



**Nicholas de Araujo Gonzalez Casaprima**

**Otimização da Configuração Geométrica de  
Risers usando o Método de Otimização  
Bayesiana**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Mecânica, do Departamento de Engenharia Mecânica da PUC-Rio .

Orientador : Prof. Ivan Fabio Mota de Menezes  
Co-orientador: Dr. José Hugo Capella Gaspar Elsas

Rio de Janeiro  
Maio de 2021



**Nicholas de Araujo Gonzalez Casaprima**

**Otimização da Configuração Geométrica de  
Risers usando o Método de Otimização  
Bayesiana**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Mecânica da PUC-Rio . Aprovada pela Comissão Examinadora:

**Prof. Ivan Fabio Mota de Menezes**

Orientador

Departamento de Engenharia Mecânica – PUC-Rio

**Dr. José Hugo Capella Gaspar Elsas**

Tecgraf / PUC-Rio

**Prof. Anderson Pereira**

Departamento de Engenharia Mecânica - PUC-Rio

**Prof. Luís Volnei Sudati Sagrilo**

COPPE / UFRJ

Rio de Janeiro, 07 de Maio de 2021

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

### **Nicholas de Araujo Gonzalez Casaprima**

Graduado em Engenharia Mecânica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2018

#### Ficha Catalográfica

Casaprima, Nicholas de Araujo Gonzalez

Otimização da Configuração Geométrica de Risers usando o Método de Otimização Bayesiana / Nicholas de Araujo Gonzalez Casaprima; orientador: Ivan Fabio Mota de Menezes; co-orientador: José Hugo Capella Gaspar Elsas. – 2021.

80 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2021.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Design ótimo. 3. Otimização Bayesiana. 4. Projeto de risers. 5. Sistema offshore. 6. Engenharia submarina. I. Menezes, Ivan Fabio Mota de. II. Elsas, José Hugo Capella Gaspar. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. IV. Título.

CDD: 621

## Agradecimentos

Agradeço ao meu orientador Ivan F. M. De Menezes por todo o suporte ao longo desse trabalho, mentoria e pelas excelentes aulas que tive no curso de Engenharia Mecânica, sendo um dos grandes professores que conheci na PUC-Rio, e ao meu coorientador José Hugo Capella Gaspar Elsas por todo o suporte, mentoria e ensinamentos ao longo deste trabalho.

Agradeço à PUC-Rio e ao departamento de Engenharia Mecânica pelo curso que me proporcionou uma pós graduação de excelência, me possibilitou conhecer grandes pessoas que hoje fazem parte da minha vida.

Agradeço ao Instituto Tecgraf/PUC-Rio, do qual fiz parte e me trouxe grandes ensinamentos e possibilidade de crescimento em todas as áreas.

Agradeço ao suporte da CAPES, o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

Agradeço muito aos meus pais, Enrique e Raquel, por terem feito de mim a pessoa que sou hoje, pelo exemplo de pessoa e de profissionais que sempre foram pra mim e por todo o apoio que me deram e sei que sempre darão na minha vida.

Agradeço ao meu irmão, Felipe, por sempre ser meu maior parceiro em todos os momentos, independente de onde estivermos, por tudo que me ensinou e que vai me ensinar.

Agradeço a todos os meus amigos, em especial meus irmãos de consideração da Velha Guarda, por todos os momentos compartilhados e risadas ao longo de todos esses anos, meu irmão de consideração Lucas de Giuseppe, pelas histórias compartilhadas e pelo suporte constante e Felipe Furtado, companheiro de engenharia mecânica e da vida.

Agradeço aos meus primos Victor Hugo e Bernardo por terem feito parte dessa trajetória comigo.

## Resumo

Casaprima, Nicholas de Araujo Gonzalez; Menezes, Ivan Fabio Mota de; Elsas, José Hugo Capella Gaspar. **Otimização da Configuração Geométrica de Risers usando o Método de Otimização Bayesiana**. Rio de Janeiro, 2021. 80p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Os *risers* são importantes componentes na produção e exploração de petróleo e derivados. São responsáveis pelo transporte do óleo e gás encontrados no reservatório até a Unidade Estacionária de Produção (UEP) ou pela injeção de gás ou água no reservatório. A crescente demanda por esse produto faz com que a exploração seja feita em regiões com condições cada vez mais adversas. Tipicamente, um projeto deste porte exige um número muito grande de análises numéricas de elementos finitos e exigem uma experiência grande por parte do projetista a fim de obter uma solução viável. Esse desafio leva engenheiros a buscarem ferramentas consistentes e seguras que auxiliem nas etapas iniciais do projeto das configurações de *risers* e que sejam capazes de diminuir o número de análises totais exigidas. Uma dessas ferramentas é a utilização de métodos de otimização para obter de maneira consistente e segura os parâmetros que definem uma configuração. Este trabalho apresenta o método de Otimização Bayesiana, um método baseado em técnicas de aprendizado de máquina capaz de resolver problemas de otimização do tipo caixa-preta de maneira eficiente explorando o uso de aproximações analíticas da função objetivo, que se quer otimizar. O método é aplicado em diferentes estudos de casos visando validá-lo como capaz de resolver problemas de configuração de *riser* de maneira eficiente e consistente. Dentre os problemas aplicados estão diferentes tipos de configurações, diferentes casos realistas, mono-objetivo e multi-objetivo.

## Palavras-chave

Design ótimo; Otimização Bayesiana; Projeto de risers; Sistema offshore; Engenharia submarina.

## Abstract

Casaprima, Nicholas de Araujo Gonzalez; Menezes, Ivan Fabio Mota de (Advisor); Elsas, José Hugo Capella Gaspar (Co-Advisor). **Optimization of Geometric Riser Configurations using the Bayesian Optimization Method**. Rio de Janeiro, 2021. 80p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Risers are an important component in the oil's production and exploration field. They are responsible for the oil and gas transportation from the reservoir to the floating unit or injection of gas or water into the reservoir. The increasing the demand for this product has lead projects to explore to areas in which conditions are harsher. Typically, such a large project demands a large number of numerical finite element analyses and a great expertise from the engineer in charge in order to obtain a viable solution. This challenge leads engineers in search of consistent and reliable tools that assist in the early stages of the riser configuration design and are capable of reducing the number of total analyses required. One of these tools is application of optimization methods to obtain in a consistent and reliable manner the parameters which define a configuration. This work presents the Bayesian Optimization method, a method based on machine learning techniques capable of efficiently solving so called black box problems by exploring analytical approximations of the objective function, the function to be minimized. The method is applied to different case studies aiming to validate it as capable of solving a wide variety of riser configuration problems in an efficient and consistent way. Among the problems applied are different types of configurations, different realistic cases, mono-objective and multi-objective.

## Keywords

Optimal Design; Bayesian Optimization; Riser design; Offshore systems; Subsea engineering.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Objetivo	15
1.2	Contribuições do Trabalho	15
1.3	Breve Revisão Bibliográfica	16
1.4	Organização	18
<b>2</b>	<b>Análise de <i>riser</i></b>	<b>19</b>
2.1	Casos de carregamento	19
2.2	Modelo numérico	20
<b>3</b>	<b>Otimização</b>	<b>23</b>
3.1	Otimizações mono-objetivo e multi-objetivo	24
3.2	Otimização global e local	26
3.3	Método de penalidade para restrições	27
3.4	Algoritmos	27
<b>4</b>	<b>Otimização Bayesiana</b>	<b>29</b>
4.1	Planejamento de experimentos	32
4.2	Regressão por Processos Gaussianos	34
4.3	Funções de aquisição	37
4.3.1	Probabilidade de Melhoria	38
4.3.2	Melhoria Esperada	39
4.3.3	Melhoria Esperada Baseada em Hipervolume	40
4.4	Otimização da função aquisição	41
4.5	Restrições	42
<b>5</b>	<b>Metodologia</b>	<b>44</b>
5.1	Paralelismo	47
<b>6</b>	<b>Estudo de caso</b>	<b>50</b>
6.1	Caso 1 - RCHA e SLWR mono-objetivo	50
6.1.1	<i>Steel Lazy Wave Catenary</i> (SLWR)	52
6.1.2	<i>Riser</i> em Catenária Hidroamortecido (RCHA)	53
6.1.3	Função objetivo	55
6.1.4	Restrições	55
6.1.5	Resultados	57
6.2	Caso 2 - SLWR com três casos de carregamento	61
6.2.1	Variáveis	63
6.2.2	Função Objetivo	64
6.2.3	Restrições	64
6.2.4	Resultados	65
<b>7</b>	<b>Conclusão e sugestões para trabalhos futuros</b>	<b>73</b>
<b>8</b>	<b>Referências bibliográficas</b>	<b>75</b>

## Lista de figuras

Figura 1.1	Produção de petróleo e gás natural <i>offshore</i> no período entre 2010 e 2020, retirado do Painel Dinâmico da ANP [1]	12
Figura 1.2	Produção de petróleo e gás natural <i>onshore</i> no período entre 2010 e 2020, retirado do Painel Dinâmico da ANP [1]	12
Figura 1.3	Exemplos de configuração de <i>riser</i> , retirado de [2] e adaptado de [3]	14
Figura 1.4	Configuração de <i>Riser</i> em Catenária Hidroamortecido	14
Figura 2.1	Modelo numérico 2D representativo da configuração em Catenária Livre do <i>riser</i>	22
Figura 3.1	Exemplo de Fronteira de Pareto, formada pelos pontos em preto	25
Figura 3.2	Fronteiras de pareto para um mesmo problema e seus "hipervolumes"	26
Figura 4.1	Fluxograma da otimização bayesiana	30
Figura 4.2	Esquema gráfico da otimização Bayesiana	32
Figura 4.3	Exemplo de amostragem <i>Latin Hypercube</i> para 2 variáveis	33
Figura 4.4	a) Três amostras tiradas aleatoriamente de um processo Gaussiano	35
Figura 4.5	Exemplo de regressão por processos Gaussianos. Os pontos em vermelho são pontos previamente avaliados, a linha em preto a função média e em verde a área de incerteza (Adaptado de [4]).	37
Figura 4.6	Representação gráfica do Probabilidade de Melhoria (PM) (adaptado de [4])	38
Figura 5.1	Operações de simplex do Nelder-Mead	47
Figura 5.2	Modelo de paralelismo Consumidor-Produtor	48
Figura 6.1	Representação visual dos seis graus de liberdade, três de rotação e três de translação	52
Figura 6.2	Configuração SLWR	53
Figura 6.3	Configuração RCHA	54
Figura 6.4	Configuração ótima de cada método para SLWR com LDA 1200 m. (a) BO, (b) MIDACO, (c) GBNM e (d) GA	58
Figura 6.5	Configuração ótima de cada método para SLWR com LDA 2150 m. (a) BO, (b) MIDACO, (c) GBNM e (d) GA	58
Figura 6.6	Configuração ótima de cada método para RCHA. (a) BO, (b) MIDACO, (c) GBNM e (d) GA	59
Figura 6.7	Evolução da melhor solução, BO x MIDACO	60
Figura 6.8	(a) Tempo médio de execução, para o BO o trecho em preto é o tempo da proposta do próximo ponto e em branco da avaliação da função objetivo; (b) Evolução do tempo médio da proposta de cada ponto e tempo de avaliação da função objetivo	61
Figura 6.9	Alturas Hog e Sag de uma configuração Lazy Wave	65
Figura 6.10	Exploração e soluções encontradas pelo método	69



Figura 6.11	Tempos de next loc e média para caso serial	70
Figura 6.12	Tempos de execução e média para caso serial	71
Figura 6.13	Tempos de next loc e média para caso paralelo	71
Figura 6.14	Tempos de execução e média para caso paralelo	72

## Lista de tabelas

Tabela 6.1	Propriedades físicas dos <i>risers</i>	51
Tabela 6.2	Movimento de topo do <i>riser</i> , imposto pelo FPSO	51
Tabela 6.3	Variáveis e os intervalos investigados representados por seus máximos e mínimos	53
Tabela 6.4	Variáveis RCHA e seus limites	54
Tabela 6.5	Propriedades dos Hidroamortecedores	54
Tabela 6.6	Restrições para o modelo SLWR	56
Tabela 6.7	Restrições para o modelo RCHA	56
Tabela 6.8	Resultados para o caso 1. Para o BO, o número de avaliações corresponde à soma de iterações da inicialização e do loop de otimização	57
Tabela 6.9	Resumo de simulações realizadas, em verde combinações que foram simuladas e em vermelho combinações que não foram simuladas	61
Tabela 6.10	Dados de ondas de cada caso de carregamento	62
Tabela 6.11	Dados de corrente para cada caso de carregamento, sendo H a profundidade, V a velocidade e D e direção.	62
Tabela 6.12	Propriedades do <i>riser</i>	63
Tabela 6.13	Variáveis, seus valores máximos, mínimos e incrementos, para o caso discreto	63
Tabela 6.14	Restrições do problema	64
Tabela 6.15	Variáveis contínuas e função objetivo para as três melhores configurações	66
Tabela 6.16	Restrições para as três melhores configurações com variáveis contínuas	66
Tabela 6.17	Variáveis discretas e função objetivo para as três melhores configurações	66
Tabela 6.18	Restrições para as três melhores configurações com variáveis discretas	67
Tabela 6.19	Fronteira de Pareto para as nove rodadas realizadas	68
Tabela 6.20	Variáveis correspondentes às soluções das fronteiras de paretos	68
Tabela 6.21	Resumo dos tempos do processo de otimização multi-objetivo	70

## Lista de Abreviaturas

ANP – Agência Nacional de Petróleo  
API – *Application Programming Interface*  
BO – Otimização Bayesiana  
DEAP – *Distributed Evolutionary Algorithm in Python*  
ME – Melhoria Esperada  
FPSO – *Floating Production Storage and Offloading*  
GBNM – *Globalized Bounded Nelder-Mead*  
LDA – Lâmina d'Água  
MIDACO – *Mixed Integer Distributed Ant Colony Optimization*  
MINLP – *Minxed Integer Non-Linear Programming*  
PE – Planejamento de Experimentos  
PG – Processo Gaussiano  
PM – Probabilidade de Melhoria  
PV – Probabilidade de Viabilidade  
RCHA – *Riser* em Catenária Hidroamortecido  
SLWR – *Steel Lazy-Wave Riser*  
SMTS – *Specified Minimum Tensile Strength*  
SMYS – *Specified Minimum Yield Strength*  
TDP – *Touchdown Point*  
TDZ – *Touchdown Zone*  
UEP – Unidade Estacionária de Produção  
VIV – Vibrações induzidas por vórtices

# 1

## Introdução

O desenvolvimento de novas tecnologias de perfuração e produção no ramo de Óleo & Gás possibilitou que empresas explorassem reservatórios em águas cada vez mais profundas. Esse desenvolvimento gerou um aumento significativo na produção: segundo a Agência Nacional do Petróleo (ANP) houve um aumento de 52,71% da produção de petróleo e de gás natural entre o período de 2010 e 2020 no Brasil [5]. Como pode ser observado nas Figuras 1.1 e 1.2, houve ainda, nesse mesmo período, um aumento significativo da produção *offshore* e uma redução na produção *onshore*.

Petróleo Equivalente

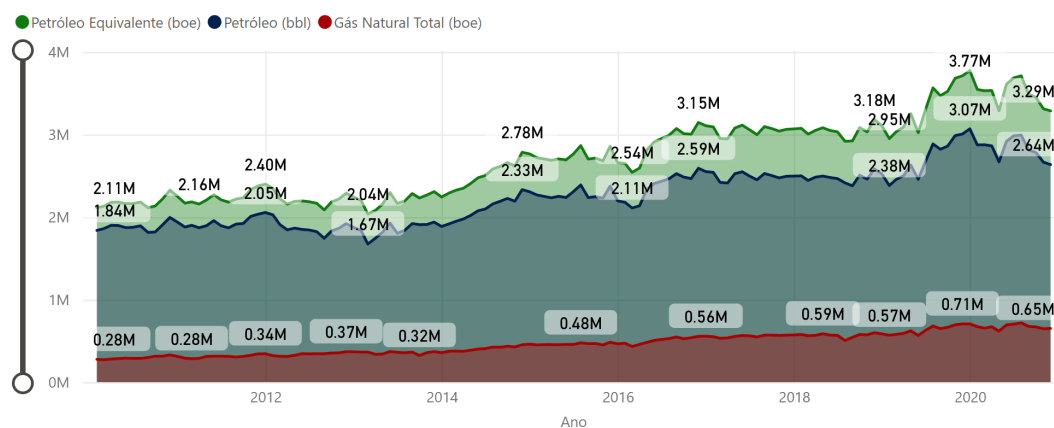


Figura 1.1: Produção de petróleo e gás natural *offshore* no período entre 2010 e 2020, retirado do Painel Dinâmico da ANP [1]

Petróleo Equivalente

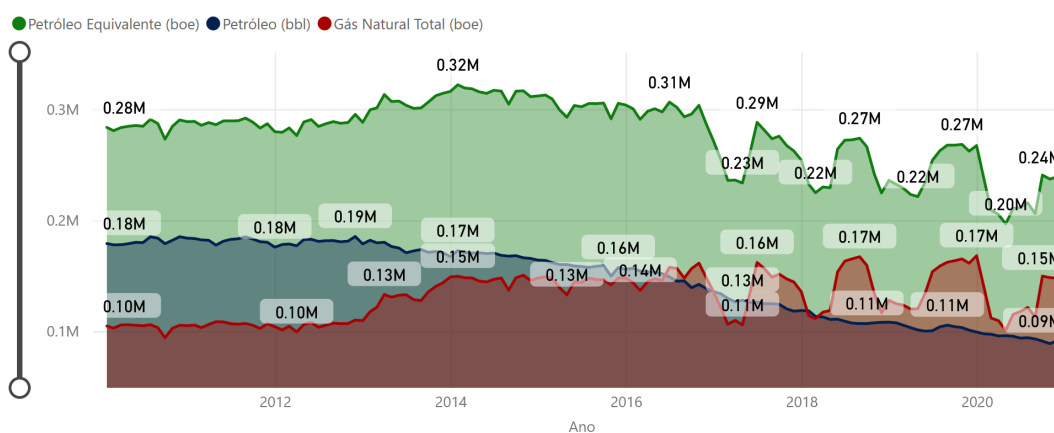


Figura 1.2: Produção de petróleo e gás natural *onshore* no período entre 2010 e 2020, retirado do Painel Dinâmico da ANP [1]

A descoberta de novos reservatórios com grande potencial ao redor do mundo (no Brasil, Golfo do México, oeste da África e outros) com lâmina d'água (LDA) podendo exceder 2000 m somada a exaustão de reservatórios de águas rasas impulsionaram novas tecnologias. A exploração de águas profundas apresenta desafios maiores em termos de custos e operação. Um dos grandes focos de melhora foram os *risers*, que são os dutos responsáveis pelo transporte de óleo e gás encontrados nos reservatórios até a Unidade Estacionária de Produção (UEP) e pela injeção de fluidos no reservatório.

Projetos de *risers* exigem muitos cuidados para sua implementação devido às altas tensões atuantes sobre a estrutura, alta pressão hidrostática, temperatura e ondas compressivas e de flexão causadas principalmente pelo movimento das plataformas. As condições ambientais adversas podem causar fadiga e flambagem, principalmente no ponto em que o *riser* entra em contato com o fundo do mar, também chamado de *Touchdown Point* (TDP), além de vibrações induzidas por vórtices (VIV). Ambos contribuem para diminuir a vida útil da estrutura e, conseqüentemente aumentar o custo de instalação.

Quando se trata de configuração de *riser*, a mais simples é a chamada Catenária Livre em que ele é simplesmente conectado em sua extremidade superior a UEP e sujeito ao peso próprio, descendo até encostar no fundo do mar e conectar-se a um equipamento submarino na outra extremidade. Devido à simplicidade dessa configuração e a seus custos menores, em geral é a primeira recomendação para aplicação. Essa configuração no entanto apresenta dificuldades a medida que a lâmina d'água aumenta pois o trecho suspenso passa a ser muito extenso e o peso passa a gerar tensões maiores principalmente no topo, inviabilizando a configuração em muitos cenários.

Alternativamente, foram desenvolvidas configurações que apresentam flutuadores instalados em um trecho central do *riser*, que faz com que seja suspenso devido ao empuxo. Os flutuadores possuem dois papéis importantes visando melhorar a integridade estrutural. Primeiramente eles aliviam parte da tensão concentrada na conexão de topo. Adicionalmente, esses componentes ainda desacoplam parcialmente o movimento do trecho superior, governado pelo movimento da embarcação, do trecho inferior, mitigando as ondas compressivas e conseqüentemente a fadiga sofrida na região. Estudos indicam que configurações com flutuadores possuem esforços de menor intensidade que a Catenária Livre [6]. Dependendo da posição dos flutuadores no *riser*, as configurações possuem diferentes designações. Exemplos de configurações com flutuadores são *Lazy Wave*, *Lazy S*, *Pliant-Wave*, *Steep Wave* e *Steep S* e são ilustrados na Figura 1.3.

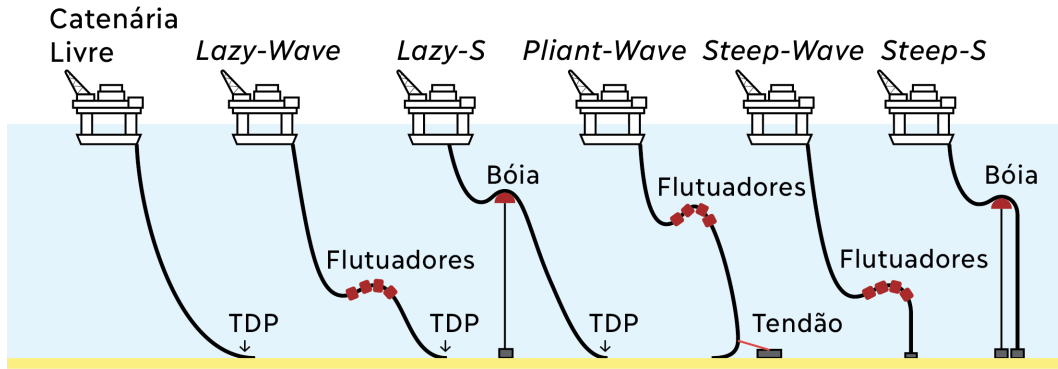


Figura 1.3: Exemplos de configuração de *riser*, retirado de [2] e adaptado de [3]

As configurações com flutuadores, apesar de eficientes, demandam logísticas mais complexas, custo mais elevado e maior tempo de instalação o que ainda não as tornam ideais em comparação com a catenária livre. Visando compensar essas deficiências técnicas e econômicas foi introduzida uma nova configuração, *Riser* em Catenária Hidroamortecido (RCHA) [7]. Os hidroamortecedores são revestimentos de baixa densidade que são instalados em diferentes trechos, cujos objetivos são impedir a propagação de ondas compressivas e induzir a flambagem em uma região mais afastada do TDP [8]. A Figura 1.4 apresenta essa nova configuração.

