XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTIONAL METHODS IN ENGENEERING

Optimal Support Placement for VIV Mitigation in Subsea Rigid Pipelines

Matheus A. Miranda, Eduardo N. Lages, Adeildo S. Ramos Jr

Laboratory of Scientific Computing and Visualization, Center of Technology, Federal University of Alagoas
Av. Lourival Melo Mota s/n, Cidade Universitária, 57072-900, Maceió/Alagoas, Brazil
matheus.miranda@ctec.ufal.br, enl@lccv.ufal.br, adramos@lccv.ufal.br

Abstract. This study presents an optimization approach for determining the placement of supports to mitigate vortex-induced vibrations (VIV) in free spans of subsea rigid pipelines. Free spans caused by seabed irregularities can critically compromise the pipeline's structural integrity. To address this challenge, the proposed methodology integrates numerical modeling based on the Finite Element Method (FEM), following the recommendations outlined in DNV-RP-F105, with discrete optimization techniques, namely, a metaheuristic genetic algorithm and a deterministic branch-and-bound algorithm. The objective function minimizes support installation costs by assigning specific penalties to each free span and incorporating constraints related to the pipeline's natural frequency. Results show that the framework is efficient, with the branch-and-bound algorithm consistently reaching optimal solutions while avoiding the variability inherent in stochastic approaches.

Keywords: Subsea rigid pipelines, Vortex-induced vibrations, Structural optimization, Branch-and-bound algorithm

1 Introduction

Subsea rigid pipelines are critical infrastructure for energy transport, but their structural integrity can be compromised by free spans resulting from seabed irregularities. These unsupported sections are highly susceptible to vortex-induced vibrations (VIV), a dynamic phenomenon driven by ocean currents that can lead to significant fatigue damage and potential pipeline failure, as described by Blevins [1], Santos [2], Lopes [3] and Sumer and Fredsøe [4]. Mitigating VIV in free spans is therefore a paramount concern in offshore engineering and requires effective and cost-efficient solutions.

The challenge lies in determining the optimal placement of supports to suppress VIV. While numerical modeling, often based on the Finite Element Method (FEM) and industry guidelines like DNV-RP-F105 (see Det Norske Veritas [5]), provides robust tools for analyzing pipeline behavior, the strategic positioning of supports constitutes a complex discrete optimization problem. The objective is to minimize overall support installation costs while ensuring that the pipeline's natural frequencies are sufficiently separated from critical VIV excitation ranges, thereby maintaining structural integrity.

This study aims to develop and evaluate a novel optimization framework for determining the optimal placement of supports to mitigate VIV in subsea rigid pipelines. The primary objective is to minimize support installation costs by assigning specific penalties to non-compliant free spans and incorporating critical constraints related to the pipeline's natural frequencies. This framework seeks to provide a robust and efficient methodology for identifying cost-effective and structurally sound support configurations.

2 Optimization Problem Formulation

The numerical model adopted in this work follows the formulation previously presented by Miranda and Lages [6] and is enhanced through the implementation of sparse matrices. The use of sparse matrix structures, specifically using the Python libraries NumPy and SciPy (see Jennings [7], Harris et al. [8] and Virtanen et al. [9]), enables efficient storage and manipulation of large system matrices, which is essential for the analysis of extensive subsea pipelines. This approach significantly reduces memory consumption and computational time, making the model more suitable for iterative optimization procedures.

The optimal support placement problem for VIV mitigation in subsea rigid pipelines can be formulated as a

discrete optimization problem. The objective is to determine the configuration of supports along the pipeline that minimizes the total installation cost, while ensuring that all free spans comply with design constraints related to natural frequencies.

The design variables adopted in this study correspond to the presence or absence of transverse supports at potential locations n, defined by the discretized nodes within the modeled free spans. These variables are assembled into a vector $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_n)$, where each component is binary: $x_i = 0$ denotes the absence of a support at location i, while $x_i = 1$ indicates its presence, as shown in Fig. 1.

