reducing computational overheads and providing more accurate results (avoiding mistakes with route interpretation), ultimately minimizing costs with respect to submarine pipeline design and installation.

Future studies will assess the performance of other constraint-handling methodologies, such as stochastic ranking and □-constrained, applied to NIAs in optimization of offshore projects.

Acknowledgements

The first author would like to acknowledge the brazilian CAPES agency for the financial support.

References

[1] Recommended Practice DNV-RP-F101, Submarine Pipeline Systems, Det Norske Veritas, 2007.

[2] Castro, L.N.; ZUBEN, F.J.V.; Learning and Optimization using the Clonal Selection Principle, Special Issue on Artificial Immune Systems, IEEE Transactions on Evolutionary Computation, 6, pp. 239-252, 2002.

[3] Albrecht, C. H., Algoritmos Evolutivos Aplicados À Síntese e Otimização de Sistemas de Ancoragem, Tese de Doutorado, Programa de Engenharia Oceânica, COPPE/UFRJ, 2005.

[4] Pina, A,A; Metodologias de Análise, Síntese e Otimização de Sistemas para Produção de Petróleo Offshore Através de Metamodelos e Enxame de Partículas, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2010.

[5] Vieira, I. N., LIMA, B.S.L.P., JACOB, B. P, “Optimization of Steel Catenary Risers for Offshore Oil Production Using Artificial Immune System”. In: International Conference on Artificial Immune Systems, Lecture Notes in Computer Science. Berlin: Springer-Verlag, v. 5132. pp. 254 – 265, 2008, Phuket, Thailand.

[6] Vieira I.N., Albrecht C.H., de Lima B.S.L.P., Jacob B.P., Rocha D.M., Cardoso C.O. Towards a Computational Tool for the Synthesis and Optimization of Submarine Pipeline Routes. Procs. of the Twentieth International Offshore and Polar Engineering Conference - ISOPE 2010, Beijing, China.

[7] Barbosa, H. J. C.; Lemonge, A. C. C, A New Adaptive Penalty Scheme for Genetic Algorithms. In Information Sciences, 2003.

[8] Lima Jr, M.H.A., Baioco, J.S., de Lima BSLP, Albrecht, C.H., Jacob, B.P., Rocha D.M, Cardoso, C.O. Synthesis and Optimization of Submarine Pipeline Routes considering On-Bottom Stability Criteria. In: Proceedings of the 30th International Conference on Ocean, Offshore and Arctic Engineering CD-ROM, paper OMAE2011-49373, Rotterdam, The Netherlands.

[9] Montes E. M., Coello C. A. C., 2011. Constraint-handling in nature-inspired numerical optimization: Past, present and future. Survey Paper. In Elsevier, 2011.