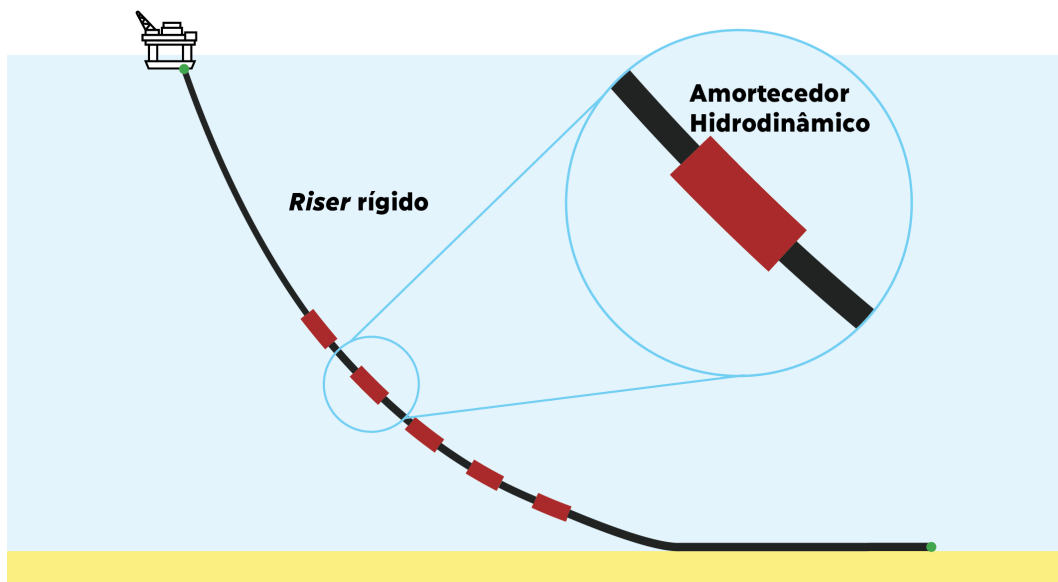


Figura 1.4: Configuração de *Riser* em Catenária Hidroamortecido

## 1.1

### Objetivo

Durante o projeto de configuração de *riser*, existem diversos parâmetros que devem ser definidos, tais como diâmetro e comprimento do *riser*, e para os casos em que flutuadores são presentes ainda é necessário dimensionar o diâmetro, espessura e comprimento, além dos espaçamentos e posicionamentos. O que define se uma configuração é viável ou não são os critérios estruturais e geométricos que garantem a sua integridade física, e que seja economicamente viável. Grande parte dos critérios estruturais são definidos por códigos de projeto que definem requisitos técnicos para que a estrutura seja considerada viável e segura. Portanto, além das inúmeras combinações possíveis entre os parâmetros que definem a configuração do *riser*, o projeto deve considerar diversas condições ambientais sobre a estrutura e garantir que os critérios de projeto sejam atendidos. Verifica-se então que existe um número muito grande de possibilidades de análises de *risers*.

Calcular todas essas possibilidades, embora seja o cenário ideal, não é uma alternativa viável. Tradicionalmente, adota-se a estratégia de tentativa e erro, em que uma combinação possível de parâmetros é testada utilizando algum software de elementos finitos. A partir dos valores encontrados os parâmetros são ajustados e reanalisados até que se obtenha uma combinação que resulte em uma configuração viável. Essa estratégia, no entanto, exige ainda um tempo muito grande e muita experiência do projetista. Ao fim, mesmo que seja encontrada uma combinação viável, é possível e até provável que exista uma outra combinação ainda melhor.

A fim de encontrar uma metodologia mais consistente de definição dos parâmetros de um projeto de *riser*, são estudadas técnicas de otimização, que visam obter a melhor combinação de parâmetros que definem uma configuração para um determinado problema. O objetivo desse trabalho é validar a aplicação da Otimização Bayesiana como um método consistente e competitivo quando comparado a outros métodos utilizados na literatura para otimização de configurações de *riser*. Os casos estudados utilizaram *risers* rígidos com análise de cargas extremas.

## 1.2

### Contribuições do Trabalho

Os principais métodos aplicados ao problema de otimização de *risers*, como por exemplo algoritmos evolutivos, em geral, necessitam de um número muito grande de avaliações da função objetivo durante o processo de otimização, implicando em um tempo elevado de execução. Este trabalho apresenta

uma nova aplicação do método de Otimização Bayesiana para a otimização de *risers*, considerando que este método já vem sendo aplicado com sucesso em outros problemas, como na área de robótica e sistemas autônomos [9, 10] e engenharia aeroespacial [11, 12]. O método utiliza técnicas de aprendizado de máquina para ser capaz de aplicar o processo de otimização com um menor número de avaliações da função objetivo e foi desenvolvido utilizando ferramentas *opensource*, não exigindo necessidade de licenças comerciais. O método permite a resolução de uma grande variedade de problemas, como otimizações mono-objetivo e multi-objetivo utilizando variáveis contínuas e discretas, podendo variar de complexidade e é capaz de lidar com restrições. Neste trabalho, as restrições foram resolvidas acoplando-se o método clássico de penalidade quadrática com o método de Probabilidade de Viabilidade comumente utilizado na Otimização Bayesiana. A aplicação foi implementada de maneira serial e paralela, reduzindo o tempo de execução e tornando a sua aplicação ainda mais atrativa durante o projeto de configuração de *risers*.

### 1.3

#### Breve Revisão Bibliográfica

A aplicação de técnicas de otimização em *risers* começou a ser aplicada no final da década de 80 e teve como um dos primeiros trabalhos publicados a utilização do método de programação sequencial matemática para minimizar o comprimento de uma configuração *steep-wave* Riggs e [13], Riggs e Leerand [14]. De lá para cá já foram aplicadas técnicas de otimização para diferentes configurações, como para minimizar custos de *risers* em catenária [15], *Lazy-S* [16] e *Lazy-Wave* [17, 18] e utilizando diferentes técnicas, como método de Nelder-Mead simplex [2], algoritmos genéticos [19] e algoritmos bio-inspirados [20, 21]. Todos exploram maneiras de encontrar rapidamente e consistentemente configurações viáveis e ótimas nas mais diversas circunstâncias, diminuindo o tempo de projeto, automatizando e tornando-o mais barato.

Na PUC-Rio o tema já foi estudado por Cardoso [22], que aplicou o método de *Globalized Bounded Nelder-Mead* ao problema de otimização de *risers* nas configurações *Lazy-Wave* e RCHA, e por Nariño [23], que aplicou Algoritmo Genético para a otimização da configuração RCHA.

Tratando-se de problemas de otimização de *risers*, seja para minimizar custos, comprimentos ou tensões, não há uma expressão analítica explícita das funções objetivos e/ou restrições. Não há uma forma simples de se formular o problema de maneira adequada para métodos que utilizem gradientes, que em geral são mais eficientes que métodos que não utilizam explicitamente essa informação. Essa restrição caracteriza um problema denominado "caixa-preta"



em que não há informações da função, avalia-se apenas utilizando dados de entrada e de saída de maneira iterativa.

Nos algoritmos, cada ponto avaliado necessita uma simulação de elementos finitos que pode ser extremamente custosa, especialmente quando são tratados múltiplos casos de carregamento simultaneamente. Em se tratando de métodos iterativos, que exigem muitas avaliações da função objetivo e restrições, essa etapa pode ser mais custosa que o método em si, dominando assim o tempo de execução. Uma das principais discussões atualmente tem sido em torno de encontrar a técnica mais eficiente, no sentido de exigir o menor número de avaliações para obter o resultado final.

O método de Otimização Bayesiana surgiu do trabalho de Kushner [24], Zhilinskas [25, 26] e Mockus [27, 28] e ganhou mais atenção com os estudos de Jones et al [29], que desenvolveu um algoritmo eficiente de otimização global. O método é baseado em modelos substitutos em que utiliza técnicas de regressão para estimar a função objetivo e restrições, quando presentes, estimando adicionalmente a incerteza associada, utilizando como base apenas pontos avaliados anteriormente. A partir dessas informações o problema é formulado como uma sequência de otimizações locais de funções denominadas "função de aquisição", construídas utilizando os modelos substitutos. Tal função é analítica e pode ser eficientemente otimizada utilizando métodos baseados em gradiente. O ponto resultado da otimização da função de aquisição é avaliado na função objetivo original e utilizado para a otimização global. O modelo substituto é construído a partir de um processo Gaussiano, o qual serve como regressor da função original e provê estimativas para a função e uma incerteza associada a essa estimativa.

A regressão por processo Gaussiano permite ter uma estimativa da função, gerando pontos potencialmente bons e faz com que o método seja eficiente na busca do mínimo global.

Nos últimos anos o método tem chamado bastante atenção de toda a comunidade de aprendizado de máquina, especialmente após Snoek [30] indicar que o método é extremamente útil para o ajuste de hyper-parâmetros de redes neurais profundas. O tema também foi aprofundado em [31] e [32].

O método de otimização Bayesiana possui aplicações de sucesso em diversas áreas, o que mostra a capacidade do método em resolver uma gama de problemas com características distintas. Alguns exemplos de aplicação são no design de antenas [33], materiais [34], engenharia aeroespacial [11, 12], predição de intrusão da água do mar [35], robótica e sistemas autônomos [9, 10], localização da fonte de contaminantes [36] e indústria bioquímica [37]. Na indústria de óleo e gás o método já foi utilizado para otimização de parâmetros

de perfuração em operações de fraturamento hidráulico [38]. A maioria desses problemas são aplicações em que a função objetivo e/ou restrições tem um custo computacional elevado.

Alternativamente outros métodos de modelo substituto foram desenvolvidos na literatura [39, 40] contendo uma lógica semelhante, diferenciando pela maneira como a regressão é feita e pelo mecanismo de escolha do ponto da iteração seguinte. Uma revisão dos métodos de modelos substitutos alternativos foi publicada em [41, 42].

O modelo base de Otimização Bayesiana é desenvolvido para resolver problemas que contém apenas um objetivo, entretanto foram desenvolvidas extensões que possibilitam a aplicação do método para otimização multi-objetivo. Uma das abordagens consideradas é utilizando o indicador de hyper-volume [43] de diferentes maneiras para a generalização das função de aquisição [44, 45, 46, 47]. Um resumo sobre o método e sua implementação pode ser encontrado em [48].

## 1.4

### Organização

O Capítulo 2 inicia uma breve explicação sobre a análise de *risers*, dos casos de carregamento e seu modelo numérico.

O Capítulo 3 introduz o problema de otimização genérico e apresenta alguns conceitos que serão importantes para o entendimento do problema.

O Capítulo 4 entra em maiores detalhes do método de otimização Bayesiana como um todo e entrando em detalhes em cada etapa do processo, como lidar com restrições e tornar o método capaz de resolver otimizações multi-objetivo.

O Capítulo 5 apresenta detalhes da implementação e metodologia aplicada e apresenta métodos utilizados para comparação de eficiência.

O Capítulo 6 expõe os casos estudados para validar e aferir a eficiência do método. Foram utilizados dois estudos de casos com complexidade diferentes com um e dois objetivos e para duas configurações diferentes: RCHA e *Lazy-Wave*.

No Capítulo 7 são realizadas discussões sobre os resultados do capítulo anterior, as conclusões e sugestões para trabalhos futuros.

## 2

### Análise de *riser*

A análise estrutural de *riser* possui um papel crucial no problema em questão. Na grande maioria de projetos de engenharia que envolvem otimização, o processo é realizado de maneira iterativa, onde em cada passo da otimização são coletados dados do problema que guiarão a iteração seguinte em busca do mínimo ou máximo. No caso da otimização de *riser*, não é viável realizar cada etapa desse processo experimentalmente e, dessa forma, é importante haver um módulo de simulação adequado que garanta que o modelo que satisfizer critérios estruturais na simulação será viável na aplicação prática.

Risers podem ser estruturas rígidas, tradicionalmente de aço, ou flexíveis, com diversas camadas de outros materiais. Eles se estendem da unidade flutuante ao solo. A estrutura no entanto está sujeita a movimentações causadas pelas condições ambientais, tais como ondas e correntes. Essa dinâmica faz com que o ponto de contato com o solo (TDP, em inglês, *Touchdown Point*) mude de lugar no *riser* e por isso é denominado de *Touchdown Zone* (TDZ) toda a região do *riser* que interage com o solo em sua dinâmica.

Para que então seja feita uma análise representativa, é preciso conhecer as grandezas que atuam na estrutura, sendo elas principalmente as ondas e correntes. Obtidos esses dados é possível prever o impacto causadas por elas direta e indiretamente. As ondas provocam movimentações na unidade flutuante que transmitem o movimento para o *riser*, propagando esforços axiais ondulatórios e forçando uma interação entre a estrutura e o solo; as ondas também atuam diretamente sobre os *risers*, contribuindo para a sua dinâmica; as correntes atingem os *risers* e podem provocar vórtices e com eles as vibrações induzidas por vórtices, podendo acelerar o processo de fadiga do material. O efeito do peso próprio da estrutura e de seu fluido interno também devem ser levados em consideração, e ele se torna ainda mais crítico com o aumento da Lâmina d'água (LDA).

### 2.1

#### Casos de carregamento

Ao longo de sua vida útil, a estrutura de um *riser* será exposta a diversas combinações de carregamentos de diferentes naturezas. A fim de representar essas combinações no modelo de análise são criados os casos de carregamentos, que irão simular as condições às quais a estrutura estará sujeita durante os anos de operação da unidade flutuante. De maneira geral, os carregamentos podem

ser divididos conforme a sua origem: funcionais, ambientais e acidentais.

Carregamentos funcionais provém da própria estrutura, como o peso próprio, o empuxo, pressão externa e interna. Carregamentos ambientais são aqueles causados seja pelo ambiente diretamente, como os de ventos, ondas e correntes, seja indiretamente como a partir do movimento que causam na embarcação que é posteriormente transferido ao *riser*. Carregamentos acidentais, como o nome indica, são esforços não previstos, seja por alguma falha ou interferência mas que devem ser levados em consideração para evitar um desastre ainda maior.

Dentre os diferentes carregamentos sobre a estrutura há alguns caracterizados como carregamentos estáticos, como por exemplo o peso próprio e empuxo, e outros como carregamentos dinâmicos, como os causados por ventos, ondas e correntes. Normalmente, a análise é realizada primeiro aplicando-se o carregamento estático e em seguida o dinâmico.

Os carregamentos, no entanto, não são constantes ao longo da vida útil do *riser*, já que correntes e ondas incidem sobre ele em diferentes direções e magnitudes. Dessa maneira, estudos são realizados para serem consideradas as várias possibilidades. Cada possibilidade representa uma combinação de carregamentos. Dessa forma a viabilidade estrutural do *riser* deve ser analisada por meio de vários casos de carregamentos refletindo os mais diversos cenários possíveis. O número total pode chegar a centenas e até milhares.

Há dois tipos de análises importantes para o estudo da viabilidade técnica de um *riser* : cargas extremas e fadiga. Para que uma configuração se torne viável é necessário que atenda a critérios de norma em ambos os tipos de análise. No caso extremo, busca-se avaliar as tensões máximas atuantes no *riser*. No caso de fadiga, não necessariamente a carga é alta, porém é estudado o comportamento da estrutura sob um carregamento cíclico por um longo período de tempo. Para uma estrutura ser considerada viável, ela deve satisfazer a critérios definidos em normas técnicas para ambos os casos de análise.

## 2.2

### Modelo numérico

Para que a análise de um sistema físico seja possível é necessário representá-lo por meio de um modelo numérico. O *riser* é modelado como um elemento de viga não linear bi ou tridimensional. Dessa forma, para um caso particular bidimensional, a equação analítica que rege a dinâmica de uma viga tensionada é:

$$[m(z) + m_a(z)] \frac{\partial^2 u}{\partial t^2} + c(z) \frac{\partial u}{\partial t} + \frac{\partial^2}{\partial z^2} \left[ EI(z) \frac{\partial^2 u}{\partial z^2} \right] - \frac{\partial^2}{\partial z^2} \left[ T_e(z) \frac{\partial u}{\partial z} \right] = f(z, t) \quad (2-1)$$

onde  $z$  é a coordenada axial,  $t$  é o tempo,  $u$  o deslocamento transversal,  $m(z)$  a massa da viga,  $m_a(z)$  a massa adicionada,  $c(z)$  o coeficiente de amortecimento,  $EI$  a rigidez a flexão da seção transversal,  $T_e$  a tensão efetiva na viga e  $f(z, t)$  as forças externas. No lado esquerdo da equação os dois primeiros termos se referem a força causada pela massa da estrutura, o terceiro referente ao amortecimento e os últimos dois são provenientes da equação estática da viga.

Como se trata de uma equação que envolve não linearidades, a resolução dessa equação de forma analítica se torna impraticável. Dessa maneira, no presente trabalho será utilizado o método dos Elementos Finitos. A equação matricial que rege a dinâmica do *riser* é apresentada pela Equação (2-2) e sua solução pode ser obtida por meio da integração numérica no tempo  $t$

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u})\mathbf{u} = \mathbf{F} \quad (2-2)$$

sendo  $\mathbf{M}$  a matriz de massa,  $\mathbf{C}$  matriz de amortecimento,  $\mathbf{K}(\mathbf{u})$  matriz de rigidez,  $\ddot{\mathbf{u}}$  vetor de aceleração,  $\dot{\mathbf{u}}$  vetor de velocidade e  $\mathbf{u}$  o vetor de deslocamento para os graus de liberdade dos pontos nodais da malha de elementos finitos que representa o *riser*. As condições iniciais para o problema dinâmico são obtidas a partir da solução do problema estático.

A estrutura do *riser* é discretizada em elementos de viga com seis graus de liberdade em cada extremidade, três de translação e três de rotação. O solo é representado por meio de molas e os movimentos da unidade flutuante são aplicados como movimentos prescritos em seis graus de liberdade no topo do *riser*. A Figura 2.1 apresenta uma representação esquemática do modelo numérico.

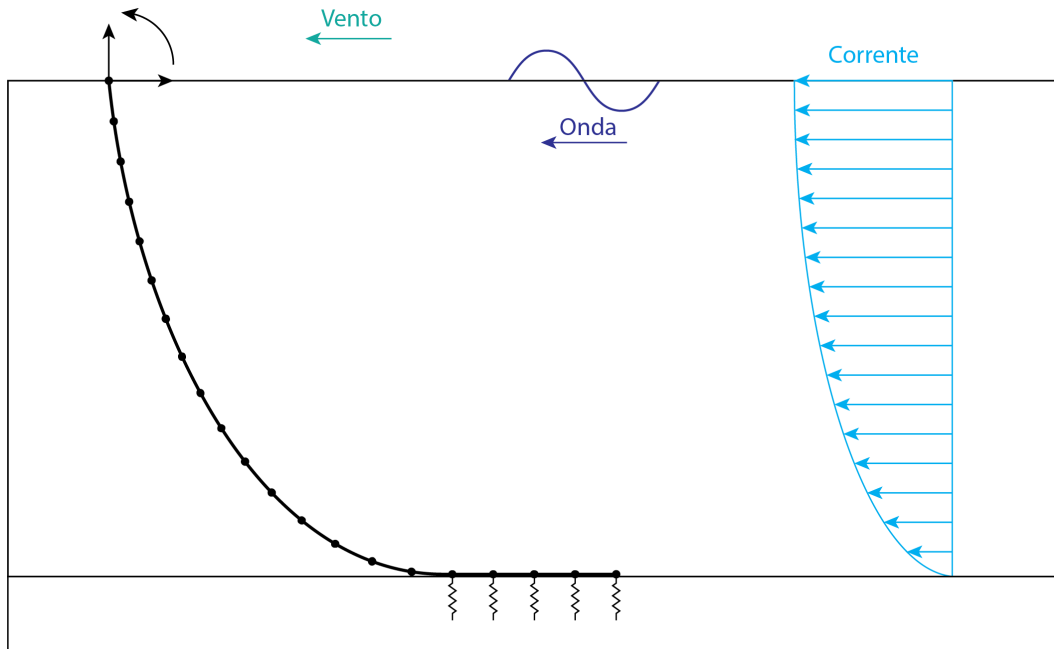


Figura 2.1: Modelo numérico 2D representativo da configuração em Catenária Livre do *riser*

Para efeito de análise, a corrente, o peso próprio e o empuxo são aplicados estaticamente e os demais carregamentos dinamicamente.

É importante ressaltar que, no presente trabalho, não será considerada a ação do vento no comportamento dinâmico do *riser* e que será considerada apenas a análise dinâmica do caso extremo para *risers* rígidos nas configurações *Lazy-Wave* e *RCHA* no espaço tridimensional. Os softwares de análises utilizados neste trabalho foram o Orcaflex [49] e ANFLEX [50].

### 3

## Otimização

A otimização está presente em diversas áreas de nossas vidas e da natureza. Sistemas físicos tendem a um estado de mínima energia, investidores querem minimizar o risco de suas ações e indústrias querem otimizar a eficiência de cada processo. Na matemática, a otimização é uma área de estudo que tem como objetivo encontrar a melhor solução para um determinado sistema. Esse objetivo pode ser definido por meio de uma formulação matemática ou de um resultado de uma simulação ou experimento e é denominado função objetivo. Os objetivos de um projeto estão relacionados às demandas dele, sejam elas de minimização de tensões, redução de custos ou de outra grandeza. Em todos os sistemas há parâmetros estabelecidos que definem o sistema e que resultam em um valor de função objetivo. A alteração desses parâmetros resulta em mudanças no sistema e no valor de função objetivo. Tais parâmetros são denominados variáveis de projeto e podem ser grandezas contínuas ou discretas no domínio. O resultado do processo de otimização consiste em apresentar os valores das variáveis que levam a minimização da sua função objetivo.

Durante a elaboração de um projeto, há diversas circunstâncias que devem ser levadas em consideração e que podem ser limitantes. Podem ser relacionados a tensões para que a estrutura esteja segura, de geometria para se adequar ao espaço encontrado ou de outra natureza. Quando essas circunstâncias existem, são levadas para o problema de otimização como restrições. As Equações (3-1) e (3-2) apresentam matematicamente o problema de otimização:

$$\min_{\mathbf{x}} f_{obj}(\mathbf{x}) \quad (3-1)$$

sujeito às restrições:

$$\begin{aligned} h_k(\mathbf{x}) &= 0 \quad k = 1...m \\ c_l(\mathbf{x}) &\leq 0 \quad l = 1...p \end{aligned} \quad (3-2)$$

sendo  $f_{obj}(\mathbf{x})$  a função objetivo que será minimizada,  $\mathbf{x}$  o vetor contendo as variáveis do projeto e as funções  $h_k(\mathbf{x})$  e  $c_l(\mathbf{x})$  as restrições de igualdade e desigualdade, respectivamente. Se a função objetivo e suas restrições são escritas a partir de funções lineares, o problema é caracterizado como problema de programação linear. Caso contrário, se ao menos uma delas for uma função não-linear, é um problema de programação não-linear. Esse segundo tipo é de uma complexidade maior e inclui a maioria dos problemas de engenharia, como

neste trabalho.