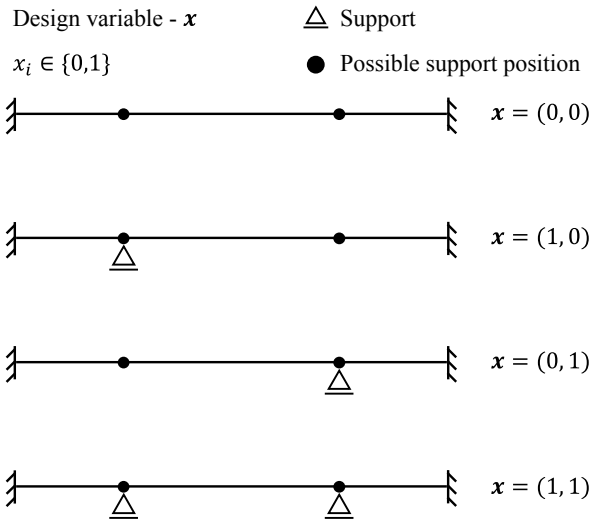


Figure 1. Schematic representation of a design variable $\mathbf{x}$

In extensive scenarios with multiple free spans, factors such as water depth, gap, soil type, and the need for specialized equipment and personnel for support installation significantly influence the overall cost. These factors play a crucial role in the decision-making process regarding interventions at specific spans. Consequently, each potential support position is associated with a cost penalty factor, forming a fixed-cost vector $\mathbf{c} = (c_1,\ c_2,\ \ldots,\ c_n)$.

The optimization problem is defined in eq. (1), which seeks the minimum-cost support configuration while enforcing a lower bound on the first vertical natural frequency:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) = \mathbf{c} \cdot \mathbf{x} \\ \text{subject to} \quad & g(\mathbf{x}) = f_{\text{target}} - f_1(\mathbf{x}) \leq 0 \\ & x_i \in \{0, 1\}; i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $f_1(\mathbf{x})$ is the first natural frequency in the vertical direction obtained from the developed numerical model, and f_{target} is the minimum desired frequency for the system.

As defined in eq. (1), the first natural frequency $f_1(\mathbf{x})$ is evaluated using a numerical model that follows the formulation previously presented by Miranda and Lages [6]. In the present work, this model is refined through the consistent use of sparse matrix representations, which is implemented via the Python ecosystem, particularly NumPy and SciPy (see Jennings [7], Harris et al. [8], Virtanen et al. [9]), and enables memory-efficient storage and fast linear-algebra operations on large system matrices. These improvements substantially reduce computational cost, ensuring that repeated evaluations of $f_1(\mathbf{x})$ remain tractable within the iterative optimization procedure.

3 Solution strategies

We employ two complementary algorithms to solve the binary placement problem in eq. (1): a genetic algorithm (GA) and a branch-and-bound (B&B) method. In both cases, the feasibility is assessed by computing the first natural frequency with the numerical model, in which sparse-matrix linear algebra keeps repeated evaluations efficient. Since the runtime of each structural simulation is nearly constant, the overall computational effort is directly proportional to the number of evaluations. Therefore, reporting evaluations provides an adequate measure

of the relative efficiency of both methods. Throughout the search, every feasible solution encountered is stored; the reported optimum is the feasible solution with minimum cost, and, in the event of ties, the one whose first natural frequency deviates the most from the target (i.e., has the largest safety margin).

3.1 Genetic algorithm

We use the Platypus optimization library (see Hadka [10]), which natively supports binary decision variables and nonlinear constraints, matching the structure of eq. (1). Given a user-defined population size, an initial set of binary candidate solutions is sampled uniformly over the feasible domain of the variables. At each iteration, fitter individuals, those with lower cost that satisfy the frequency constraint, are selected to produce offspring via crossover, which recombines parental bit patterns, and mutation, which flips a small fraction of bits to preserve population diversity. The number of model evaluations scales primarily with the population size and the maximum number of generations; these two parameters are, therefore, the main levers to trade off solution quality against runtime.