Quando se comparam dois pontos de uma mesma função objetivo, é comum descrever que um ponto é melhor que o outro quando ele está mais próximo do objetivo final. Para o caso em que a otimização busca a minimização de um função objetivo, um ponto é dito melhor que outro quando o valor da sua função objetivo é menor.

Nas próximas seções serão descritos mais alguns conceitos básicos porém importantes para a compreensão do problema de otimização.

### 3.1

#### Otimizações mono-objetivo e multi-objetivo

A classificação do problema de otimização se diferencia pelo número de objetivos que ele apresenta. Nos casos em que o problema possui mais de um objetivo, o problema é classificado como multi-objetivo e, caso contrário, é considerado problema de otimização mono-objetivo. Em muitas aplicações de engenharia, existem mais de uma grandeza que precisam ser otimizadas. Entretanto, muitas vezes os objetivos são conflitantes. A busca pela minimização de um objetivo pode resultar em um aumento do outro objetivo. Por exemplo, de maneira geral minimizar a espessura de um duto e a sua resistência a tensões atuantes sobre ele são conflitantes. A minimização da espessura pode tornar a estrutura mais frágil. Esse conflito faz com que os casos mono-objetivo e multi-objetivos sejam tratados de maneiras distintas.

Quando se trata de um problema mono-objetivo o resultado final será apenas um valor, que será o valor ótimo (mínimo) da função objetivo. Para problemas de multi-objetivo cada solução deixa de ser um valor e passa a ser um vetor em que cada componente está relacionada a um objetivo. O vetor pode ser representado como:

$$Y = (f_{obj1}(\mathbf{x}), f_{obj2}(\mathbf{x}), \dots, f_{objn}(\mathbf{x})) \quad (3-3)$$

sendo  $n$  o número de objetivos do problema. Devido à característica conflitante entre os objetivos, o resultado final da otimização será um conjunto de soluções. Isso se dá pois, comparando duas soluções  $Y_1$  e  $Y_2$ , é possível que  $Y_1$  tenha uma resposta melhor que  $Y_2$  para o primeiro objetivo, porém, pior para o segundo objetivo. Dessa maneira se introduz o conceito de dominância. Uma solução  $Y_1$  é dita não-dominada se não houver uma outra solução que seja melhor em todos os objetivos. O conjunto de todas as soluções não dominadas é então o resultado final da otimização e é denominado de Fronteira de Pareto. Na Figura 3.1 os pontos representados por um "x" são os pontos não dominados enquanto o restante dos pontos vermelhos, mais à direita são pontos dominados.



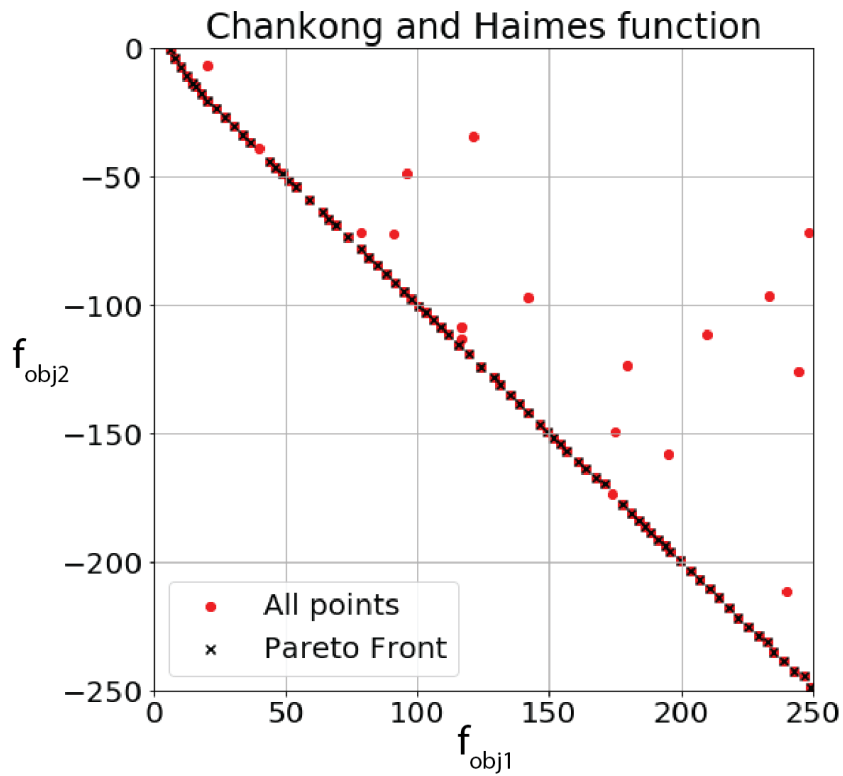


Figura 3.1: Exemplo de Fronteira de Pareto, formada pelos pontos em preto

Diferentes métodos podem resultar em diferentes Fronteiras de Pareto. Apenas observando as fronteiras não é possível definir se uma solução é melhor que a outra. Como uma métrica para quantificar e comparar dois métodos que geram duas fronteiras diferentes se calcula o hiper-volume entre um mesmo ponto de referência  $P_{ref}$ , que deve ser um ponto dominado por todas as soluções da Fronteira de Pareto, e a fronteira. A fronteira que resultar em um maior hiper-volume pode ser considerada a melhor. A Figura 3.2 apresenta duas fronteiras de Pareto para um mesmo problema, com dois valores de "hipervolume", que para o caso de dois objetivos se torna o cálculo de uma área.

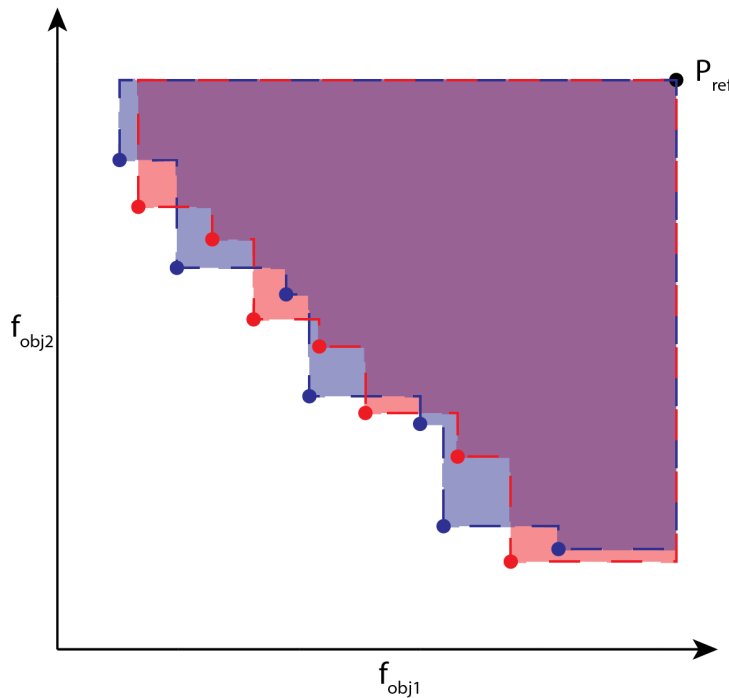


Figura 3.2: Fronteiras de pareto para um mesmo problema e seus "hipervolumes"

### 3.2

#### Otimização global e local

Otimização local é aquela em que o processo de otimização irá retornar apenas um mínimo local, um ponto em que todos os valores da função objetivo avaliada nos pontos localizados na sua proximidade são maiores que o valor da função objetivo no próprio ponto. Encontrar um mínimo local não significa no entanto que não há outro mínimo com valor da função objetivo menor em outra região do domínio. Dessa maneira há também o conceito de otimização global que, como o próprio nome indica, é aquela otimização que busca o menor de todos os mínimos locais, ou seja, o ponto que retorna o menor valor da função objetivo dentro do domínio.

Uma função é dita convexa quando apresenta apenas um valor de mínimo no domínio, que é então o mínimo global. Para funções não convexas, ou seja, que possuem mais de um mínimo local, não se pode garantir que os algoritmos encontrem o mínimo global e, portanto, limita-se a utilizar o menor valor dentre todos os mínimos encontrados.

Ambos os casos devem ser tratados de maneiras distintas. Um problema pode requerer apenas mínimos locais da solução ou o mínimo global. Mínimo global em geral é mais difícil de ser encontrado e muitos algoritmos conhecidos se baseiam em múltiplas buscas locais iniciadas de pontos diferentes visando

chegar a mínimos diferentes e sendo o menor deles mais provavelmente um mínimo global.

### 3.3

#### Método de penalidade para restrições

Um dos métodos clássicos mais conhecidos para lidar com restrições é o método da penalidade em que a função objetivo é substituída por uma nova função objetivo penalizada. Ela consiste da função original e um termo adicional para cada restrição do problema, tornando um problema de otimização com restrições em um problema sem restrições com função objetivo penalizada. A penalização força que o método saia da região de violação das restrições para a região viável, que não viole restrições.

A maneira mais simples de penalização é a penalização quadrática e é representada matematicamente pela Equação (3-4), sendo  $Q$  a nova função objetivo penalizada

$$Q(\mathbf{x}; \mu) = f_{obj}(\mathbf{x}) + \frac{1}{2\mu} \left( \sum_{k \in m} h_k^2(\mathbf{x}) + \sum_{l \in p} \max(0, c_l)^2 \right) \quad (3-4)$$

Analizando a equação, o primeiro termo é a sua função objetivo original e os termos restantes estão relacionados a inserção das restrições na função objetivo. Quando as restrições são violadas, esses termos adicionais são positivos e atuam sobre a função objetivo para ir a uma região sem violação. A medida que as restrições deixam de ser violadas, os termos vão se anulando e ao final, quando todas as restrições estiverem satisfeitas, a função objetivo penalizada converge para a função objetivo original.

O termo  $\mu$  da equação, chamado coeficiente de penalidade, assume apenas valores positivos e é o termo indicativo de quão forte será a penalização. Se colocarmos valores baixos, próximos de zero, a penalização será forte e o método será mais rapidamente forçado a uma região sem violação de restrições e à medida que o valor aumenta a função é mais suave. Recomenda-se, em geral, a inicialização de um coeficiente de penalidade  $\mu_1$  com um valor pequeno ( $\mu_1 = 0.5$ , por exemplo) e, para cada passo do processo esse coeficiente vai sendo aumentado de um fator  $\beta$ , ou seja,  $\mu_{k+1} = \mu_k \cdot \beta$ , com  $\beta = 2$ , por exemplo.

### 3.4

#### Algoritmos

Existem diversos tipos de algoritmos para resolver um problema de otimização. A qualidade do algoritmo pode ser avaliada principalmente por três características [51]: robustez, eficiência e acurácia. Robustez está ligada à gama de problemas que o método é capaz de solucionar; um método

é denominado eficiente se o custo computacional utilizado por ele não for excessivo para o problema e a acurácia do algoritmo está relacionado à precisão da solução, independentemente de erros ou ruídos do modelo. Muitas vezes essas características podem ser conflitantes entre si. Nesses casos, cabe ao usuário analisar qual delas é mais importante e priorizá-la.

A escolha do algoritmo se baseia em diversos fatores mas principalmente na quantidade de informação que se tem sobre a função objetivo e sua complexidade. Os métodos empregados pelos algoritmos podem ser classificados da seguinte maneira: métodos de ordem zero que são aqueles em que o processo ocorre utilizando apenas os valores da função objetivo em um determinado ponto; métodos de ordem 1, que são aqueles que utilizam informações sobre a primeira derivada (gradiente) para a obtenção do mínimo e os métodos de ordem 2, que são aqueles que utilizam primeiras e segundas derivadas (gradientes e matrizes hessianas, respectivamente).

De maneira geral, os métodos de ordem 2 são aqueles que, por se ter mais conhecimento da função e suas derivadas, possuem a convergência mais rápida, conseguindo assim chegar ao mínimo com menor número de iterações. No caso real de engenharia, entretanto, muitas vezes é difícil obter informações sobre a função objetivo e suas derivadas. Em sua maioria são necessárias simulações, cálculos custosos, funções e restrições não lineares e não convexas e então diferentes métodos foram desenvolvidos para suprir essa necessidade. No próximo capítulo será discutido em detalhes um desses métodos que é o principal utilizado no trabalho: Otimização Bayesiana.

O método de Otimização Bayesiana foi selecionado como objeto desse estudo devido a sua característica de necessitar poucas avaliações da função objetivo para resolver o problema de otimização. Na otimização de configurações de *risers* cada avaliação da função objetivo consiste em uma simulação computacionalmente custosa e, por isso, é importante que o método utilize o menor número de avaliações da função objetivo possível.

## 4

### Otimização Bayesiana

O método de otimização Bayesiana (BO) é baseado em Processos Gaussianos (PG) e foi desenvolvido para ser capaz de resolver problemas de otimização global caixa-preta de maneira eficiente, não utilizando gradientes da função objetivo mas explorando o uso de gradientes de aproximações analíticas da função objetivo. O termo caixa preta se refere ao fato de que o desenvolvimento do método é realizado com base no valor pontual da função objetivo e restrições e não utilizando valores de derivadas. Por consequência, um método para otimização em caixa preta só requer a avaliação da função objetivo em pontos requeridos mas não os gradientes da mesma.

O método de otimização Bayesiana reescreve o problema de otimização caixa preta como uma sequência de otimizações analíticas: A cada iteração, os pontos já avaliados da função objetivo são interpolados utilizando um método de regressão como regressão por Processos Gaussianos. A partir da interpolação é construída uma função objetivo intermediária, denominada Função de Aquisição (FA). O ponto que maximiza essa função de aquisição indica o ponto seguinte a ser avaliado na função objetivo original, como expressa a Equação (4-1) (sendo  $FA$  a função de aquisição)

$$x_{n+1} = \arg \max_{\mathbf{x}} FA(\mathbf{x}) \quad (4-1)$$

A Figura 4.1 apresenta um fluxograma do passo a passo a ser seguido pelo método com uma breve descrição dos passos na sequência.

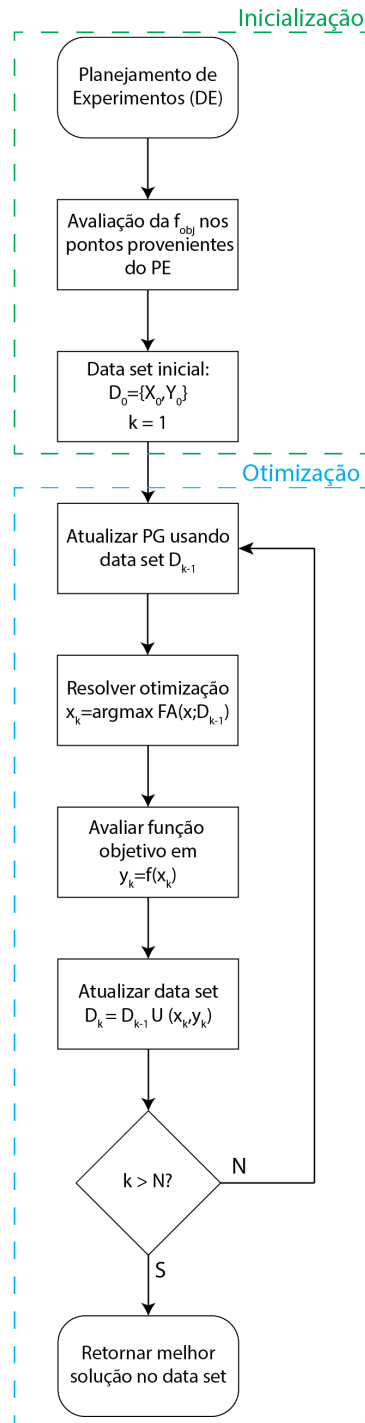


Figura 4.1: Fluxograma da otimização bayesiana

O método requer duas etapas: Inicialização e Otimização. A etapa de inicialização tem como objetivo analisar pontos do domínio que são armazenados em um *dataset* para que sejam utilizados na realização de uma regressão, que será ponto de partida da etapa seguinte.

O processo se inicia a partir do *design* de experimentos, em que se definem os pontos que serão utilizados durante o processo de inicialização. Esses pontos

podem ser selecionados de maneira randômica ou por outras técnicas, que serão apresentadas na Seção 4.1. Esses pontos escolhidos serão avaliados na função objetivo e dando início ao *dataset*. O passo seguinte consiste em gerar uma regressão entre eles para que se possa dar início à etapa de otimização. Em geral, esse processo é feito a partir de Processos Gaussianos [52] ou Kriging [53]. A regressão tem como resultados informações sobre a média da distribuição do processo e também a incerteza sobre ela. Outros métodos são estudados como alternativas quando são necessários um número grande de avaliações da função objetivo [54].

A etapa de otimização se inicia avaliando a regressão realizada sobre o *dataset* e extraindo dela informações sobre o melhor ponto a ser avaliado em seguida. A intenção da regressão não é modelar perfeitamente a função inteira, mas sim descobrir e mapear os pontos em que há um provável mínimo e áreas de incerteza grande. Se há uma área em que se sabe que não há pontos relevantes próximo de um mínimo, não é necessário avaliar essa região. O processamento dessas informações de média e incerteza são auxiliadas a partir de uma função de aquisição. Os diferentes tipos, vantagens e desvantagens são discutidos na Seção 4.3. A função de aquisição é capaz de avaliar o quão vantajoso será avaliar cada ponto do domínio segundo critérios de cada função e então, maximizando essa função se obtém o ponto mais vantajoso a ser avaliado. A vantagem de se trocar um problema de otimização por outro, de maximizar a função de aquisição, é que nova a função a ser otimizada é analítica e portanto diferenciável, podendo-se usar um método mais eficiente para otimizá-la, de ordem 2. O ponto que maximizar a função de aquisição é então avaliado na função objetivo do problema e a etapa se reinicia refazendo a regressão por processos Gaussianos para avaliar a iteração seguinte. Não há ainda na literatura um consenso sobre critério de parada do método, sendo mais comumente definido pelo usuário o número de iterações que serão avaliadas na inicialização e na etapa de otimização.

O processo de otimização para os casos mono-objetivo e multi-objetivo modifica basicamente as etapas interpolação por PG e da função de aquisição. Para o caso mono-objetivo haverá uma regressão para a única função objetivo e no caso multi-objetivo haverá uma regressão para cada objetivo. Assim, é necessário que haja uma nova função de aquisição adaptada, capaz de incorporar dados de todos os objetivos em uma única função.

Para lidar com restrições a alteração também é na função de aquisição, que será explicada em mais detalhes na seção 4.3.3.

A Figura 4.2 apresenta um exemplo simples do processo de otimização Bayesiana. A primeira imagem apresenta dois pontos previamente avaliados

na etapa de inicialização e a sua regressão. Em preto tracejado observa-se a verdadeira função, a função objetivo do problema, em vermelho a média da regressão e a área em verde representando a área de incerteza, menor próximo a pontos conhecidos e maior em pontos mais distantes. Mais abaixo há a função de aquisição correspondente, modelando os ganhos de cada ponto possível do domínio. Maximizando-a obtém-se o ponto marcado por uma seta vermelha e aquele ponto é o ponto avaliado para a iteração seguinte, marcado em vermelho. a regressão é refeita e o processo se repete ainda mais uma vez.

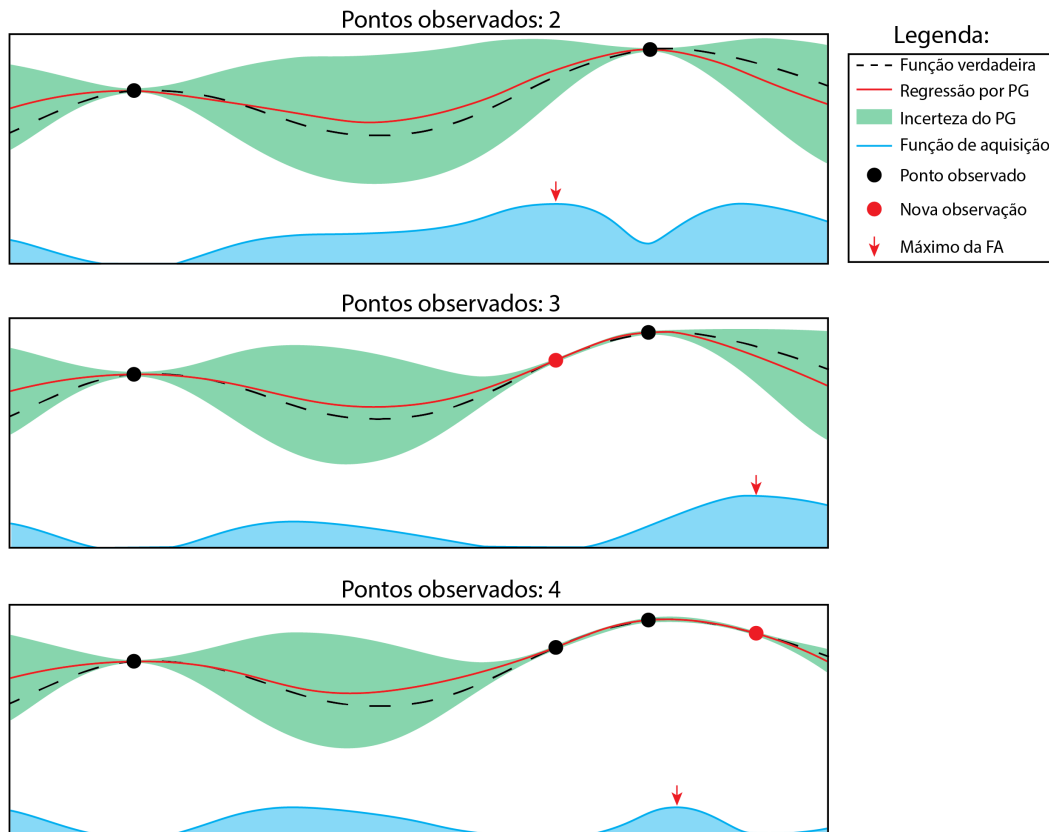


Figura 4.2: Esquema gráfico da otimização Bayesiana

Nas seções seguintes serão descritas cada passo do método mais detalhadamente.