3.2 Branch-and-bound method

We also implement a custom branch-and-bound procedure inspired by the monotonic, non-decreasing properties of the objective function and the pruning rationale of Huang and Arora [11]. Because the objective is a non-decreasing linear function of the number and location of supports, once a feasible incumbent is found, any partial assignment whose current cost already exceeds that of the incumbent can be safely pruned. This rule, combined with early infeasibility detection from frequency checks, substantially reduces the search space, as shown in Fig. 2. Following Tseng et al. [12], the exploration proceeds in a breadth-first manner, which tends to identify good incumbents early and strengthens subsequent pruning. In our implementation, we also memoize objective/feasibility/frequency evaluations to avoid repeated simulations, filter duplicate assignments with a seen set, and break ties among equal-cost solutions by keeping the one with the highest first natural frequency (largest safety margin).

4 Case study

This case study considers a pipeline segment with three free spans of $35\,\mathrm{m}$, $15\,\mathrm{m}$ and $27\,\mathrm{m}$, arranged as shown in Fig. 3. Candidate support locations are defined at the abscissas $58.75\,\mathrm{m}$, $67.5\,\mathrm{m}$, $76.25\,\mathrm{m}$, $100\,\mathrm{m}$, $105\,\mathrm{m}$, $134\,\mathrm{m}$ and $143\,\mathrm{m}$. A complete summary of the data is provided in Table 1.

Table 1. Parameters adopted for the case study

Parameter	Value	Unit
Steel outer diameter	0.27305	m
Steel wall thickness	0.0127	m
Young's modulus	206.0	GPa
Poisson's ratio	0.29	-
Steel density	7850.0	$\mathrm{kg/m^3}$
Seawater density	1025.0	$\mathrm{kg/m^3}$
Soil density	1500.0	$\mathrm{kg/m^3}$
Internal content density	200.0	$\mathrm{kg/m^3}$
Axial inertia coefficient	0.1	-
Lateral inertia coefficient	1.0	-
Axial soil stiffness per unit length	160.0	kN/m/m
Lateral soil stiffness per unit length	160.0	kN/m/m

Note that each free span exhibits a distinct gap; consequently, the installation cost for a support at a given can-

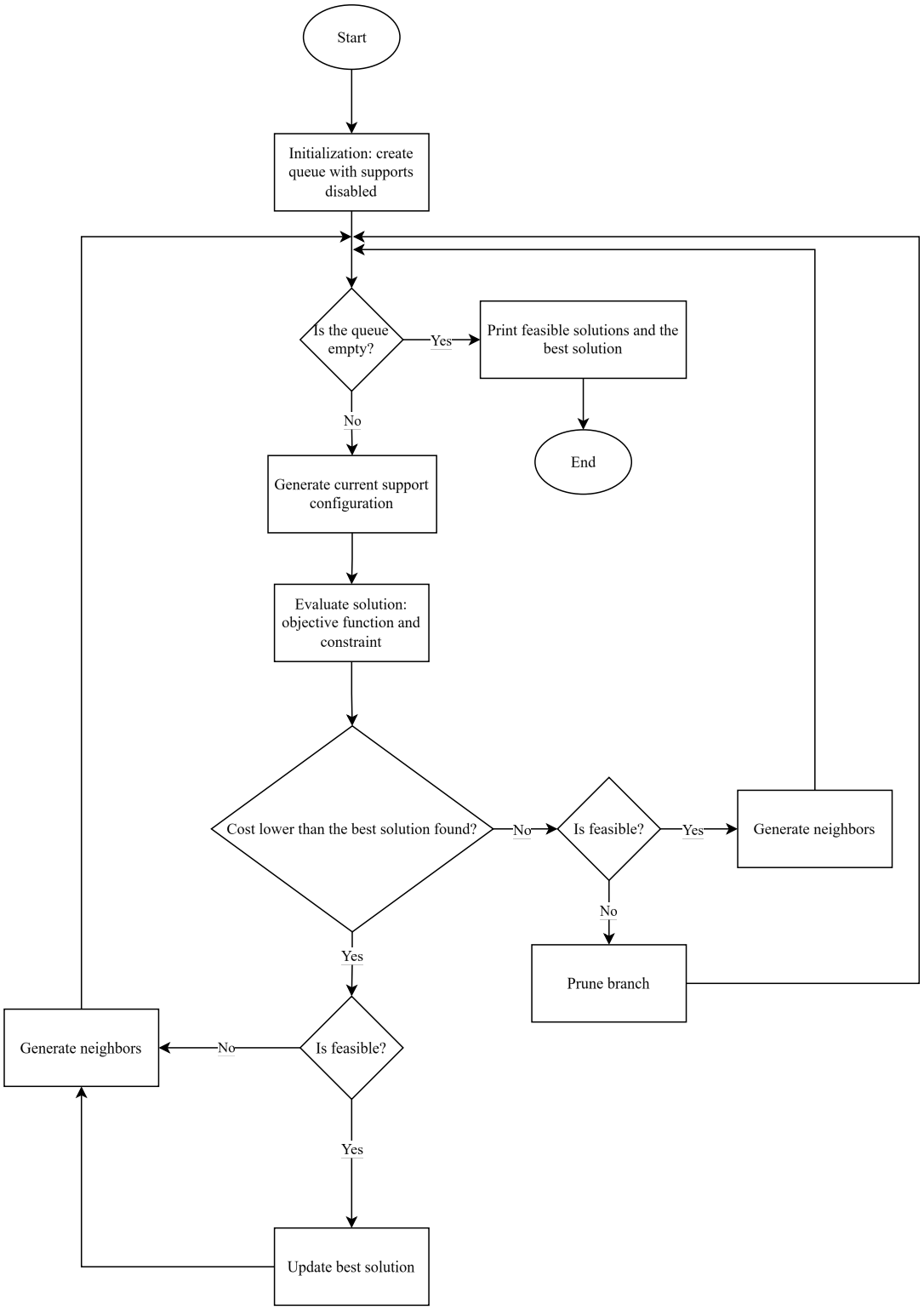


Figure 2. Branch-and-bound flowchart for the support configuration optimization problem

didate position is taken to be proportional to the local gap. This results in the fixed-cost vector $\mathbf{c} = (8,\ 8,\ 8,\ 2,\ 2,\ 4,\ 4)$. With seven candidates, $2^7 = 128$ possible support configurations. Because costs differ across spans, solutions with more supports may still be optimal, a feature that is naturally handled by the cost-weighted objective in eq. (1) and by the search procedures described previously.

Using a target frequency of $f_{\text{target}} = 1\,\text{Hz}$, the branch-and-bound (B&B) algorithm identifies a feasible solution after only 13 objective evaluations. The search then continues to examine additional promising combina-