#### 4.1

##### Planejamento de experimentos

O planejamento de experimentos (PE) é a primeira e importante etapa do processo pois os pontos a serem definidos aqui servirão como guia para a escolha nas iterações seguintes do método. A maneira mais fácil seria gerar pontos de maneira aleatória no domínio. Entretanto, essa estratégia afeta o seu desem-



penho em termos de consistência. Idealmente, essa etapa tem como objetivo gerar pontos de maneira uniforme pelo domínio, para que na etapa seguinte de otimização o método consiga ter um melhor desempenho, não tendo que avaliar pontos em regiões totalmente desconhecidas desnecessariamente. Dentre os métodos comumente utilizados com este objetivo, destacam-se o Método Aleatório que, apesar de sua inconsistência, é o menos custoso computacionalmente e apresenta bons resultados para domínios pequenos, Hipercubo Latino e Sequência de Sobol [55]. Quando comparadas as inicializações desses três métodos, concluiu-se que amostragem Latin Hypercube possui um desempenho melhor [55] e, portanto, é o método padrão recomendado para a Otimização Bayesiana.

A amostragem de pelo método Hipercubo Latino consiste em dividir o domínio de cada variável em  $N$  intervalos de probabilidades iguais, sendo  $N$  o número de pontos da sua inicialização. Para o caso de duas dimensões é selecionada uma região desse novo "tabuleiro" de maneira que não haja uma outra região selecionada na mesma linha e coluna como pode ser observado na Figura 4.3. Para finalizar, o ponto é escolhido aleatoriamente dentro daquela região a ele designada. Esse conceito pode ainda ser estendido para mais dimensões evitando a escolha de regiões no eixo de cada variável, tornando os pontos menos concentrados.

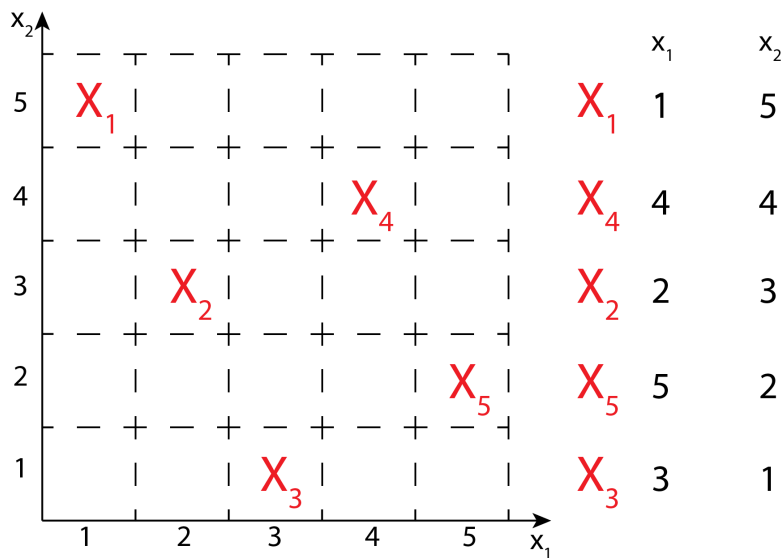


Figura 4.3: Exemplo de amostragem *Latin Hypercube* para 2 variáveis

## 4.2

**Regressão por Processos Gaussianos**

Com os dados dos pontos já avaliados e armazenados em um *dataset*, o objetivo desta etapa é inferir a relação entre o dado de entrada da otimização, no caso o vetor de variáveis, e o dado de saída, o valor da função objetivo. A maneira mais simples de realizar tal procedimento seria restringir o tipo de funções, como por exemplo para apenas funções lineares e tentar encontrar a que melhor se adequa aos pontos já encontrados. Entretanto, na maioria dos problemas de engenharia as funções objetivos e restrições são funções não convexas e a tarefa de restringir acaba gerando um erro muito grande. Outra possibilidade consiste em se calcular a probabilidade a priori para cada função possível, dando maior probabilidade àquelas que os dados se aproximam dos valores já encontrados. Essa possibilidade no entanto também não é viável por existirem infinitas funções e não há como avaliar todas elas. Para solucionar esses problemas existem técnicas como a regressão por processos Gaussianos, que é o foco de estudo do trabalho.

Em sua definição, um processo Gaussiano (PG) é um conjunto contendo um número finito de variáveis aleatórias, que possuem uma distribuição Gaussiana conjunta [52]. Um processo Gaussiano  $f(\mathbf{x})$  (Equação (4-2)) é definido pela sua função média  $m(\mathbf{x})$  e covariância  $k(\mathbf{x}, \mathbf{x}')$  descritos na Equação (4-3) em que  $\mathbf{x}$  e  $\mathbf{x}'$  são dois pontos distintos do domínio de variáveis, ou seja

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (4-2)$$

e

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned} \quad (4-3)$$

podendo o valor esperado ser calculado na sua forma integral para variáveis contínuas como  $\mathbb{E}[X] = \int_{\mathbb{R}} x \cdot \phi(x) dx$ , sendo  $\phi$  a função de densidade de probabilidade.

A partir da informação de pontos, obtendo a média e a covariância do processo Gaussiano, é possível amostrar funções com essa distribuição, como pode-se observar da Figura 4.4 a), assumindo uma média zero. Nesse caso três funções distintas foram amostradas.

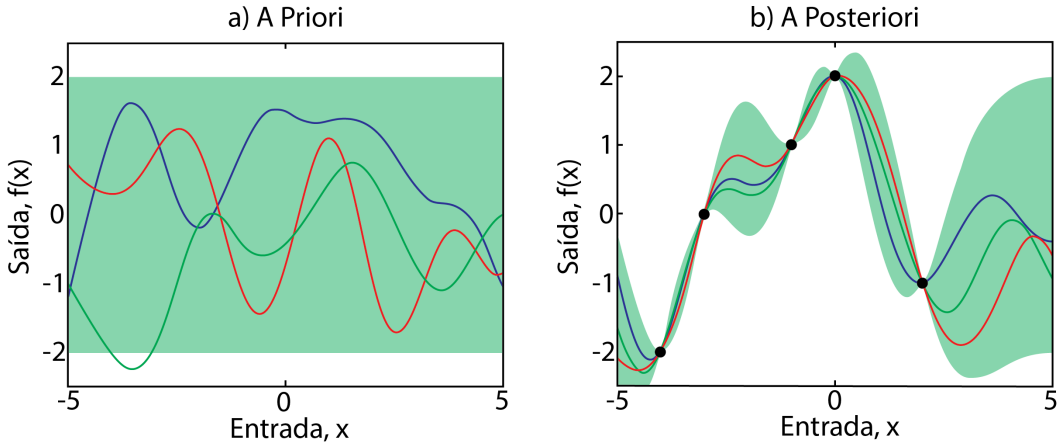


Figura 4.4: a) Três amostras tiradas aleatoriamente de um processo Gaussiano a priori. b) Três amostras tiradas de um processo Gaussiano a posteriori, condicionado a pontos já avaliados anteriormente, representados por pontos pretos (Adaptado de [52]).

Para o caso de regressão, as variáveis aleatórias representam o valor da função  $f(\mathbf{x})$  no ponto  $\mathbf{x}$ . É comum estabelecer para um processo Gaussiano o índice do tempo, ou seja, ter a média e variância para cada instante de tempo. No caso da regressão, o que se busca é saber a média e variância para cada vetor de variáveis  $\mathbf{x}$  dentro de todo o domínio  $\mathbf{X}$ . O PG pode então ser interpretado como uma distribuição sobre funções, de maneira que, para cada vetor de variáveis  $\mathbf{x}$ , há uma correspondente média e variância obedecendo a uma distribuição Gaussiana. A covariância  $k(\mathbf{x}, \mathbf{x}')$  também é chamada de função *kernel*, uma função cujo objetivo é mapear a relação de dois dados de entrada  $\mathbf{x}$  e  $\mathbf{x}'$ . Há na literatura múltiplas opções de *kernel* capazes de mapear essa relação, sendo nesse trabalho utilizada a função Matérn-5/2 [56].

O foco para resolver o problema de otimização é, dada algumas informações já coletadas sobre a função, ser capaz de estimar valores para pontos ainda desconhecidos. Essa tarefa então indica que deve ser realizada a probabilidade condicional no sentido de que, dado um conjunto dados iniciais  $\mathcal{D} = \{(x_i, f(x_i)), i = 1 : N\}$ , objetiva-se prever o valor de outros pontos de teste. Essa aplicação leva a Figura 4.4 b, em que agora os pontos já observados são levados em consideração na geração de amostras.

Aplicando a probabilidade condicional em uma distribuição Gaussiana multivariada, obtém-se também uma distribuição normal com as expressões descritas em (4-4) e (4-5), i.e.

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix} \right) \quad (4-4)$$

$$\begin{aligned}
p(\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{f}) &= \mathcal{N}(\mathbf{f}_* | \mu_*, \sigma_*) \\
\mu_n^*(\mathbf{x}) &= \mu(\mathbf{X}_*) + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{f} - \mu(\mathbf{X})) \\
\sigma_*^2(\mathbf{x}) &= \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*
\end{aligned} \tag{4-5}$$

sendo:

- $\mathbf{f}$  os valores de função já conhecidos
- $\mathbf{f}_*$  os valores de função desconhecidos, que se deseja conhecer
- $\mathbf{X}$  pontos do *dataset* já conhecidos
- $\mathbf{X}_*$  pontos em que se quer informação do valor da função
- $\mu$  a média do PG
- $\mu_*$  média do PG condicional
- $\sigma_*$  covariância da condicional
- $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$  a matriz de covariância do *dataset* original
- $\mathbf{K}_* = k(\mathbf{X}, \mathbf{X}_*)$  a matriz de covariância dos dados originais com novo ponto que se deseja prever
- $\mathbf{K}_{**} = k(\mathbf{X}_*, \mathbf{X}_*)$  covariância do ponto que se deseja prever

Essas equações geram então expressões analíticas, que serão necessárias para estimar o ganho de cada ponto, que é tarefa das funções de aquisição, detalhada na Seção 4.3. A Figura 4.5 ilustra uma regressão por processos Gaussianos.

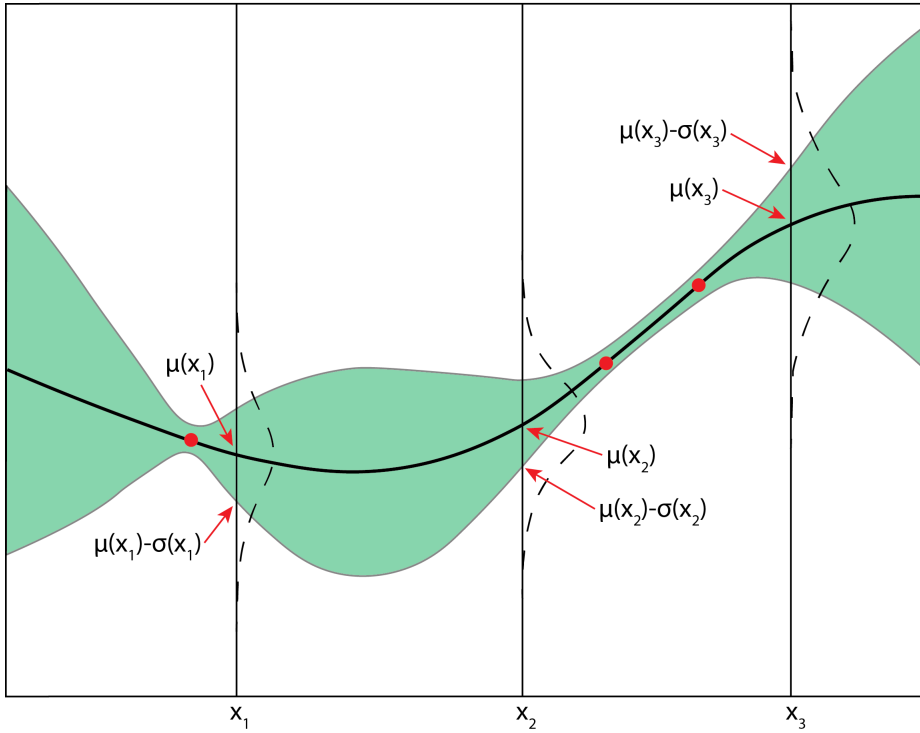


Figura 4.5: Exemplo de regressão por processos Gaussianos. Os pontos em vermelho são pontos previamente avaliados, a linha em preto a função média e em verde a área de incerteza (Adaptado de [4]).

### 4.3

#### Funções de aquisição

Após a modelagem da função, a etapa seguinte consiste na tomada de decisão sobre qual ponto será avaliado pela iteração seguinte. Existe um dilema neste processo (em inglês conhecido como *exploration-exploitation*) em que de um lado é necessário explorar regiões não bem conhecidas, que são regiões em que há uma incerteza grande podendo ou não retornar um ponto ótimo e de outro lado há o desejo de explorar ainda mais regiões que já são conhecidas e que potencialmente retornarão um ponto ótimo. Com intuito de inserir este dilema matematicamente no problema foram desenvolvidas as chamadas Funções de Aquisição (FA). São funções que incorporam informações de média e variâncias e que de alguma maneira balanceiam esse *trade-off* e guiam o problema de otimização.

Um estudo foi realizado comparando o desempenho da otimização escolhendo o ponto seguinte de três maneiras diferentes: Aleatoriamente, selecionando sempre o ponto com maior variância e utilizando a função de aquisição Melhoria Esperada, que será apresentada em 4.3.2 [4]. A solução que usava a função de aquisição foi consistentemente mais precisa, retornando a melhor so-

lução para o problema proposto. Nas seções seguintes serão apresentadas algumas das funções de aquisição, como Probabilidade de Melhoria [24] e Melhoria Esperada [26], que são duas das principais funções para funções mono-objetivo, e Hypervolume based Expected Improvement para problemas multi-objetivo.

### 4.3.1

#### Probabilidade de Melhoria

O método de Probabilidade de Melhoria (PM) tem como proposta calcular a probabilidade de um novo ponto ser ainda melhor que o melhor ponto encontrado até aquele momento. Para isso é preciso calcular a probabilidade condicional, dado o *dataset* observado, do valor da função objetivo em um ponto ser melhor que o melhor valor da função objetivo já observado, da seguinte forma

$$PM(\mathbf{x}) = P(f(\mathbf{x}) \leq f(\mathbf{x}^+) + \xi | \mathcal{D}_n) \quad (4-6)$$

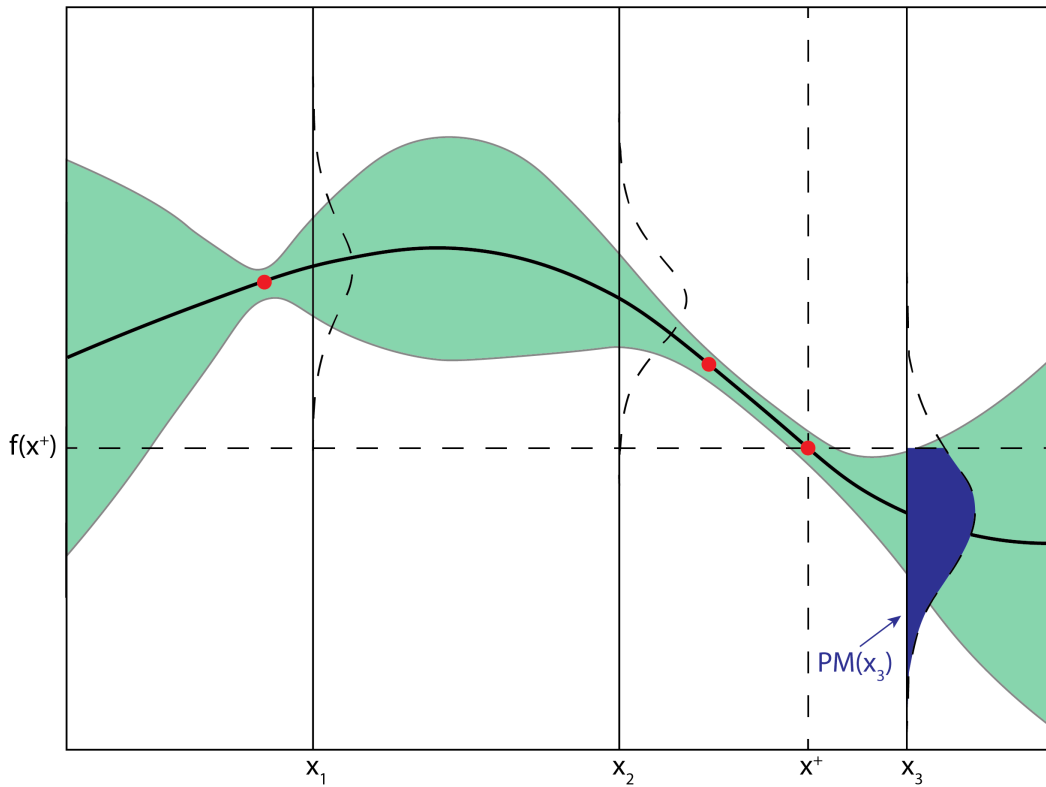


Figura 4.6: Representação gráfica do Probabilidade de Melhoria (PM) (adaptado de [4])

onde  $\mathbf{x}$  é o ponto candidato a nova iteração,  $\mathbf{x}^+$  o ponto que gera o maior valor de função objetivo,  $f(\mathbf{x}^+)$  o valor da função objetivo neste ponto e  $\xi$  um termo muitas vezes chamado de "jitter", que assume um valor muito pequeno. Essa equação pode ainda ser descrita como a probabilidade cumulativa de um ponto

ser maior que aquele valor, e é representada pela área azul escura na Figura 4.6

$$PM(\mathbf{x}) = \Phi \left( \frac{f(\mathbf{x}^+) - \mu^*(\mathbf{x}) + \xi}{\sigma^*(\mathbf{x})} \right) \quad (4-7)$$

em que  $\Phi$  é a função cumulativa de probabilidades.

Essa técnica sempre busca pontos que são melhores que o anterior, havendo pouca margem para que o método explore áreas não tão conhecidas, ou seja, haver um mínimo até menor que o previsto na região mais conhecida. O termo *jitter* foi incorporado com o objetivo de aumentar o poder exploratório do método no domínio das variáveis. Ele permite que um termo seja adicionado ao melhor ponto observado visando aumentar a área de busca e pontos possíveis do método. Esse parâmetro pode ter um valor fixo durante todo o processo ou pode ser ajustado dinamicamente, tipicamente iniciando com um valor mais alto e diminuindo com as iterações, tornando a busca pelo melhor ponto mais intensa.

Por ser um método em que se explora pouco o domínio, ele é mais indicado para situações em que o usuário possui conhecimento prévio do valor do mínimo ou pelo menos uma estimativa dele e sua região. Como em grande parte dos problemas não se possui essa informação o mais indicado é utilizar outros métodos como o Melhoria Esperada.

### 4.3.2 Melhoria Esperada

O Melhoria Esperada (ME) se baseia no critério de Utilidade Máxima Esperada (UME). Nesse critério, a escolha do ponto seguinte é realizada buscando o ponto que minimiza o valor esperado da diferença entre os valores da função real do problema  $f_{true}(\mathbf{x}^*)$  e da função oriunda da regressão por processos Gaussianos  $f(\mathbf{x})$ :

$$\begin{aligned} \mathbf{x}_{n+1} &= \arg \min_{\mathbf{x}} \mathbb{E}(\|f_{true}(\mathbf{x}^*) - f(\mathbf{x})\| \mid \mathcal{D}_n) \\ &= \arg \min_{\mathbf{x}} \int \|f_{true}(\mathbf{x}^*) - f(\mathbf{x})\| p(f(\mathbf{x}) \mid \mathcal{D}_n) df \end{aligned} \quad (4-8)$$

Entretanto, na maioria dos problemas não se possui a função real do problema. Então, para superar esse problema, Mockus [26] propôs a função Melhoria Esperada, utilizando o critério de *Improvement* ou fator de aprimoramento ( $I(\mathbf{x})$ ). Em sua definição, é a diferença entre o valor da função do melhor ponto observado e o valor da função interpolada, sendo positivo quando a solução do novo ponto é menor que o atual melhor valor e zero caso contrário. Pode ser descrito matematicamente como:

$$I(\mathbf{x}) = \left( \max\{0, f(\mathbf{x}^+) - f(\mathbf{x}) \mid \mathcal{D}_n\} \right) \quad (4-9)$$

em que  $\mathbf{x}^+$  é o melhor ponto já observado e  $x$  o vetor de variáveis do domínio. A partir desse conceito, o EI passa a ser o valor esperado dessa função de ganho  $I(\mathbf{x})$  com função semelhante ao PoI. Essa função de aquisição é expressa da seguinte forma

$$ME(\mathbf{x}) = \mathbb{E} \left( \max\{0, f(\mathbf{x}^+) - f(\mathbf{x}) \mid \mathcal{D}_n\} \right) \quad (4-10)$$

A função de verossimilhança do aprimoramento  $\mathcal{L}(I)$  em uma distribuição a posteriori definida pela  $\mu_n^*(\mathbf{x})$  e  $\sigma_n^*$  é calculada pela função de densidade da distribuição normal

$$\mathcal{L}(I) = \frac{1}{\sqrt{2\pi}\sigma_n^*(\mathbf{x})} \exp \left( -\frac{(f(\mathbf{x}^+) - \mu_n^*(\mathbf{x}) - I)^2}{2(\sigma_n^*(\mathbf{x}))^2} \right) \quad (4-11)$$

A função Melhoria Esperada é então definida pela integral sobre essa função de verossimilhança

$$\begin{aligned} \mathbb{E}(I) &= \int_{I=0}^{I=\infty} I \frac{1}{\sqrt{2\pi}\sigma_n^*(\mathbf{x})} \exp \left( -\frac{(f(\mathbf{x}^+) - \mu_n^* - I)^2}{2(\sigma_n^*)^2(\mathbf{x})} \right) dI \\ &= \sigma_n^*(\mathbf{x}) \left[ \frac{f(\mathbf{x}^+) - \mu_n^*(\mathbf{x})}{\sigma_n^*(\mathbf{x})} \Phi \left( \frac{f(\mathbf{x}^+) - \mu_n^*(\mathbf{x})}{\sigma_n^*(\mathbf{x})} \right) + \phi \left( \frac{f(\mathbf{x}^+) - \mu_n^*(\mathbf{x}) - \xi}{\sigma_n^*(\mathbf{x})} \right) \right] \end{aligned} \quad (4-12)$$

A função de aquisição Melhoria Esperada pode ser descrita analiticamente, incluindo dessa vez o *jitter*, da seguinte forma

$$ME(\mathbf{x}) = \begin{cases} (f(\mathbf{x}^+) - \mu_n^*(\mathbf{x}) - \xi)\Phi(Z) + \sigma_n^*(\mathbf{x})\phi(Z), & \sigma_n^* > 0 \\ 0 & \sigma_n^* = 0 \end{cases} \quad (4-13)$$

sendo  $\phi$  a função de densidade de probabilidade,  $\Phi$  a função cumulativa de probabilidades e  $Z$  o *z-score*, obtido por meio da seguinte expressão

$$Z = \frac{f(\mathbf{x}^+) - \mu_n^*(\mathbf{x}) - \xi}{\sigma_n^*(\mathbf{x})} \quad (4-14)$$

### 4.3.3

#### Melhoria Esperada Baseada em Hipervolume

Quando se trata de otimização multi-objetivo, como discutido na Seção 3.1, não há uma única solução para o problema de otimização e sim um conjunto de soluções que formam a fronteira de Pareto. Assim, a definição de *Improvement* utilizada para o caso tradicional de ME não é válida para o caso multi-objetivo. Visando resolver problemas com mais de um objetivo foi pro-



posto o Melhoria Esperada Baseada em Hipervolume  $ME_{\mathcal{H}}$  [48]. Nesse método, o novo fator de aprimoramento ( $I_{\mathcal{H}}(\mathbf{x})$ ) é definido como a diferença entre o hipervolume ( $\mathcal{H}$ ) calculado utilizando o *dataset* ( $D_n$ ) antes da iteração atual e o o hipervolume contabilizando o novo ponto da iteração atual ( $f(\mathbf{x}_{n+1})$ ), ou seja

$$I_{\mathcal{H}}(\mathbf{x}_{n+1} \mid D_n) = (\mathcal{H}(D_n \cup f(\mathbf{x}_{n+1})) - \mathcal{H}(D_n) \mid D_n) \quad (4-15)$$

Com a expressão de aprimoramento é possível escrever de maneira análoga a expressão para a nova função de aquisição  $EI_{\mathcal{H}}$  na sua forma de integral, lembrando que na forma integral  $\mathbb{E}[X] = \int_R x \cdot \phi(x) dx$  e que o  $EI$  é o valor esperado do aprimoramento  $I$ , i.e.

$$ME_{\mathcal{H}} = \int_R I_{\mathcal{H}}(\mathbf{x}_{n+1} \mid D_n) \cdot \phi(\mathbf{x}_{n+1}) d\mathbf{x}_{n+1} \quad (4-16)$$

Essa integral é então calculada numericamente para a resolução do problema de otimização.