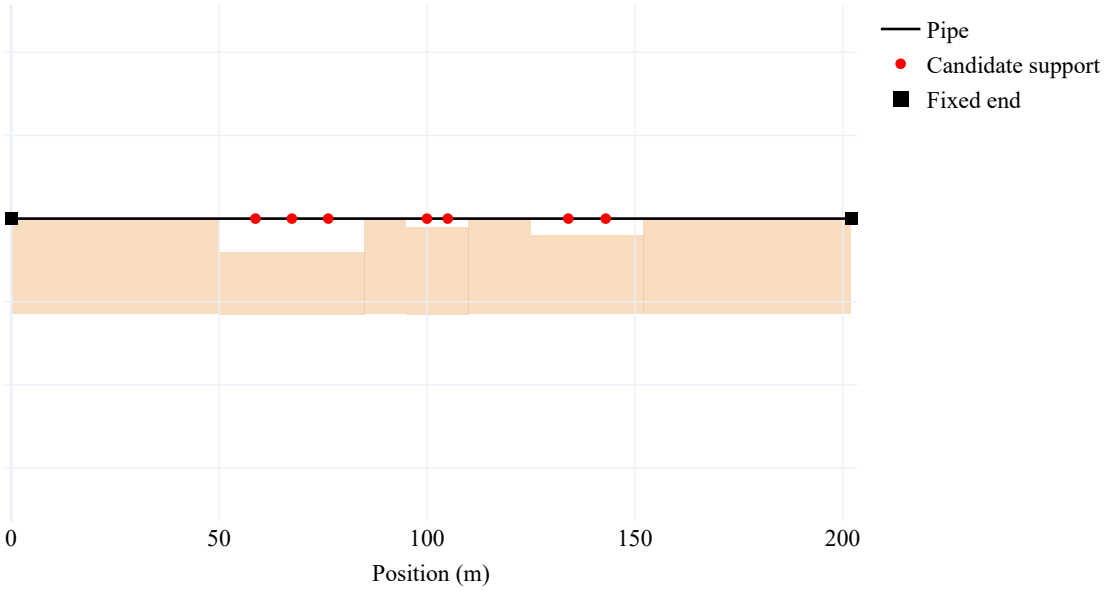


Figure 3. Geometric configuration and candidate support locations for the multi-span scenario

tions—potentially including some with more supports—whenever they can yield the same or lower cost while increasing the first natural frequency, thereby increasing the safety margin relative to the target. Leveraging the monotonic, non-decreasing structure of the cost, the breadth-first exploration prunes any partial assignment whose current cost already exceeds that of the incumbent, which substantially reduces the number of model evaluations.

The optimal solution is certified after 75 objective evaluations (see Fig. 4), representing a 41% reduction relative to an exhaustive search over all 128 combinations. This reduction is achieved without sacrificing optimality and illustrates the practical benefit of the pruning rules for this case.

The genetic algorithm (GA) produced results comparable to those obtained with the branch-and-bound (B&B) method. Because the Platypus implementation controls the computational budget primarily via the population size, we test multiple population sizes and perform 10 independent runs for each size, due to the algorithm's stochastic nature. As a result, the best GA solution does not always match the B&B optimum. Figure 5 summarizes the number of feasible solutions found across population sizes and runs. The results show that larger populations generally increase both the number of feasible solutions obtained and the likelihood of recovering the same optimal configuration identified by B&B (green markers). At small population sizes, feasible solutions are found quickly but most do not coincide with the B&B optimum (red markers). As the population grows, however, the GA not only finds more feasible solutions consistently but also matches the B&B optimum in several runs, highlighting the trade-off between computational budget and solution quality.

When the available objective-function evaluations are fewer than the total number of combinations, there is a non-zero chance that the GA will not sample the globally optimal configuration identified by B&B. Moreover, being a metaheuristic, the GA does not exploit the monotonic structure of the objective, which can lead to unnecessary evaluations; consequently, convergence to the global optimum is not guaranteed, as discussed by Pinto et al. [13]. Nevertheless, if the goal is to obtain feasible configurations quickly, the GA is effective: with a population of 20 individuals, it achieves feasible solutions with about 84% fewer evaluations than exhaustive search in this case.

5 Conclusions

This work formulates optimal support placement for VIV mitigation as a binary cost-minimization problem with a frequency constraint and addresses it with two complementary strategies: a custom branch-and-bound (B&B) procedure and a genetic algorithm (GA) implemented with the Platypus library. In the case study with three free spans and seven candidate positions, B&B identified a feasible configuration in only 13 evaluations and