#### 4.4

#### Otimização da função aquisição

Obtidas as funções de aquisição, que agora são funções analíticas, pode-se otimizá-las aplicando-se métodos que utilizam gradientes, como o Newton-Raphson ou BFGS. Newton-Raphson é um método de ordem 2, que usa o gradiente da função ( $\mathbf{g} = \nabla f(\mathbf{x})$ ) e sua matriz hessiana ( $\mathbf{H} = \nabla^2 f(\mathbf{x})$ ). Ambos são métodos de direção de busca, em que definem a direção de busca como o vetor  $\mathbf{d}$  e a partir dessa direção encontra-se o mínimo ao longo da reta  $\mathbf{x}(\alpha) = \mathbf{x} + \alpha \mathbf{d}$ .

No método Newton-Raphson a direção de busca é dada por:

$$\mathbf{d} = -(\mathbf{H})^{-1} \mathbf{g} \quad (4-17)$$

Entretanto, as operações para calcular a matriz Hessiana e sua inversa podem ser computacionalmente custosas o que, somado ao tempo de execução do próprio método e avaliação da função objetivo, pode não ser adequado para resolver o problema.

O método BFGS (Broyden–Fletcher–Goldfarb–Shanno) [57] é efetivamente de primeira ordem, entretanto se aproxima de um método de segunda ordem pois calcula de maneira iterativa uma aproximação da inversa da matriz Hessiana ( $\mathbf{S}$ ). A partir dessa nova matriz o processo de busca do mínimo é semelhante ao do método Newton-Raphson. Devido à essa característica é conhecido como Método Quase-Newton.

Visando adequar ainda mais o problema de custo computacional para que esta etapa seja menos custosa, uma prática comum em estudos de aprendizado

de máquina é utilizar uma variação do método BFGS conhecida como L-BFGS-B [58], que é utilizada neste trabalho. O prefixo indica que é a versão *Limited-memory BFGS* que efetivamente não altera a metodologia mas apenas a implementação. Nessa variação, ao invés de armazenar na memória do computador a matriz inteira de tamanho  $n \times n$ , sendo  $n$  o número de variáveis do problema, o método armazena vetores que representam essa matriz de maneira implícita. O sufixo do método indica que o problema é "*Bounded*", ou seja a busca pelo mínimo é feita com domínio restrito das variáveis.

Estabelecido o método utilizado, o ponto da iteração seguinte é obtido encontrando o conjunto de variáveis que maximizam a função de aquisição (Equação (4-18) usando EI como exemplo). A partir desse ponto o processo se reinicia, avaliando-o na função objetivo, atualizando o *dataset* e refazendo o GP para a próxima iteração adicionando o ponto seguinte, calculado como

$$x_{n+1} = \arg \max_{\mathbf{x}} ME(\mathbf{x}) = \mathbb{E} \left( \max\{0, f(\mathbf{x}^+) - f(\mathbf{x}) - \xi \mid \mathcal{D}_n\} \right) \quad (4-18)$$

## 4.5 Restrições

Apesar de o método tradicional se encerrar na seção anterior, para grande parte dos problemas de engenharia é necessário que o método lide com restrições. O método pode ser enriquecido por meio da técnica de Probabilidade de Viabilidade (PV) que lida com as restrições do tipo caixa-preta, a partir apenas das variáveis de entrada e os resultados de saída dessas restrições.

A técnica se originou de estudos realizados por Schonlau [59] e aprimorada por Gardner et al. [60] e Gelbart et al. [61].

A abordagem sugere duas modificações com relação ao processo original de otimização. Primeiramente, o melhor ponto  $\mathbf{x}^+$  passa a não ser apenas o ponto com melhor solução da função objetivo, mas sim aquele com a melhor solução e que satisfaz as restrições impostas. Segundo, é necessário alterar o valor da função de aquisição para zero para todas as soluções que violem as restrições. Essa segunda parte é realizada fazendo o produto de uma função de aquisição já existente por uma função de Heaveside ou função degrau  $H$ . Essa função degrau assume o valor 0 quando a restrição é violada e valor 1 quando for satisfeita. Se aplicada ao EI, por exemplo, a função de aquisição é descrita como:

$$ME_C(\mathbf{x}) = H(c(\mathbf{x}) \geq 0) \mathbb{E} \left[ \max 0, f(\mathbf{x}^+) - f(\mathbf{x}) - \xi \mid \mathcal{D}_n \right] = \Delta(\mathbf{x}) EI(\mathbf{x}) \quad (4-19)$$

Entretanto, quando se tratam de restrições caixa-preta, nem sempre é possível mapear exatamente o domínio com regiões válidas. Visando mapear as regiões de restrição é proposto então que se use uma estratégia como a utilizada na função objetivo, ou seja, modelar as restrições por meio de regressão por Processos Gaussianos utilizando um novo *dataset*, contendo os dados de restrição para cada iteração  $\mathcal{D}_n^c = \{c(\mathbf{x}_1), c(\mathbf{x}_2), \dots, c(\mathbf{x}_n)\}$ , em que:

$$c(\mathbf{x}) \sim \mathcal{GP}(m_c(\mathbf{x}), k_c(\mathbf{x}, \mathbf{x}')) \quad (4-20)$$

Utilizando essa nova abordagem da restrição considera-se então a restrição como uma variável aleatória, no caso uma variável aleatória de Bernoulli, que assume valor 1 quando satisfeita e 0 quando não. Então, a probabilidade da restrição ser satisfeita para um ponto qualquer é calculada por meio do valor esperado

$$\begin{aligned} \mathbb{E}[\Delta(\mathbf{x}) \mid \mathcal{D}_n^c] &= P[c(\mathbf{x}) \mid \mathcal{D}_n^c] = \int_0^1 p(c(\mathbf{x}) \mid \mathcal{D}_n^c) dc(\mathbf{x}) \\ &= \int_{-\infty}^{-\mu_n^c/\sqrt{2}\sigma_n^c} dz \frac{1}{\sqrt{\pi}} e^{-z^2} = \Phi\left[\frac{\mu_n^c}{\sqrt{2}\sigma_n^c}\right] = PF(\mathbf{x}) \end{aligned} \quad (4-21)$$

Acoplando esse termo à função de aquisição obtém-se a função de aquisição com restrição completa, expressa por

$$ME_C(\mathbf{x}) = \mathbb{E} \left[ \Delta(\mathbf{x}) \max\{0, f(\mathbf{x}^+) - f(\mathbf{x}) - \xi \mid \mathcal{D}_n, \mathcal{D}_n^c\} \right] = PF(\mathbf{x})EI(\mathbf{x}) \quad (4-22)$$

É possível expandir ainda a definição de PF para o caso de múltiplas restrições tratando cada uma delas separadamente, com sua própria regressão e depois acoplá-las fazendo-se o produto entre elas

$$PV(\mathbf{x}) = \prod_{l=1}^{n_c} P[c_l(\mathbf{x}) \geq 0 \mid \mathcal{D}_{l,n}^c] = \prod_{l=1}^{n_c} \Phi \left[ \frac{\mu_{l,n}^c(\mathbf{x})}{\sqrt{2}\sigma_{l,n}^c(\mathbf{x})} \right] \quad (4-23)$$

Essa abordagem é vital para a otimização de *risers* onde existem funções objetivos relativamente simples de serem calculadas e modeladas porém restrições bastante complexas, resultantes de uma análise dinâmica de elementos finitos, por exemplo.

## 5 Metodologia

Este capítulo será dedicado a detalhar a implementação de métodos e técnicas de paralelismo e metodologia utilizada para o estudo de caso.

Para a avaliação da função objetivo é necessária uma análise dinâmica de elementos finitos levando em consideração todas as informações de estado de mar. Foram realizados dois estudos de casos, visando abranger os diversos tipos de problemas de otimização que podem surgir para um projetista. No primeiro estudo a simulação foi realizada por meio do *software* comercial *Orcaflex* [49] enquanto no segundo foi utilizado o *software* de simulação da Petrobras ANFLEX [50]. Ambos realizam análises estáticas e dinâmicas de estruturas esbeltas *offshore* (linhas de ancoragem, *risers*, umbilicais, etc.) no domínio do tempo ou da frequência. A natureza do método de otimização ser caixa-preta permite que o *software* de simulação seja facilmente trocado sem nenhuma modificação no processo de otimização em si, apenas alterando a chamada do *software* de análise estrutural e o retorno dos valores.

A comunicação com o Orcaflex se deu por meio de uma API (*Application Programming Interface*) para *Python* [62], que permite a comunicação com o *software* e troca de dados. O valor das variáveis, correspondendo aos parâmetros da configuração de *riser*, são utilizados por meio da API para alterar o modelo, em seguida há um comando para rodar a simulação e os resultados da simulação são retornados para análise da função objetivo e restrições.

Já o ANFLEX não possui uma API direta de *Python* e então a chamada foi feita por meio de um script em Lua [63], já presente na interface do *software* de simulação da Petrobras. As variáveis escolhidas pela otimização são escritas em um arquivo que é então lido pelo *script* Lua. O modelo é alterado de acordo com essas variáveis, a simulação é feita e as informações relevantes ao problema são salvas em um arquivo. Em seguida, já no ambiente *Python* as informações necessárias são lidas e interpretadas, dando prosseguimento à otimização.

As diferentes variáveis e os resultados da simulação, tratando-se de dimensões de *risers* e análise de tensões, possuem ordens de grandezas bastante diferentes e podem impactar a qualidade da interpolação por Processos Gaussiano. Visando uma maior precisão e melhor comportamento do algoritmo em termos de convergência, todas as variáveis, função objetivo e restrições foram reescritas para ficarem sempre próximo ao intervalo  $[0,1]$ . Para o caso das variáveis o método gera pontos dentro deste intervalo e para que sejam convertidos

em valores compatíveis para o modelo da simulação é utilizada a Equação (5-1); sendo  $\bar{x}_i$  a variável normalizada,  $u_i$  o limite superior da variável e  $l_i$  o limite inferior da variável, i.e.

$$x_i = \bar{x}_i(u_i - l_i) + l_i \quad (5-1)$$

Em termos de função objetivo, para que ela se enquadre dentro do intervalo desejado é estimado um valor que seja factível com a solução esperada da otimização e a sua nova função objetivo será a razão entre o valor verdadeiro e esse novo valor estimado. As restrições serão readequadas de acordo com a seguinte equação

$$\begin{aligned} c_l(\mathbf{x}) \geq a &\rightarrow \bar{c}_l(\mathbf{x}) = \frac{c_l(\mathbf{x}) - a}{a} \geq 0 \\ c_l(\mathbf{x}) \leq a &\rightarrow \bar{c}_l(\mathbf{x}) = \frac{a - c_l(\mathbf{x})}{a} \geq 0 \end{aligned} \quad (5-2)$$

A aplicação de Otimização Bayesiana foi realizada utilizando a biblioteca *opensource GPyOpt* [64] em *Python* e foi desenvolvida utilizando *Jupyter notebooks* [65]. A biblioteca originalmente não permitia resolver problemas com restrições do tipo caixa-preta e solucionava apenas problemas mono-objetivos e portanto foram desenvolvidos códigos capazes de solucionar essas duas deficiências, a técnica de *probability of feasibility*, discutida na Seção 4.5 e Melhoria Esperada baseada em Hipervolume, discutida na Seção 4.3.3.

A função de aquisição utilizada foi a Melhoria Esperada em que há um balanceamento maior entre a exploração do domínio e a busca do mínimo global, com o valor do *jitter* fixado em  $\xi = 2,5 \cdot 10^{-4}$ . O *kernel* utilizado para a interpolação, calculado internamente no algoritmo, foi a função de covariância Matérn-5/2, que é *kernel* padrão da biblioteca *GPyOpt*, e expresso pela com a Equação (5-3). Foram testados outros dois *kernels*, o exponencial  $k(x, x') = \sigma^2 \exp(-r/l)$  e o Gaussiano  $k = \sigma^2 \exp(-r^2/2l^2)$ , porém, o Matérn-5/2 foi o que obteve o melhor desempenho, ou seja

$$k(x, x') = \sigma^2 \left( 1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2} \right) \exp \left( -\frac{\sqrt{5}r}{l} \right) \quad (5-3)$$

sendo  $r = \sqrt{(x - x')^T (x - x')}$  a distância Euclidiana entre os dois pontos,  $\sigma^2$  o desvio padrão e  $l$  o comprimento característico, definido pelo usuário. Foram testados diferentes valores para o comprimento característico  $l \in 1, 1/2, 1/3, 1/10$  sem muita diferença no resultado final da otimização.

Na prática verificou-se que, apesar da técnica PV solucionar e direcionar para regiões que não violam restrições, se adicionarmos ao problema uma técnica de penalização quadrática, como aquela apresentada na Seção 3.3,

o processo é acelerado e mais consistente. Então, para esse trabalho, como estratégia de violação de restrições foram utilizadas a técnica PV e penalização quadrática em conjunto.

A fim de avaliar a eficiência da Otimização Bayesiana foram escolhidos adicionalmente três outros métodos para fazer uma comparação: o algoritmo genético (GA), *Globalized Bounded Nelder-Mead* (GBNM) [66] e MIDACO (Mixed Integer Distributed Ant Colony Optimization) [67].

O algoritmo genético, um método popular e já testado em otimização de *risers* [19], foi implementado utilizando a biblioteca em *Python* DEAP (*Distributed Evolutionary Algorithms in Python*) [68]. Os algoritmos genéticos se baseiam no conceito de Charles Darwin em que os indivíduos mais bem adaptados terão maior chance de sobreviver e gerar descendentes. Inicialmente uma população é gerada por indivíduos aleatoriamente e cada indivíduo expressa um ponto avaliado da função objetivo. Esses indivíduos são então comparados de acordo com sua função de aptidão e por meio da metodologia do torneio são divididos em trios em que o mais apto de cada trio é selecionado. O conjunto de indivíduos mais aptos dão origem a uma nova geração de indivíduos com características dos pais e possibilidade de mutações. O processo é repetido até a solução ser encontrada, o indivíduo mais apto ao final é a solução do problema.

Como parâmetros para esse estudo, a população foi definida com 200 indivíduos ao longo de 35 gerações, baseado no trabalho de Vieira et al. [20]. O número de torneios realizados é o mesmo que o número de indivíduos, 200 torneios que dão origem a 200 acasalamientos. A probabilidade de cruzamento é de 80% e com 10% de probabilidade de mutação. A função de aptidão utilizada é a função objetivo com o método do Lagrangeano aumentado [51]. A otimização se encerra quando o desvio padrão das aptidões da população for menor que 1 ou após as 35 gerações.

O método GBNM é uma generalização do método de Nelder-Mead [69], um algoritmo baseado em simplex para otimização local, para solucionar problemas de otimização global. Essa generalização é realizada a partir de reinícios probabilísticos resultando em múltiplas buscas locais iniciadas em diversos pontos do domínio. O simplex que resultar no melhor resultado é considerado o "mínimo global". O simplex é uma figura geométrica de  $n+1$  dimensões, sendo  $n$  o número de variáveis do problema, e cada vértice expressa uma avaliação da função objetivo. O vértice de maior valor da função objetivo é substituído por meio de operações como mostra a Figura 5.1. Para lidar com as restrições foi utilizado o método do Lagrangeano Aumentado. O algoritmo utilizado foi desenvolvido em *Python* e baseado no trabalho Cardoso et al. [2].

Para esse estudo foram realizados 20 reinícios probabilísticos e o parâmetro inicial de Lagrange e de penalidade estabelecidos como  $\lambda = 1000$  e  $\mu = 1000$ .

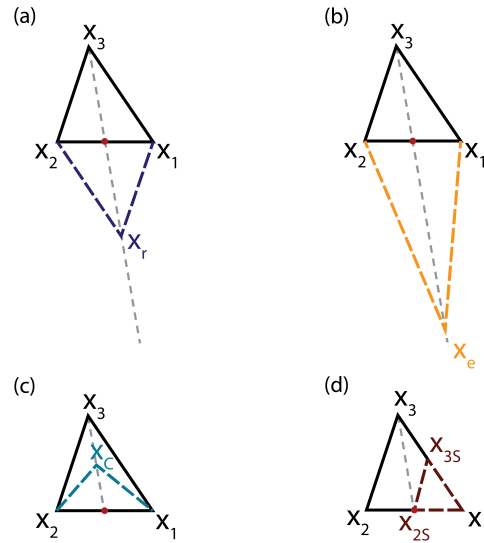


Figura 5.1: Operações de simplex do Nelder-Mead

O MIDACO é um código comercial, que exige licenças para sua utilização, desenvolvido na linguagem C para resolver problemas de programação não linear envolvendo variáveis discretas e contínuas (*Mixed Integer Nonlinear Programming* ou MINLP) para aplicação na indústria espacial [70, 71]. O código implementa uma versão estendida do método da Colônia de Formigas para busca em domínios discretos e contínuos [72] e o método de penalidade Oracle para lidar com restrições [73].

O método encerra a otimização quando se passarem 100 iterações sem melhora com relação ao melhor ponto já avaliado. Por padrão, o MIDACO considera viáveis soluções que violam até  $10^{-3}$  das restrições. Neste trabalho, esse valor foi alterado para  $10^{-7}$ , visando uma melhor acurácia ao método, já que algumas das restrições podem ser bastante sensíveis. Os demais parâmetros foram mantidos e podem ser encontrados no manual de usuário [74].

## 5.1 Paralelismo

Durante o processo de otimização a etapa mais custosa computacionalmente é a análise dinâmica pelo método elementos finitos. A maneira mais fácil e intuitiva de realizar essa etapa é sequencialmente, uma iteração após a outra. Existem, no entanto estratégias de paralelização em que é possível realizar múltiplas análises simultaneamente, reduzindo o tempo total de execução

drasticamente. Para a Otimização Bayesiana a estratégia escolhida foi a do *Consumer-Producer* (ou Consumidor-Produtor) [75].

Na estrutura mais usual as tarefas do processo são divididas entre nós classificados como produtores e outros como consumidores. Em geral há apenas um produtor, que será encarregado de coordenar a geração de pontos e múltiplos consumidores, que serão responsáveis por executar análises da função objetivo simultaneamente. Esse tipo é ideal para problemas em que a função objetivo é mais custosa computacionalmente que a geração do ponto seguinte, como é o caso do problema de otimização da configuração de *riser*.

Como o processo de otimização é dividido entre a Inicialização e a Otimização, haverá dois produtores, um para cada etapa. O produtor da inicialização será responsável pelo PE, nesse caso utilizando a técnica de amostragem Hipercubo Latino, enviando os pontos gerados para os consumidores e recebendo o resultado da avaliação da função objetivo. Terminada a inicialização o segundo produtor será ativado, responsável por todo o processo de otimização; realização da regressão por meio de Processos Gaussianos, cálculo da função de aquisição e otimização da mesma. Quando obtiver o ponto, o mesmo será enviado ao consumidor que durante todo o processo apenas receberá pontos do produtor, executará a função objetivo e retornará os dados relevantes à otimização para que o processo seja reiniciado. O processo é resumido na Figura 5.2

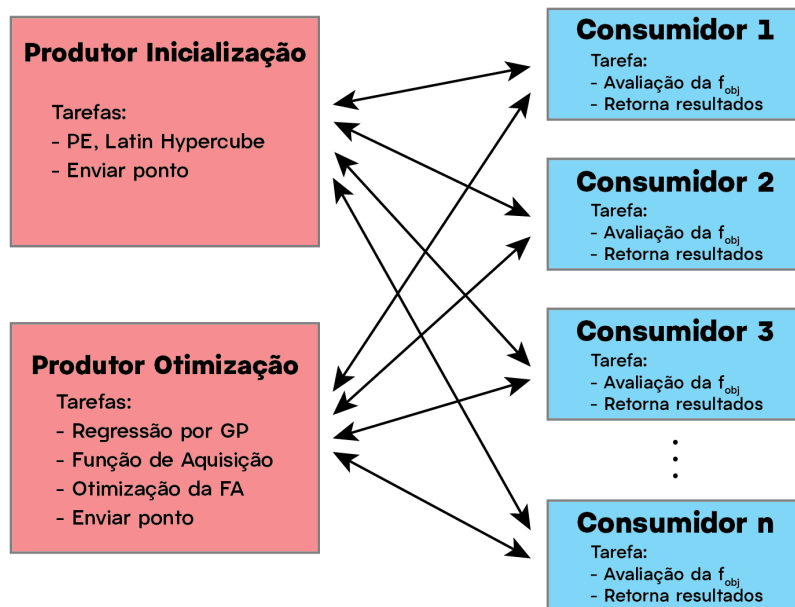


Figura 5.2: Modelo de paralelismo Consumidor-Produtor

A paralelização foi realizada de maneira assíncrona, i.e., cada consumidor executa as tarefas independentemente dos demais. Todos os pontos gerados



pelos produtores são introduzidos em uma fila de pontos a serem executados e a medida que os consumidores terminarem suas tarefas consultam essa fila e, se houver algum ponto na fila, já o executam em seguida. Esse estilo de paralelização evita que os consumidores fiquem muito tempo ociosos, esperando os demais terminarem suas tarefas, já que o tempo de análise não é constante.