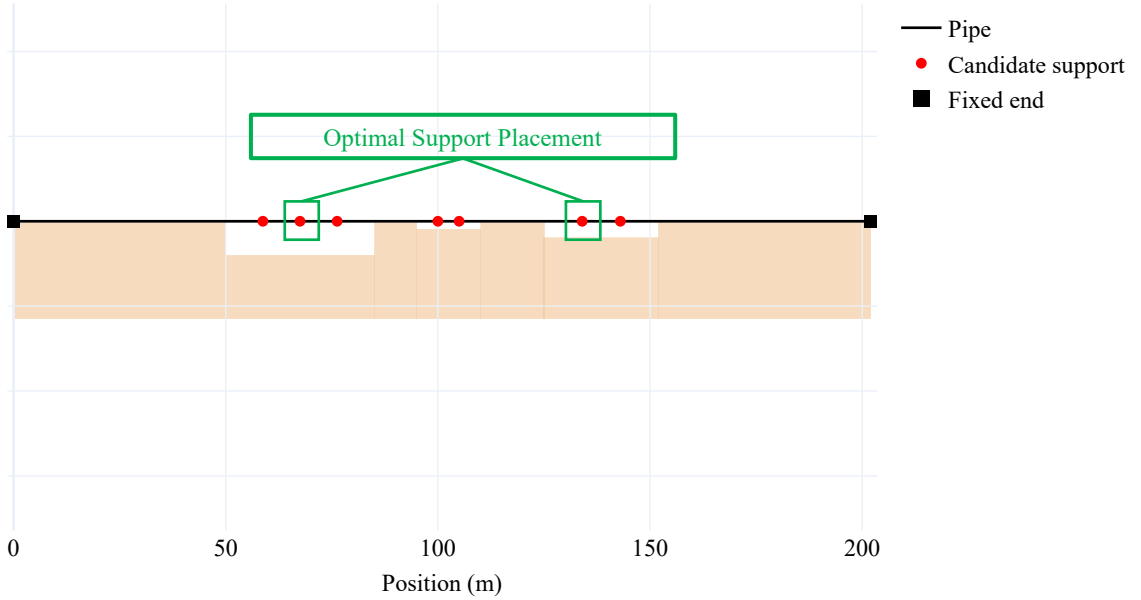


Figure 4. Geometric configuration and optimal support placement for the multi-span case study

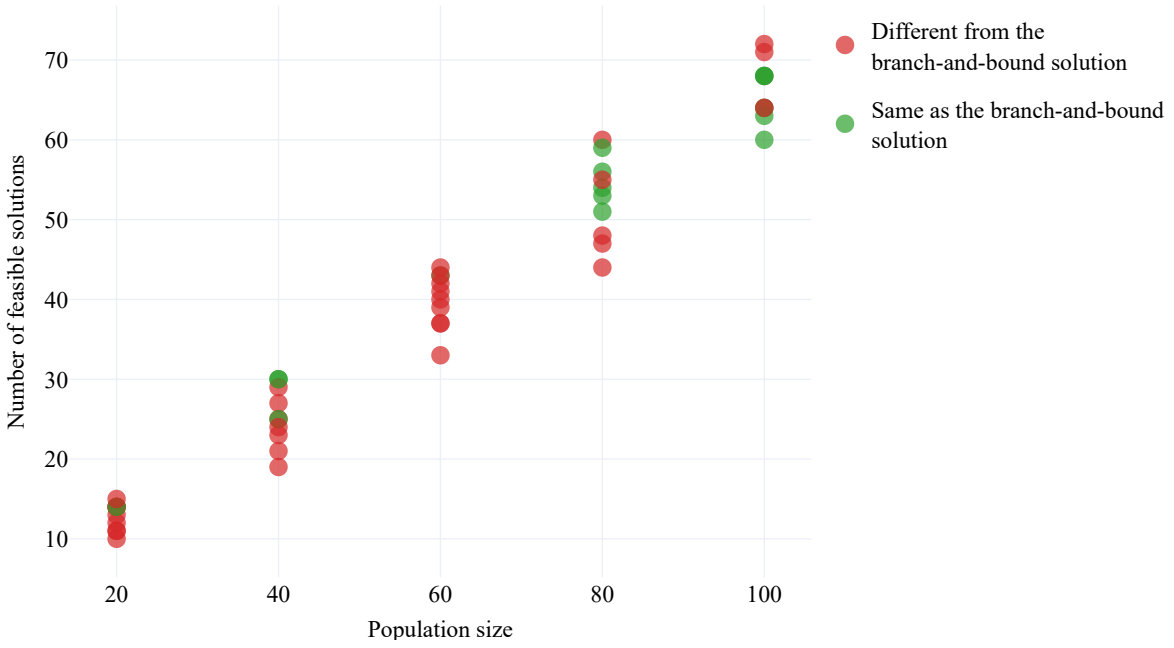


Figure 5. Number of feasible solutions found at each population size; comparison against the branch-and-bound result

certified optimality in 75 evaluations, representing a 41% reduction relative to exhaustive search over 128 combinations. The GA with Platypus, although not guaranteeing optimality, produced comparable solutions across multiple population sizes and runs, often recovering the B&B optimum and in some cases reaching feasible solutions more quickly. With 20 individuals, for instance, the GA achieved feasibility with about 84% fewer evaluations than exhaustive search. Thus, while B&B ensures global optimality with higher evaluation counts, GA provides faster

near-optimal solutions with a lower computational budget.

The proposed workflow is practical for engineering studies: it delivers minimum-cost placements that meet a target frequency and, in the event of ties, favors solutions with greater frequency margins, thereby improving robustness. Furthermore, the workflow can be directly adopted to obtain configurations based on the VIV Avoidance criterion of DNV-RP-F105 (see Det Norske Veritas [5]).

Comparing the two approaches, the branch-and-bound method guarantees global optimality and provides a clear benchmark for assessing other strategies, but this comes at the cost of a larger number of evaluations, especially when certifying optimality. In contrast, the genetic algorithm is faster and more flexible, as it typically requires fewer evaluations to identify feasible solutions; however, its stochastic nature means that it does not always recover the proven optimum. Therefore, the branch-and-bound method is preferable when optimality certification is critical, while the genetic algorithm is more attractive for scenarios where near-optimal solutions obtained with lower computational effort are sufficient for engineering practice.

Limitations include the focus on the first vertical natural frequency and the simplified cost model; future work will extend the optimization to multiple modes and directions, incorporate more detailed installation costs and constraints and couple the frequency target to fatigue-based VIV criteria and uncertainty considerations. We also plan to strengthen the branch-and-bound by exploring alternative search strategies and branching rules tailored to cost–impact.

Acknowledgements. This work was supported by the Federal University of Alagoas (UFAL), Coordination for the Improvement of Higher Education Personnel (CAPES), the National Council for Scientific and Technological Development (CNPq), and Petrobras.

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] R. D. Blevins. *Flow-Induced Vibration*. Krieger, Malabar, Fla, 2 edition, 2001.

[2] C. M. P. M. Santos. *Análise de Estruturas Esbeltas Offshore Sujeitas a Vibrações Induzidas Por Vórtices (VIV)*. Tese (Doutorado em Engenharia Civil), COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

[3] R. K. D. Lopes. *Análise de Estruturas Sujeitas a Vibrações Induzidas Por Vórtices*. Dissertação (Mestrado em Engenharia Civil), COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

[4] B. M. Sumer and J. Fredsøe. *Hydrodynamics around Cylindrical Strucures*. Number v. 26 in Advanced Series on Ocean Engineering. World Scientific Publishing, Singapore ; London, revised ed edition, 2006.

[5] Det Norske Veritas. DNV-RP-F105, 2017.

[6] M. A. Miranda and E. N. Lages. Comparison of Soil Modeling Strategies with Springs in Free-Span Submarine Pipelines. In *Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE)*, 2024.

[7] A. Jennings. A sparse matrix scheme for the computer analysis of structures. *International Journal of Computer Mathematics*, vol. 2, n. 1-4, pp. 1–21, 1968.

[8] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, and others. Array programming with NumPy. *Nature*, vol. 585, n. 7825, pp. 357–362, 2020.

[9] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, and others. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, vol. 17, n. 3, pp. 261–272, 2020.

[10] D. Hadka. Platypus: A Framework for Evolutionary Computing in Python, 2024.

[11] M.-W. Huang and J. S. Arora. OPTIMAL DESIGN WITH DISCRETE VARIABLES: SOME NUMERICAL EXPERIMENTS. *International Journal for Numerical Methods in Engineering*, vol. 40, n. 1, pp. 165–188, 1997.

[12] C. H. Tseng, L. W. Wang, and S. F. Ling. Enhancing Branch-and-Bound Method for Structural Optimization. *Journal of Structural Engineering*, vol. 121, n. 5, pp. 831–837, 1995.

[13] A. R. F. Pinto, N. J. Martarelli, and M. S. Nagano. Algoritmo Genético: Principais Gaps, Trade-offs e Perspectivas para Futuras Pesquisas. *Trends in Computational and Applied Mathematics*, vol. 23, n. 3, pp. 413–438, 2022.