Foi investigada a possibilidade de fazer um paralelismo de maneira síncrona em que o produtor gera de uma vez pontos para todos os consumidores e só seguem para a etapa seguinte assim que todos os consumidores terminarem suas tarefas. Uma das vantagens é que, em vez de repetir o processo de otimização (interpolação, função de aquisição e sua otimização) a cada lote, esse processo é realizado uma vez a cada conjunto de pontos gerados. No caso assíncrono, todo esse processo é repetido a cada iteração, pois cada ponto é gerado individualmente. A desvantagem que impulsionou a escolha do modelo assíncrono foi a perda de eficiência pois a atualização do modelo e escolha do ponto é feita apenas a cada lote. Como a análise da função objetivo é mais custosa computacionalmente que a geração do ponto seguinte, a necessidade de menos iterações do método assíncrono o tornou mais vantajoso.

## 6

### Estudo de caso

#### 6.1

##### Caso 1 - RCHA e SLWR mono-objetivo

O primeiro estudo de caso consiste na realização de uma otimização mono-objetivo utilizando o método de otimização Bayesiana em casos realistas de *risers* e na avaliação do desempenho dos mesmos comparando com outros encontrados na literatura tendo como principal parâmetro de comparação o número de avaliações da função objetivo. Ou seja, os métodos serão classificados, principalmente, de acordo com o número total de avaliações da função objetivo necessárias para encontrar consistentemente a solução do problema. Esse parâmetro foi escolhido pois, para cada avaliação da função objetivo, é necessário rodar uma análise dinâmica de elementos finitos testando o ponto definido pelo método, fazendo com que o tempo total de uma iteração seja dominado por essa análise. Dessa forma, reduzindo o número de análises pode-se obter um menor tempo total de execução do método de otimização.

Esse exemplo trata de um caso cujo único objetivo é minimizar o comprimento total do *riser* e compara o método de otimização Bayesiana com outros métodos como o código comercial MIDACO, globalized bounded Nelder-Mead (GBNM) e Algoritmo Genético (GA). Todos os métodos foram aplicados para três casos realistas distintos, sendo dois deles aplicados à configuração SLWR e o terceiro aplicado a uma configuração RCHA. Apesar de todos terem a mesma função objetivo, as variáveis do problema e as restrições para cada configuração são diferentes.

Em todos os modelos as propriedades do *riser* rígido são as mesmas e estão listadas na Tabela 6.1. As características do material são definidas pelo módulo de Young, peso específico, tensão nominal de escoamento SMYS (em inglês *Specified Minimum Yield Strength*) e tensão nominal de ruptura SMTS (em inglês *Specified Minimum Tensile Strength*).

Diâmetro interno	0,165 m
Diâmetro externo	0,212 m
Espessura do revestimento	0,081 m
Módulo de Young	207 GPa
Peso específico	7728 kg/m <sup>3</sup>
SMYS	482,6 MPa
SMTS	565,4 MPa

Tabela 6.1: Propriedades físicas dos *risers*

Em todos os casos, os carregamentos aplicados foram os que produziram os piores resultados em termos de tensões distribuídas ao longo do *riser* dentre todas as combinações de ondas e correntes. A corrente aplicada apresenta um perfil de velocidade triangular tendo o máximo de 1.15 m/s próximo a superfície. O movimento de topo do *riser*, imposto pelo FPSO, é apresentado na Tabela 6.2. O tempo de simulação é equivalente a dois períodos do movimento com uma rampa de um período do movimento, totalizando três períodos (37,23s). A Figura 6.1 apresenta a representação visual dos 3 graus de liberdade do topo, sendo três de translação e três de rotação.

Movimento	Amplitude	Período (s)	Fase (°)
<i>Surge</i>	0,81 m	12,41	53,73
<i>Sway</i>	2,97 m	12,41	70,12
<i>Heave</i>	8,97 m	12,41	321,04
<i>Roll</i>	5,51°	12,41	287,88
<i>Pitch</i>	1,01°	12,41	257,78
<i>Yaw</i>	0,85°	12,41	177,44

Tabela 6.2: Movimento de topo do *riser*, imposto pelo FPSO

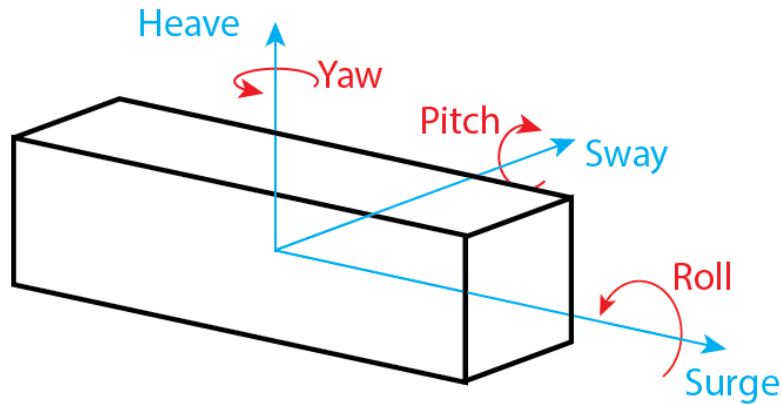


Figura 6.1: Representação visual dos seis graus de liberdade, três de rotação e três de translação

Nas próximas seções serão descritos com mais detalhes os modelos utilizados.

### 6.1.1

#### ***Steel Lazy Wave Catenary (SLWR)***

Como a configuração Lazy Wave é uma das mais utilizadas na indústria, principalmente quando se trata de regiões com lâmina d'água profunda, nesse caso serão tratados dois modelos dessa configuração com lâminas d'água distintas, sendo a mais profunda com 2150 *m* e a mais rasa 1200 *m* e com a mesma projeção horizontal de 2861 *m*. As grandezas que definem o *riser*, e que também são as variáveis do problema de otimização, são os comprimentos do trecho antes dos flutuadores (L1), do trecho central com os flutuadores (L2) e do trecho final depois dos flutuadores (L3), as características dos flutuadores como o diâmetro ( $D_f$ ), comprimento ( $L_f$ ) e o espaçamento entre eles (SP) e o ângulo de topo ( $\alpha$ ). O ângulo de topo é levado em consideração como uma restrição de projeto, sendo mais detalhada na seção 6.1.4. Todas as variáveis estão apresentadas na Figura 6.2 e possuem um domínio delimitado como apresentado na Tabela 6.3, válidos para ambos os modelos.

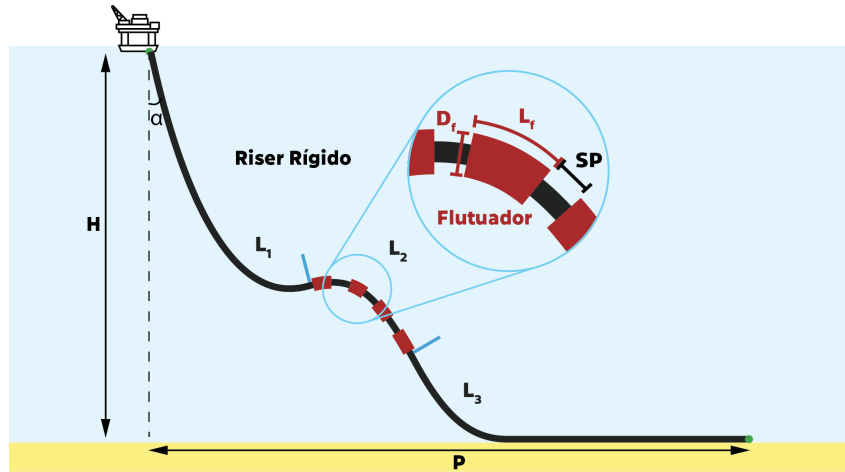


Figura 6.2: Configuração SLWR

Variável	Mínimo	Máximo
$L_1$	700 m	3000 m
$L_2$	300 m	1600 m
$L_3$	700 m	3000 m
$D_f$	0,375 m	2,0 m
$L_f$	0 m	2 m
$SP$	1 m	3 m

Tabela 6.3: Variáveis e os intervalos investigados representados por seus máximos e mínimos

O limite inferior do comprimento do flutuador apresentado na Tabela 6.3 sendo zero permite que o resultado final seja uma configuração sem a presença de flutuadores, caso essa seja a configuração que impõe os melhores resultados. A ausência dos flutuadores, se houver, faz com que a configuração seja uma Catenária Livre. Os flutuadores possuem um peso específico de  $750 \text{ kg/m}^3$ .

### 6.1.2

#### **Riser em Catenária Hidroamortecido (RCHA)**

A possibilidade de se ter uma configuração que se aproxima de uma catenária livre trás grande importância a esse tipo de configuração, menos complexa que uma SLWR e que, por isso, também será alvo desse estudo. O caso estudado possui uma lâmina d'água de  $2150 \text{ m}$ , uma projeção horizontal de  $2861 \text{ m}$  e as variáveis utilizadas para definir a configuração são o ângulo de topo ( $\alpha$ ) e os comprimentos de cada trecho do riser, da parte superior ( $L_1$ ),

das partes com amortecedores ( $L2$ ,  $L4$ ,  $L6$ ,  $L8$  e  $L10$ , do espaçamento entre os amortecedores ( $L3$ ,  $L5$ ,  $L7$  e  $L9$ ) e da parte final ( $L11$ ). Estudos anteriores ([8]) indicam que cinco amortecedores é a quantidade ideal de nesse tipo de configuração. O domínio de cada variável está apresentado na Tabela 6.4 e as propriedades dos amortecedores hidrodinâmicos estão na Tabela 6.5.

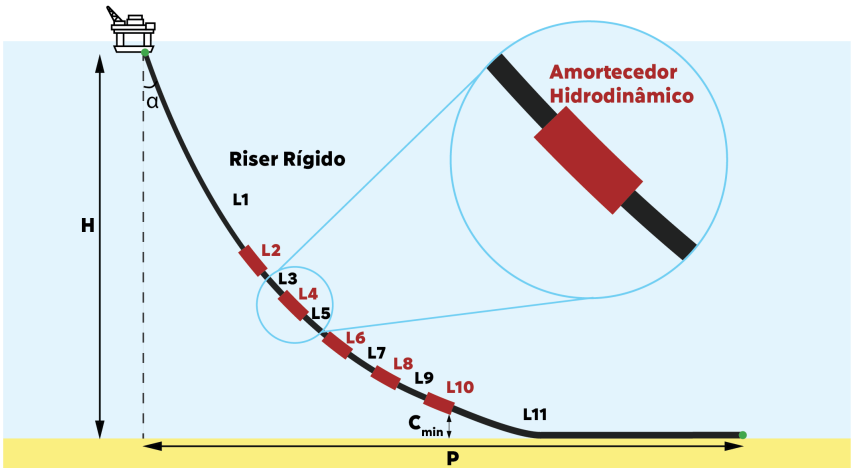


Figura 6.3: Configuração RCHA

Variável	Mínimo	Máximo
$L1$	1000 m	2050 m
$L2$	12 m	240 m
$L3$	12 m	240 m
$L4$	12 m	240 m
$L5$	12 m	240 m
$L6$	12 m	240 m
$L7$	12 m	240 m
$L8$	12 m	240 m
$L9$	12 m	240 m
$L10$	12 m	240 m
$L11$	1000 m	2050 m

Tabela 6.4: Variáveis RCHA e seus limites

Densidade do HA	$738kg/m$
Espessura do HA	$0,195m$

Tabela 6.5: Propriedades dos Hidroamortecedores

### 6.1.3

#### Função objetivo

Quando se trata de configurações de *riser*, é importante minimizar o comprimento já que está diretamente relacionado ao seu custo. Dessa maneira, pode-se descrever matematicamente a função de acordo com a equação:

$$f(\mathbf{x}) = \sum_{i=1}^{n_L} (L_i) \quad (6-1)$$

em que  $n_L$  é o número de segmentos que compõem o *riser* e  $L_i$  o comprimento do segmento correspondente. Para o caso RCHA a função objetivo corresponde à soma de todas as variáveis do problema exceto ângulo de topo, enquanto que, para o caso SLWR, é a soma das três primeiras variáveis apenas, que correspondem aos comprimentos das seções.

Apesar de ser um objetivo simples de minimizar, a dificuldade do problema está nas restrições. A minimização do comprimento pode fazer com que as tensões se elevem de maneira a violarem algum critério de tensão. Dessa maneira o foco do problema está em como lidar com as restrições durante o problema de otimização.

### 6.1.4

#### Restrições

Para a análise e verificação de *risers*, existe uma série de critérios estruturais e operacionais que deverão ser avaliados durante as simulações pelo método dos elementos finitos, para que uma estrutura seja considerada adequada para aquelas condições. Em ambos os casos, há uma limitação dada pelo ângulo de topo do *riser*, podendo ter uma variação de 6° a 20°.

Como garantia da integridade estrutural, foi utilizada como restrição o Fator de utilização da seção transversal (UF) definida pela norma DNVGL-ST-F201 [76] em que combina a ação de diferentes tipos de carregamentos e coeficientes de segurança. Para se garantir a integridade, o *riser* deverá satisfazer às seguintes expressões

$$\left\{ \begin{array}{l} \gamma_{SC}\gamma_m \left[ \frac{|M_d|}{M_k} \sqrt{1 - \left( \frac{p_i - p_e}{p_b} \right)^2} + \left( \frac{T_e}{T_k} \right)^2 \right] + \left( \frac{p_i - p_e}{p_b} \right)^2 \leq 1 \text{ se } p_i > p_e \\ (\gamma_{SC}\gamma_m)^2 \left\{ \left[ \frac{|M_d|}{M_k} + \left( \frac{T_e}{T_k} \right)^2 \right]^2 + \left( \frac{p_e - p_{min}}{p_c} \right)^2 \right\} \leq 1 \text{ se } p_i \leq p_o \end{array} \right. \quad (6-2)$$

onde

–  $\gamma_{SC}$  = Fator para classe de segurança do material

- $\gamma_m$  = Fator de resistência do material
- $M_d$  = Momento fletor
- $M_k$  = Resistência ao momento fletor plástico
- $T_e$  = Tensão efetiva
- $T_k$  = Resistência à força axial plástica
- $p_i$  = Pressão interna
- $p_e$  = Pressão externa
- $p_b$  = Pressão de resistência ao colapso da parede (*busting*)
- $p_e$  = Pressão de resistência à flambagem circunferencial (*hoop buckling*)
- $p_{min}$  = Pressão interna mínima

A Tabela 6.6 resume as restrições impostas ao modelo de SLWR.

Ângulo de topo máx	20°
Ângulo de topo min	6°
UF máx	1

Tabela 6.6: Restrições para o modelo SLWR

Para o caso RCHA duas restrições adicionais foram impostas. O comprimento total do *riser* não deve exceder a soma da lâmina d'água com a projeção horizontal entre o nó inicial e final do *riser* e a distância vertical entre o último hidroamortecedor e o fundo do mar deve ter um valor mínimo de 10 metros. Essa última restrição é adicionada para evitar que haja contato entre o hidroamortecedor e o fundo do mar, podendo danificar a estrutura. O resumo das restrições para o caso RCHA pode ser encontrado na Tabela 6.7.

Ângulo de topo máx	20
Ângulo de topo min	6
UF máx	1
Comprimento max do <i>riser</i>	5011 m
Distância máx entre solo e HA	10 m

Tabela 6.7: Restrições para o modelo RCHA



## 6.1.5

## Resultados

Os resultados obtidos a partir da aplicação de cada método estudado estão resumidos na Tabela 6.8 que mostra o melhor resultado de cada otimização. Para o número de avaliações da função objetivo do BO, o número total de avaliações corresponde a soma do número de avaliações da inicialização com o número de avaliações do loop de otimização, descritos individualmente entre parênteses.

<b>SLWR LDA 1200 m</b>	<b>N avaliações</b>	<b>UF</b>	<b><math>\theta_{max}</math></b>	<b>Comp. mín.</b>
BO	115 (50+65)	0,89	19,77	3555 m
MIDACO	280	0,92	12,31	3649 m
GBNM	1569	0,22	18,15	3700 m
GA	3844	0,26	20,54	3673 m
<b>SLWR LDA 2150 m</b>	<b>N avaliações</b>	<b>UF</b>	<b><math>\theta_{max}</math></b>	<b>Comp. mín.</b>
BO	115 (50+65)	0,98	19,94	3949 m
MIDACO	310	0,58	19,99	4012 m
GBNM	1104	0,36	18,55	4152 m
GA	4004	0,37	20,42	4045 m
<b>RCHA LDA 2150 m</b>	<b>N avaliações</b>	<b>UF</b>	<b><math>\theta_{max}</math></b>	<b>Comp. mín.</b>
BO	350 (150 + 200)	0,97	19,73	4118 m
MIDACO	240	0,95	19,91	4138 m
GBNM	2296	1,00	19,92	4113 m
GA	5958	0,67	18,54	4169 m

Tabela 6.8: Resultados para o caso 1. Para o BO, o número de avaliações corresponde à soma de iterações da inicialização e do loop de otimização

Observando o número de avaliações da função objetivo de cada caso na Tabela 6.8, apesar do método GA ser um algoritmo bastante utilizado na indústria, ele foi o método que necessitou do maior número de avaliações. Esse fato não necessariamente se reflete numa melhor solução, inclusive em dois casos não foi possível respeitar a restrição de ângulo de topo. O BO se mostrou mais eficiente em relação ao número de avaliações nos dois casos de SLWR e necessitou de algumas avaliações a mais que o MIDACO no caso RCHA. O fato de o BO utilizado ser um código *opensource* faz com que ele seja mais acessível que o MIDACO, que exige uma licença para sua utilização.

Levando-se em consideração apenas a qualidade do resultado em todos os casos, apesar de valores próximos, o BO foi o método que obteve o menor

comprimento total de *risers*, satisfazendo as restrições impostas. A qualidade do resultado somado ao fato de seu resultado exigir uma ordem de grandeza a menos de iterações fazem com que o BO se destaque perante ao método GA.

As Figuras 6.4, 6.5 e 6.6 apresentam uma visualização da melhor configuração de cada método para cada um dos modelos, SLWR com LDA 1200m, SLWR com LDA 2150m e RCHA com LDA 2150m. Essa representação permite observar que, apesar de em alguns casos os comprimentos serem semelhantes, podem levar a diferentes soluções.

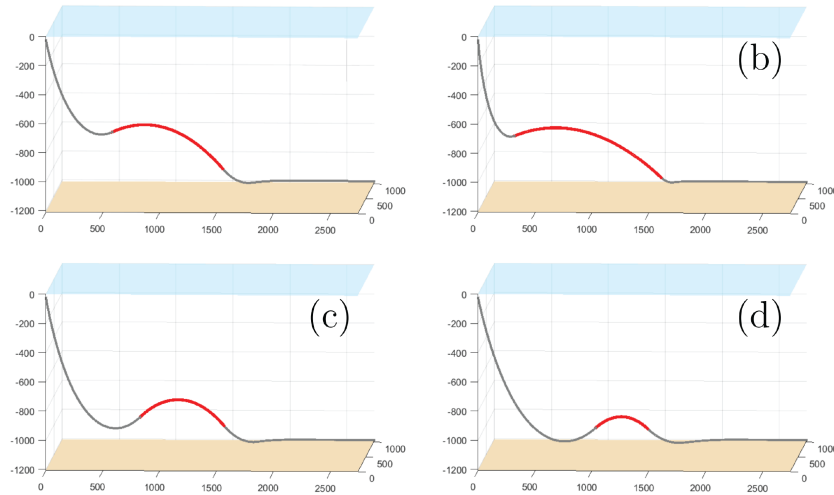


Figura 6.4: Configuração ótima de cada método para SLWR com LDA 1200 m. (a) BO, (b) MIDACO, (c) GBNM e (d) GA

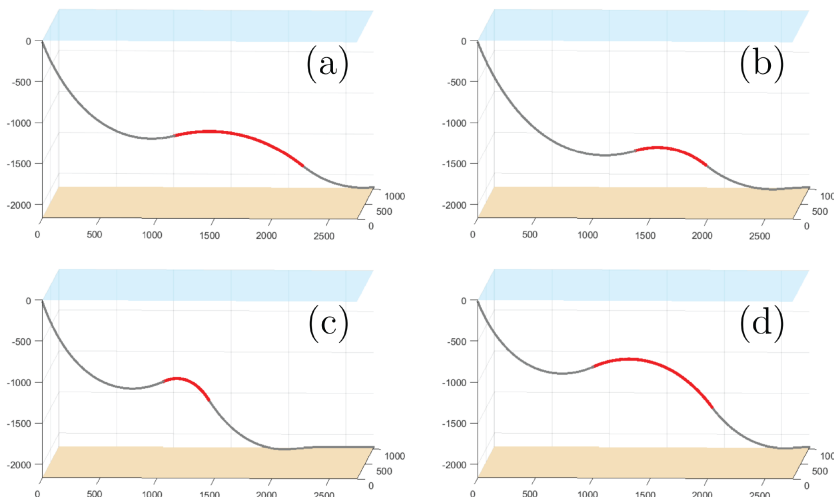


Figura 6.5: Configuração ótima de cada método para SLWR com LDA 2150 m. (a) BO, (b) MIDACO, (c) GBNM e (d) GA

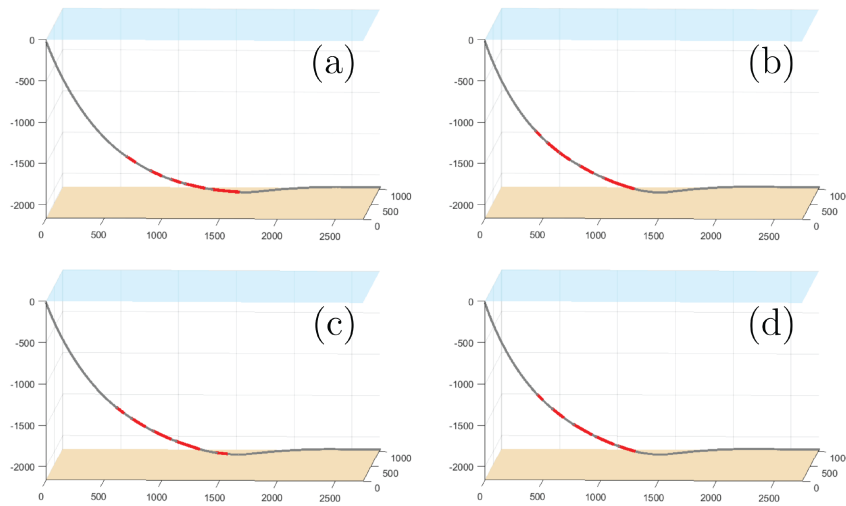


Figura 6.6: Configuração ótima de cada método para RCHA. (a) BO, (b) MIDACO, (c) GBNM e (d) GA

A Figura 6.7 nos apresenta o progresso do melhor ponto durante o processo de otimização. As Figuras 6.7 a), b) e c) são referentes ao método de BO e cada curva representa uma rodada do método, totalizando cinco rodadas de cada caso, enquanto que para o método MIDACO na Figura 6.7 d) foi rodada uma vez apenas para cada caso e cada curva representa um caso diferente. É importante observar que, em todos os casos utilizando BO as diferentes rodadas convergem para resultados próximos, mostrando a consistência do método. Para o MIDACO as soluções também convergem com poucas iterações. O melhor valor da solução para a configuração RCHA no MIDACO eventualmente assume um valor maior pois as configurações inicialmente testadas violavam alguma restrição e é considerado que qualquer configuração que não viole restrições é melhor que uma que viole restrições. Eventualmente a solução convergiu para um mínimo.

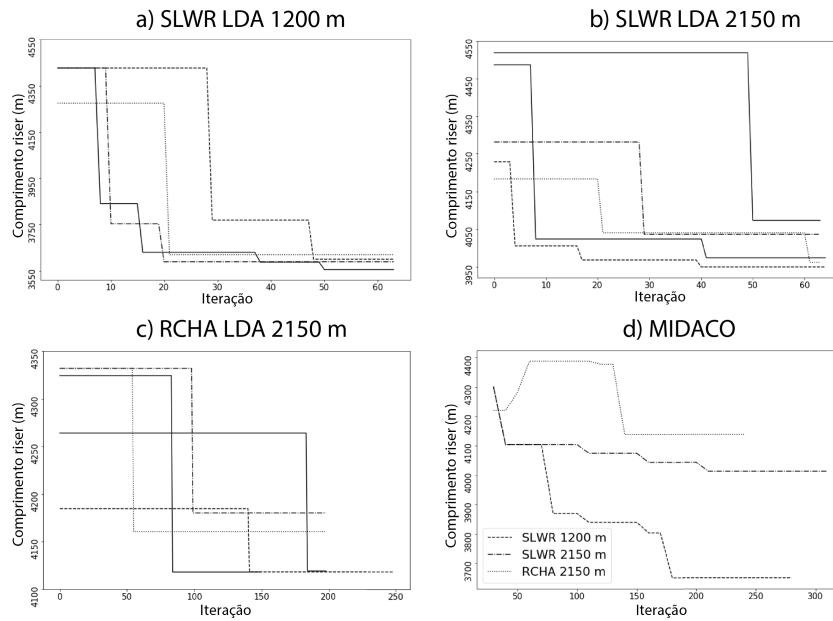


Figura 6.7: Evolução da melhor solução, BO x MIDACO

Fazendo uma comparação de tempo de execução entre os métodos MIDACO e BO mostrados na Figura 6.8, que são os dois métodos mais competitivos em termos de avaliações da função objetivo, observa-se uma vantagem significativa do BO. O tempo de proposta de um novo ponto representa o tempo gasto para realizar as etapas necessárias para geração do ponto (PG, obtenção da FA e otimização da mesma) e de avaliação o tempo da avaliação da função objetivo, nesse caso análise de elementos finitos usando o Orcaflex. Como observado na Figura 6.8 (a), para os dois casos SLWR, que são os mais usuais atualmente, o tempo de execução do BO quando executado serialmente é equivalente ao tempo gasto pelo MIDACO executado paralelamente. No modelo RCHA o método possui um desempenho um pouco inferior, porém mesmo de maneira serial o tempo de execução é viável e com a paralelização, ele pode ser ainda mais reduzido.

A Figura (b) apresenta a evolução do tempo médio da proposta de cada ponto e tempo de avaliação da função objetivo. O tempo de proposta do próximo ponto possui uma tendência de aumentar com o número de pontos avaliados pois o processo gaussiano envolve a inversão de uma matriz cuja dimensão é o número de pontos avaliados. Esse processo é computacionalmente custoso e por isso tende a aumentar. Como nesse caso o número de avaliações é baixo não chega a ser um problema.

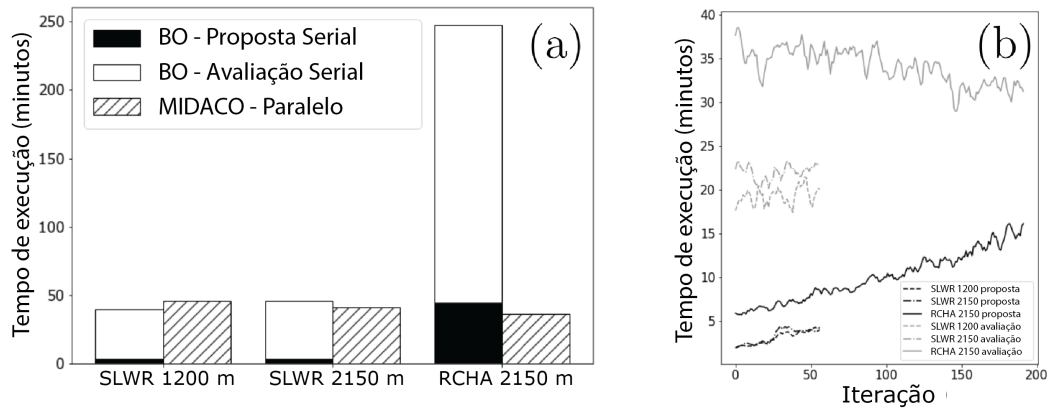


Figura 6.8: (a) Tempo médio de execução, para o BO o trecho em preto é o tempo da proposta do próximo ponto e em branco da avaliação da função objetivo; (b) Evolução do tempo médio da proposta de cada ponto e tempo de avaliação da função objetivo

## 6.2

### Caso 2 - SLWR com três casos de carregamento

O objetivo deste segundo estudo de caso é verificar e validar a implementação do método para lidar com um problema de complexidade maior com três casos de carregamento (LC) a serem avaliados simultaneamente e com três tipos de fluido interno (com água, vazio e peso intermediário, caracterizado nas tabelas como médio) e posteriormente validar a implementação multi-objetiva do método. Entretanto, nem todos os carregamentos foram simulados com todos os fluidos internos. Ao todo foram quatro simulações que estão resumidas na Tabela 6.9. Todas as simulações para esse caso foram realizadas utilizando o *software* de elementos finitos da Petrobras ANFLEX.

	Médio	Água	Vazio
LC1			
LC2			
LC3			

Tabela 6.9: Resumo de simulações realizadas, em verde combinações que foram simuladas e em vermelho combinações que não foram simuladas

Para todos os casos de carregamento foram aplicadas ondas irregulares com espectro Jonswap. As informações sobre a onda de cada caso de carregamento estão resumidas na Tabela 6.10 e as informações sobre as correntes na Tabela 6.11

	LC1	LC2	LC3
Período da onda (s)	13	12	17,5
Altura significativa (m)	7,9	6,5	10,9
Azimuth (°)	67,5	270	45

Tabela 6.10: Dados de ondas de cada caso de carregamento

LC1			LC2			LC3		
H(m)	V(m/s)	D	H(m)	V(m/s)	D	H(m)	V(m/s)	D
0	1.04	ENE	0	1.32	W	0	0.34	ENE
50	1.03	ENE	50	1.31	W	100	0.36	ENE
100	0.81	ENE	100	1.15	W	150	0.39	ENE
150	0.73	ENE	150	1.01	W	200	0.42	ENE
200	0.65	ENE	200	0.80	W	250	0.44	ENE
250	0.58	ENE	250	0.75	W	300	0.40	ENE
300	0.51	ENE	300	0.68	W	350	0.39	ENE
350	0.47	ENE	350	0.61	W	375	0.37	ENE
375	0.44	ENE	375	0.59	W	800	0.36	NE
800	0.32	ENE	800	0.35	WSW	1200	0.29	NE
1200	0.25	ENE	1200	0.22	WSW	1600	0.30	E
1600	0.26	ENE	1600	0.21	SW	2000	0.26	E
2000	0.27	ENE	2000	0.20	SSW	2200	0.36	E
2200	0.25	ENE	2200	0.20	SSW			

Tabela 6.11: Dados de corrente para cada caso de carregamento, sendo H a profundidade, V a velocidade e D e direção.

O estudo consiste em duas etapas independentes sendo a primeira um estudo mono-objetivo e a segunda visando também validar a implementação do método multi-objetivo. O estudo mono-objetivo será ainda subdividido em duas partes, sendo a primeira utilizando variáveis contínuas e em seguida utilizando variáveis discretas. O uso de variáveis discretas é importante para o momento em que o projeto passa da simulação para o campo prático. Não é possível implementar na prática medidas tão precisas quanto as provenientes das variáveis contínuas. Há algumas restrições de projeto que requerem, por exemplo, que os *risers* rígidos sejam produzidos em módulos de 12 m de comprimento. É possível realizar aproximações a partir das variáveis contínuas

para encaixar a realidade porém já trabalhando com variáveis discretas agiliza o processo e garante que as restrições estão sendo satisfeitas.

O modelo base para esse segundo estudo é de umma configuração Lazy-Wave (SLWR) numa lâmina d'água de 2200 m de profundidade. As propriedades do *riser* podem ser resumidas na Tabela 6.12.

Diâmetro externo	435,60 mm
Diâmetro interno	355,60 mm
<b>Propriedade do material</b>	
Peso específico	77,66 kN/m <sup>3</sup>
Limite de escoamento (SMYS)	414 MPa
<b>Flutuadores</b>	
Peso específico	6 kN/m <sup>3</sup>

Tabela 6.12: Propriedades do *riser*

### 6.2.1 Variáveis

As variáveis para esse problema são praticamente as mesmas do caso anterior, os três comprimentos dos trechos superior, com flutuadores e inferior, as propriedades do flutuadores (como diâmetro comprimento e espaçamento) e a inclusão do ângulo de topo. O ângulo de topo pode ser tratado como uma restrição, como no caso 1 estudado ou ser implementado como uma variável de projeto como no caso atual. A Tabela 6.13 resume as variáveis, seus limites superiores e inferiores e o incremento de cada, para o caso discreto.

	Mínimo	Máximo	Incremento
$\alpha$	6	8	0,5
$L_1$ (m)	500	3000	12
$L_2$ (m)	100	1500	12
$L_3$ (m)	100	2000	12
$D_f$ (m)	2	3	0,05
$L_f$ (m)	2	3	0,05
$SP$ (m)	12	36	12

Tabela 6.13: Variáveis, seus valores máximos, mínimos e incrementos, para o caso discreto

### 6.2.2

#### Função Objetivo

O primeiro estudo envolvendo o modelo foi realizado considerando o problema mono-objetivo em que o objetivo é minimizar a projeção horizontal do trecho suspenso do *riser*. Esse objetivo se assemelha ao anterior em que de maneira geral ambos os objetivos estão relacionados, visam diminuir o comprimento, que representa também uma diminuição do custo. Neste caso específico, apesar de não ser direto, também estabelece que a embarcação estará mais próxima do ponto de perfuração, o que apresenta algumas vantagens. Diferentemente do objetivo anterior, esse só é possível de ser obtido na simulação e não é possível obter o valor do objetivo diretamente pelas variáveis.

O segundo estudo foi realizando considerando o problema multi-objetivo com dois objetivos. O primeiro objetivo é o mesmo do problema mono-objetivo, minimizar a projeção horizontal do trecho suspenso do *riser*, e o objetivo adicional é minimizar o número de flutuadores. Os flutuadores são estruturas caras e que exigem um cuidado adicional na sua instalação e, dessa maneira, é importante que sejam minimizados quando se deseja reduzir custos.

### 6.2.3

#### Restrições

Um projeto de *riser*, para que seja viável, precisa respeitar restrições de geometria e restrições estruturais, em geral ligadas a esforços sobre a estrutura. Dessa maneira, 10 restrições foram impostas ao problema, sendo quatro ligadas a geometria e seis a capacidade estrutural, e estão resumidas na Tabela 6.14.

Geometria possível?	Sim ou não
Hsag mínimo	20 m
Hhog mínimo	200 m
Hhog máximo	500 m
UF(médio) máx	1
FX topo (médio) máx	0
UF(água) máx	1
FX topo (água) máx	0
UF(vazio) máx	1
FX topo (vazio) máx	0

Tabela 6.14: Restrições do problema

A restrição UF é o fator de utilização descrito na norma DNV-OF-F201



e está associada à capacidade estrutural, em que diz que as tensões atuantes devem ser menores que as tensões admissíveis do material. A outra restrição estrutural FX do topo é a tensão axial do elemento na região de topo e a restrição imposta impede que a tensão axial seja negativa, ou seja, que haja apenas tração.

As restrições de altura estão relacionadas ao posicionamento dos flutuadores, a SAG é a altura do ponto de inflexão mínimo antes dos flutuadores e HOG o ponto máximo de inflexão com os flutuadores, como mostra a Figura 6.9. A restrição de altura mínima do SAG é uma proteção para que não haja contato com o fundo do mar, enquanto a restrição de HOG é uma proteção para que diferentes *risers* possam estar na mesma região e seguramente não haja contato entre eles.

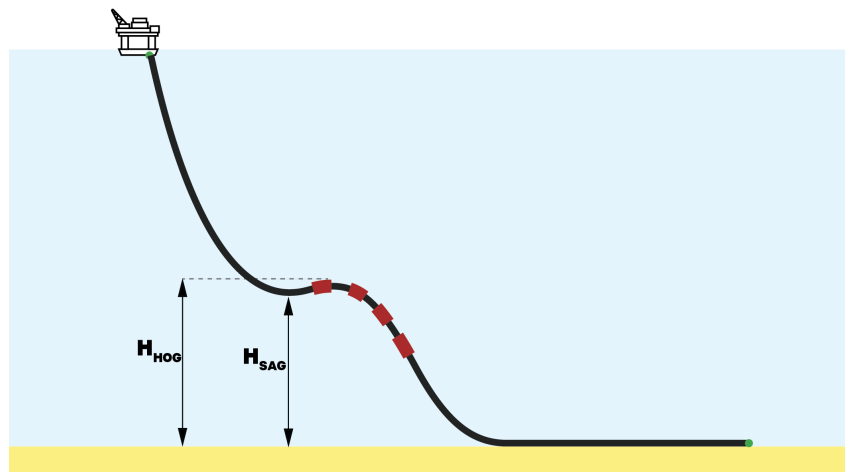


Figura 6.9: Alturas Hog e Sag de uma configuração Lazy Wave

#### 6.2.4 Resultados

##### Caso Mono-objetivo

Em se tratando de um caso mais complexo, envolvendo três casos de carregamentos e três tipos distintos de fluido interno, são necessárias mais iterações para que o método encontre soluções viáveis. Nesse caso mono-objetivo com variáveis contínuas foram rodadas 400 avaliações da função objetivo no total, sendo 200 delas de inicialização e mais 200 no loop principal da otimização. O resumo das variáveis das três melhores configurações obtidas durante um processo de otimização em termos da função objetivo, que neste

caso é a Projeção Horizontal do trecho suspenso, está na Tabela 6.15 e as suas respectivas restrições na Tabela 6.16.

$\alpha(^{\circ})$	L1 (m)	L2 (m)	L3 (m)	Df (m)	Lf (m)	SP (m)	Proj. H (m)
7,92	2239,00	463,36	1019,07	2,80	2,58	12,90	1535,05
7,92	2265,20	502,76	973,24	2,81	2,56	12,92	1603,12
7,95	2306,78	481,56	934,34	2,83	2,55	12,71	1608,31

Tabela 6.15: Variáveis contínuas e função objetivo para as três melhores configurações

Viável	Hsag min (m)	Hhog min (m)	Hhog max (m)
Sim	229,52	229,52	392,57
Sim	225,78	227,90	421,49
Sim	201,61	206,61	398,60

UF médio	FX topo médio (kN)	UF água	FX topo água (kN)	UF vazio	FX topo vazio (kN)
0,83	-4881,70	0,91	-3894,18	0,90	-3544,78
0,81	-4906,01	0,89	-3946,43	0,86	-3546,83
0,79	-4978,121	0,87	-4045,59	0,90	-3601,33

Tabela 6.16: Restrições para as três melhores configurações com variáveis contínuas

Ao todo, para o caso contínuo, das 400 configurações analisadas 22 satisfizeram todas as restrições impostas. É importante que mesmo em um sistema mais complexo o método seja capaz de satisfazer as restrições geométricas e estruturais, como é possível observar.

No caso discreto foram 400 avaliações da função objetivo, sendo 150 de inicialização e 250 de otimização do loop principal. O resumo das três melhores configurações com variáveis discretas está na Tabela 6.17 e as respectivas restrições na Tabela 6.18.

$\alpha(^{\circ})$	L1 (m)	L2 (m)	L3 (m)	Df (m)	Lf (m)	SP (m)	Proj. H (m)
6,60	2160	660	576	2,95	2,6	36	1521,94
6,75	2520	384	1776	2,15	2,8	24	1582,93
6,5	2688	780	912	2,65	2,3	24	1597,10

Tabela 6.17: Variáveis discretas e função objetivo para as três melhores configurações

Viável	Hsag min (m)	Hhog min (m)	Hhog max (m)
Sim	261,45	261,89	452,74
Sim	235,80	236,80	412,72
Sim	217,29	239,17	472,62

UF médio	FX topo médio (kN)	UF água	FX topo água (kN)	UF vazio	FX topo vazio (kN)
0,87	-4763,94	0,96	-3863,38	0,90	-3450,30
0,81	-4899,33	0,87	-4032,06	0,95	-3545,49
0,82	-4865,43	0,92	-4149,08	0,93	-3521,67

Tabela 6.18: Restrições para as três melhores configurações com variáveis discretas

Devido a limitações de opções, que dificulta a exploração do método pelo domínio, das 400 configurações estudadas, apenas seis configurações viáveis foram encontradas. Entretanto, o menor número de configurações viáveis não implica em uma solução pior. Quando comparado ao caso discreto as soluções são compatíveis, sendo até sua solução ótima ligeiramente melhor.

### Caso Multi-objetivo

Para esse estudo apenas variáveis contínuas foram consideradas e para observar a consistência do método, foram comparadas nove rodadas de otimização. Em todas elas foram realizadas 350 iterações no total, sendo na primeira rodada 100 iterações de inicialização e 250 na busca pelo mínimo enquanto para os demais casos foram 150 de inicialização e 200 na busca pelo mínimo. Os resultados de ambos os objetivos são encontrados na Tabela 6.19 e suas respectivas variáveis na Tabela 6.20.

Rodada	Proj H (m)	N Flut
1	1515,09	34
2	1563,88	33
3	1586,91	33
4	1518,58	34
5	1572,36	44
6	1574,18	44
	1576,40	43
	1691,53	40
	1698,39	39
7	1553,66	43
8	1628,80	29
9	1561,33	30

Tabela 6.19: Fronteira de Pareto para as nove rodadas realizadas

Rodada	$\alpha(^{\circ})$	$L_1$ (m)	$L_2$ (m)	$L_3$ (m)	$D_f$ (m)	$L_f$ (m)	SP (m)
1	7,96	2257,00	420,27	953,76	2,95	2,28	12,3
2	7,29	2271,96	484,16	1505,54	2,90	2,85	14,43
3	6,44	2369,18	534,00	2000,00	3,00	3,00	16,47
4	6,69	2298,57	494,49	910,49	2,95	2,61	14,42
5	6,00	1957,56	255,66	1252,12	2,85	3,00	12,00
6	7,26	2252,12	534,85	2000,00	3,00	2,00	12,00
	7,24	2222,18	541,38	1464,24	2,89	3,00	12,49
	8,00	2375,63	487,87	2000,00	2,64	3,00	12,00
	7,90	2391,50	472,62	1862,43	2,72	2,98	12,00
7	6,35	2343,87	516,90	1027,08	3,00	2,98	12,00
8	7,87	2031,67	866,21	1116,28	2,38	2,73	12
9	8,00	2287,04	453,88	2000,00	2,79	3,00	15,10

Tabela 6.20: Variáveis correspondentes às soluções das fronteiras de paretos

Quando se trata de otimização multi-objetivo, não é possível mais obter uma única solução que seja melhor que as demais, por muitas vezes os objetivos serem conflitantes. Nesse caso, se o projetista priorizar o número de flutuadores, mas ainda buscando uma solução satisfatória de projeção horizontal, provavelmente a configuração escolhida seria a da rodada 8 com 29 flutuadores. Caso o projetista esteja priorizando a projeção horizontal, seria

escolhida a rodada 1 e ainda, se for um equilíbrio entre as duas poderia ser a rodada 9.

Observando apenas a projeção horizontal vemos que as soluções foram próximas àquelas encontradas no teste mono-objetivo. Em algumas rodadas o método, buscando um equilíbrio das duas variáveis obteve uma projeção horizontal um pouco superior a encontrada com 1 objetivo porém com um menor número de objetivos, como é o caso da rodada 9.

Com a otimização desses dois objetivos, a fronteira de pareto em muitos casos se resumiu a apenas um ponto, como pode ser observado na Figura 6.10. Esta figura ilustra todas as soluções avaliadas do sistema. Em cada um dos eixos está um objetivo e os pontos em vermelho representam os pontos que violaram alguma restrição, os em amarelo as soluções que não violaram restrições mas não fazem parte da Fronteira de Pareto e em azul as soluções que fazem parte da Fronteira de Pareto. Em boa parte dos casos válidos, quando se diminui um dos objetivos o mesmo acontece com o outro. Se fossem objetivos mais conflitantes as soluções estariam mais espalhadas pelo domínio.

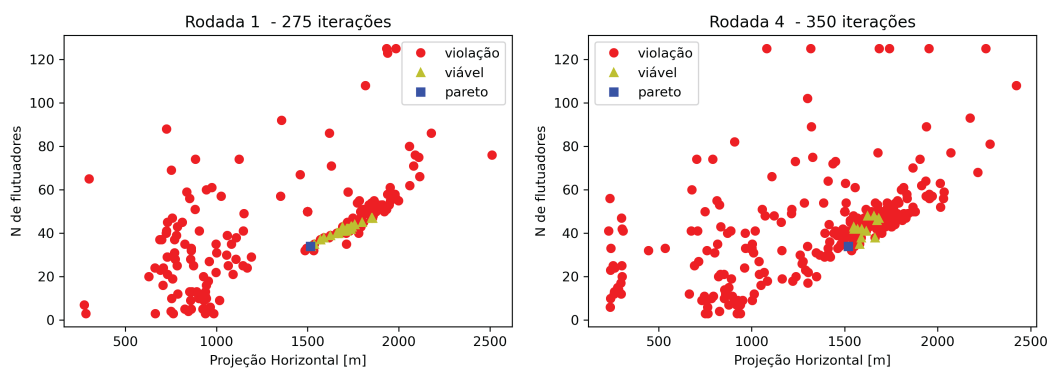


Figura 6.10: Exploração e soluções encontradas pelo método

O estudo adicionalmente teve como objetivo avaliar o desempenho do método aplicado de maneira serial e paralela. Para tal foi rodado o mesmo caso com o mesmo número de iterações, sendo na paralelização utilizados 10 processos paralelos. Como esperado, a qualidade dos resultados foram equivalentes e portanto a discussão focará nos tempos gastos por cada um. Foram comparados então os tempos totais e também o tempo gasto para avaliar a função objetivo, denominado de tempo de execução (tempo gasto pelos "consumidores") e o tempo gasto para avaliar o ponto da iteração seguinte, denominado de tempo *next loc* (tempo gasto pelo produtor). A Tabela 6.21 apresenta um resumo dos resultados obtidos.

	Serial	Paralelo
Tempo Total	29 h e 45 min	9 h e 10 min
Total next loc	1 h e 42 min	8 h e 27 min
Execução/core	28 h e 3 min	6 h e 29 min
Média next loc	24,44 s	121,64 s
Média execução	6,7 min	15,56 min

Tabela 6.21: Resumo dos tempos do processo de otimização multi-objetivo

As Figuras 6.11 e 6.12 mostram os gráficos de tempos gastos com *next loc* e execução para cada iteração e suas médias para o caso serial e as Figuras 6.13 e 6.14 para o caso paralelo.

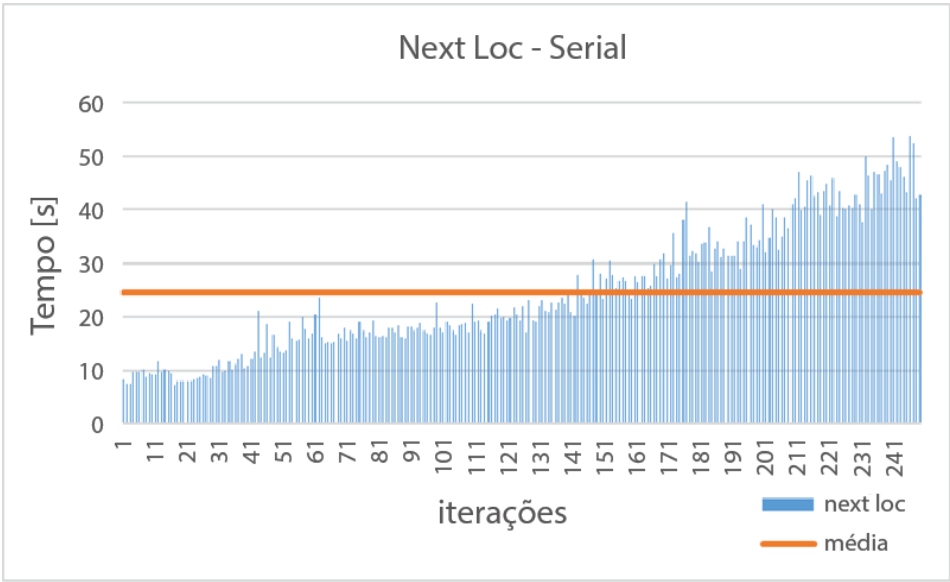


Figura 6.11: Tempos de next loc e média para caso serial

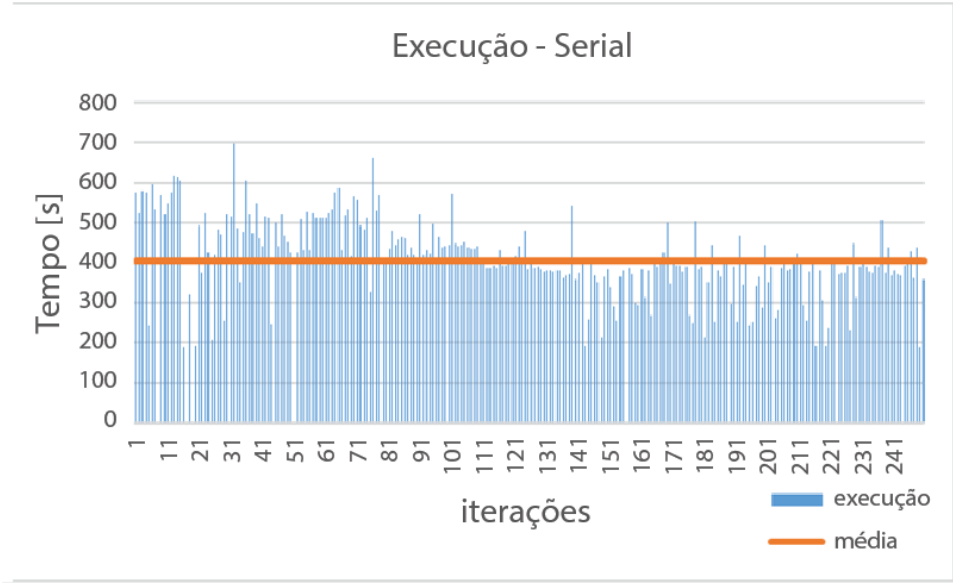


Figura 6.12: Tempos de execução e média para caso serial

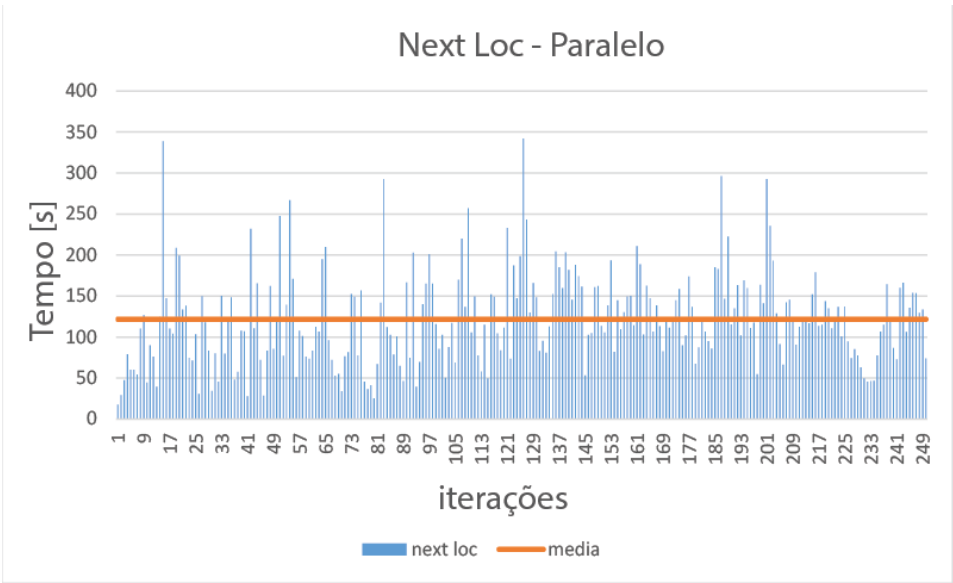


Figura 6.13: Tempos de next loc e média para caso paralelo

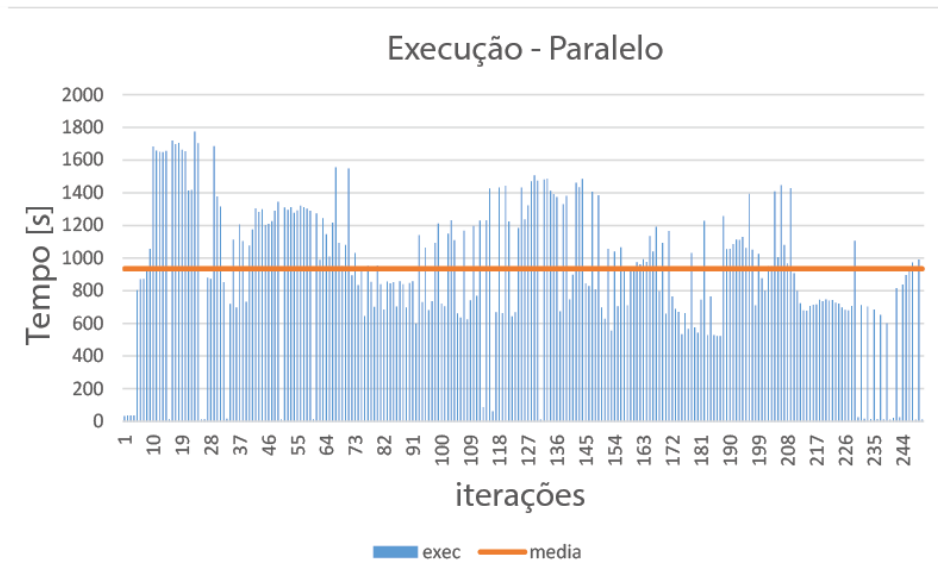


Figura 6.14: Tempos de execução e média para caso paralelo

O tempo total gasto no processo de otimização é significativamente reduzido quando se utilizam as técnicas de paralelismo. Na versão paralela observa-se um código três vezes mais rápido quando comparada com a versão serial. Se forem comparadas a média de cada atividade isoladamente, observa-se que a média é maior quando rodam-se processos paralelos, em que exigem mais da máquina e retardam os cálculos. Esse acréscimo, no entanto, é compensado pela existência de múltiplos cálculos simultâneos, sendo extremamente vantajosa a paralelização.

Na versão serial, como ocorre sequencialmente, observa-se que o tempo total de execução equivale a soma dos tempos de cada etapa, *next loc* e execução. No entanto, no processo paralelo esse cálculo não pode ser realizado em função da existência de múltiplas tarefas simultâneas. Uma conclusão a ser observada é que boa parte do tempo é ditada pelo produtor, em média ele ficou ocioso apenas 43 minutos do tempo total, enquanto os consumidores ficaram mais tempo ociosos, em média 2 horas e 41 minutos. Esse resultado era esperado pois é o produtor o responsável por ditar o caminho a ser seguido e é calculado um por vez.



Este trabalho apresentou a ferramenta de Otimização Bayesiana aplicada ao problema de otimização de configuração de *risers*. Os principais objetivos do trabalho eram implementar e validar o método como uma alternativa viável e eficaz para a resolução do problema. Para tal, a ferramenta foi aplicada a uma gama de problemas com características diferentes. O primeiro estudo consistiu em aplicar o método para o problema de um único objetivo: minimizar o comprimento total do *riser* utilizando o software de elementos finitos Orcaflex em 3 cenários diferentes, SLWR com lâminas d'água de 1200 m e 2150 m e RCHA com lâmina d'água de 2150 m. A fim de saber se o seu desempenho era competitivo, os mesmos cenários foram rodados para diferentes métodos encontrados na literatura, como MIDACO, Algoritmo Genético e GBNM.

O segundo caso aplicado tinha como objetivos principais aplicar o método para um problema de configuração mais robusto com mais casos de carregamento, validar a implementação multi-objetivo e validar o paralelismo do método. Para isso o método foi aplicado a uma configuração SLWR com três casos de carregamento e três preenchimentos utilizando o *software* de elementos finitos ANFLEX. Inicialmente o problema foi resolvido com a única função objetivo de minimizar a projeção horizontal do trecho suspenso utilizando variáveis contínuas e discretas. O segundo teste com o mesmo caso consistiu na otimização com dois objetivos. Adicionalmente à minimização da projeção do trecho suspenso do estudo anterior foi minimizado o número de flutuadores. A versão multi-objetivo foi rodada de maneira serial e maneira paralela. Essa diferenciação foi feita para que na última investigação fosse observada a eficiência da paralelização para tornar o processo ainda mais viável em termos de tempo de execução para um projetista.

Em ambos os casos estudados o método demonstrou ser eficaz e versátil. No primeiro caso, quando comparado a outros métodos encontrados na literatura a otimização Bayesiana se mostrou bastante competitiva. Houve um desempenho significativamente superior quanto ao número de avaliações da função objetivo em relação aos métodos GBNM e Algoritmo Genético e próximo (em alguns casos superior) em termos do valor da função objetivo e número de avaliações em relação ao código MIDACO.

No segundo caso o método se mostrou eficaz para encontrar múltiplas soluções viáveis e com bons resultados de maneira consistente em ambos os testes com um objetivo e multi-objetivo. A paralelização do método multi-

objetivo se mostrou eficaz, reduzindo o tempo total em 70% com 10 processos simultâneos.

As configurações resultantes, no entanto, ainda não representam o dimensionamento completo do *riser*. Para que o projeto esteja completo é necessário ainda uma análise detalhada de fadiga, de VIV e ainda outras questões relacionadas a outras áreas que envolvem o projeto. A ferramenta desenvolvida neste trabalho ajuda o projetista a pular algumas etapas iniciais que anteriormente eram desgastantes e demoradas.

Como sugestão para trabalhos futuros é interessante que sejam introduzidas, além da análise de carga extrema, uma análise de fadiga e/ou VIV. Essas análises são consideravelmente mais complexas e mais custosas computacionalmente. Para lidar com esse problema há algumas técnicas que poderiam agilizar o processo de otimização. Por exemplo, a utilização de matrizes esparsas pode agilizar a escolha do ponto da iteração, a utilização de redes neurais para agilizar o processo Gaussiano e até implementações de processos Gaussianos utilizando placa de vídeo. Do ponto de vista das simulações de elementos finitos, as análises podem ser paralelizadas em ambientes de *clusters* ou até *clouds* comerciais para que seja possível a utilização de mais processos em paralelo e que os cálculos sejam ainda mais rápidos.

## Referências bibliográficas

- 1 PAINÉIS Dinâmicos sobre Exploração e Produção de Petróleo e Gás. <<https://www.gov.br/anp/pt-br/centrais-de-conteudo/paineis-dinamicos-da-anp/paineis-dinamicos-sobre-exploracao-e-producao-de-petroleo-e-gas>>. Acessado em 03/06/2021.
- 2 CARDOSO, P.; CASAPRIMA, N.; SENHORA, F.; MENEZES, I. Optimization of catenary risers with hydrodynamic dampers. **Ocean Engineering**, Elsevier, v. 184, p. 134–142, 2019.
- 3 CLAUSEN, T.; D'SOUZA, R. Dynamic risers key component for deepwater drilling, floating production. **Offshore (Conroe, Tex.)**, v. 61, n. 5, p. 89–93, 2001.
- 4 BROCHU, E.; CORA, V. M.; FREITAS, N. D. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. **arXiv preprint arXiv:1012.2599**, 2010.
- 5 PRODUÇÃO de petróleo e gás teve recorde em 2020 e aumentou 52,71% em relação a 2010. <[www.gov.br/anp/pt-br/canais\\_atendimento/imprensa/noticias-comunicados/producao-de-petroleo-e-gas-teve-recorde-em-2020-e-aumentou-52-71-em-relacao-a-2010](http://www.gov.br/anp/pt-br/canais_atendimento/imprensa/noticias-comunicados/producao-de-petroleo-e-gas-teve-recorde-em-2020-e-aumentou-52-71-em-relacao-a-2010)>. Acessado em 03/06/2021.
- 6 JACOB, B. P.; REYES, M. C.; LIMA, B. S. de; TORRES, A. L.; MOURELLE, M. M.; SILVA, R. et al. Alternative configurations for steel catenary risers for turret-moored fpsos. In: INTERNATIONAL SOCIETY OF OFFSHORE AND POLAR ENGINEERS. **The Ninth International Offshore and Polar Engineering Conference**. [S.l.], 1999.
- 7 AGUIAR, L. L.; MENEZES, I. F. M.; MOURELLE, M. M. **Patent application: "Hydrodynamic Damper for Catenary Risers"**. 2010.
- 8 BRITO, R.; AGUIAR, L.; PRADO, V. et al. Technical feasibility study of steel catenary risers for pre-salt field developments. In: OFFSHORE TECHNOLOGY CONFERENCE. **OTC Brasil**. [S.l.], 2017.
- 9 CALANDRA, R.; SEYFARTH, A.; PETERS, J.; DEISENROTH, M. P. An experimental comparison of bayesian optimization for bipedal locomotion. In: IEEE. **2014 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2014. p. 1951–1958.
- 10 MARTINEZ-CANTIN, R. Funneled bayesian optimization for design, tuning and control of autonomous systems. **IEEE transactions on cybernetics**, IEEE, v. 49, n. 4, p. 1489–1500, 2018.
- 11 LAM, R.; POLOCZEK, M.; FRAZIER, P.; WILLCOX, K. E. Advances in bayesian optimization with applications in aerospace engineering. In: **2018 AIAA Non-Deterministic Approaches Conference**. [S.l.: s.n.], 2018. p. 1656.

- 12 BARTOLI, N.; LEFEBVRE, T.; DUBREUIL, S.; OLIVANTI, R.; PRIEM, R.; BONS, N.; MARTINS, J. R.; MORLIER, J. Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. **Aerospace Science and technology**, Elsevier, v. 90, p. 85–102, 2019.
- 13 RIGGS, H. Automated preliminary design of flexible riser configurations. In: **International Conference on Offshore Mechanics and Arctic Engineering**. [S.l.: s.n.], 1989.
- 14 RIGGS, H.; LERAAND, T. Efficient static analysis and design of flexible risers. **The Journal of Offshore Mechanics and Arctic Engineering**, v. 61, n. 3, p. 235–240, 1991.
- 15 LARSEN, C. M.; HANSON, T. Optimization of catenary risers. **The Journal of Offshore Mechanics and Arctic Engineering**, v. 121, n. 2, p. 90–94, 1999.
- 16 RODRIGUES, G. J. de O. **FERRAMENTAS COMPUTACIONAIS PARA OTIMIZAÇÃO E SÍNTESE DE SISTEMAS HÍBRIDOS DE RISERS BASEADOS NO CONCEITO DE BÓIA DE SUBSUPERFÍCIE**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2004.
- 17 LIMA, B. d. S. Leite Pires de; JACOB, B. P.; EBECKEN, N. F. F. A hybrid fuzzy/genetic algorithm for the design of offshore oil production risers. **International Journal for numerical methods in engineering**, Wiley Online Library, v. 64, n. 11, p. 1459–1482, 2005.
- 18 PINA, A. A. de; ALBRECHT, C. H.; LIMA, B. S. L. P. de; JACOB, B. P. Tailoring the particle swarm optimization algorithm for the design of offshore oil production risers. **Optimization and engineering**, Springer, v. 12, n. 1, p. 215–235, 2011.
- 19 ANDRADE, E. Q. D.; AGUIAR, L. L. de; SENRA, S. F.; SIQUEIRA, E. F. N.; TORRES, A. L. F. L.; MOURELLE, M. M. et al. Optimization procedure of steel lazy wave riser configuration for spread moored fpsos in deepwater offshore brazil. In: OFFSHORE TECHNOLOGY CONFERENCE. **Offshore Technology Conference**. [S.l.], 2010.
- 20 VIEIRA, I. N.; SILVA, A. J. M.; LIMA, B. de; JACOB, B.; ALBRECHT, C. H. A comparative study applied to risers optimization using bio-inspired algorithms. **International Journal of Modeling and Simulation for the Petroleum Industry**, v. 3, n. 1, 2009.
- 21 VIEIRA, I. N.; LIMA, B. S. L. P. de; JACOB, B. P. Bio-inspired algorithms for the optimization of offshore oil production systems. **International journal for numerical methods in engineering**, Wiley Online Library, v. 91, n. 10, p. 1023–1044, 2012.
- 22 CARDOSO, P. H. d. S. **Otimização da configuração geométrica de risers rígidos**. Dissertação (Mestrado) — PUC-Rio, 2019.
- 23 NARIÑO, G. A. R. **Otimização de Risers em Catenária com Amortecedores Hidrodinâmicos**. Dissertação (Mestrado) — PUC-Rio, 2014.

- 24 KUSHNER, H. J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. **Journal of Basic Engineering**, v. 86, n. 1, p. 97–106, 1964.
- 25 ZHILINSKAS, A. Single-step bayesian search method for an extremum of functions of a single variable. **Cybernetics**, Springer, v. 11, n. 1, p. 160–166, 1975.
- 26 MOCKUS, J.; TIESIS, V.; ZILINSKAS, A. The application of bayesian methods for seeking the extremum. **Towards global optimization**, v. 2, n. 117–129, p. 2, 1978.
- 27 MOČKUS, J. On bayesian methods for seeking the extremum. In: SPRINGER. **Optimization techniques IFIP technical conference**. [S.l.], 1975. p. 400–404.
- 28 MOCKUS, J. **Bayesian approach to global optimization: theory and applications**. [S.l.]: Springer Science & Business Media, 2012. v. 37.
- 29 JONES, D. R.; SCHONLAU, M.; WELCH, W. J. Efficient global optimization of expensive black-box functions. **Journal of Global optimization**, Springer, v. 13, n. 4, p. 455–492, 1998.
- 30 SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. **arXiv preprint arXiv:1206.2944**, 2012.
- 31 MENDOZA, H.; KLEIN, A.; FEURER, M.; SPRINGENBERG, J. T.; HUTTER, F. Towards automatically-tuned neural networks. In: PMLR. **Workshop on Automatic Machine Learning**. [S.l.], 2016. p. 58–65.
- 32 SHAHRIARI, B. **Practical Bayesian optimization with application to tuning machine learning algorithms**. Tese (Doutorado) — University of British Columbia, 2016.
- 33 INMAN, M. J.; EARWOOD, J. M.; ELSHERBENI, A. Z.; SMITH, C. E. Bayesian optimization techniques for antenna design. **Progress In Electromagnetics Research**, EMW Publishing, v. 49, p. 71–86, 2004.
- 34 FRAZIER, P. I.; WANG, J. Bayesian optimization for materials design. In: **Information Science for Materials Discovery and Design**. [S.l.]: Springer, 2016. p. 45–75.
- 35 KOPSIAFTIS, G.; PROTOPAPADAKIS, E.; VOULODIMOS, A.; DOULAMIS, N.; MANTOGLOU, A. Gaussian process regression tuned by bayesian optimization for seawater intrusion prediction. **Computational intelligence and neuroscience**, Hindawi, v. 2019, 2019.
- 36 PIROT, G.; KRITYAKIERNE, T.; GINSBOURGER, D.; RENARD, P. Contaminant source localization via bayesian global optimization. **Hydrology and earth system sciences**, Copernicus GmbH, v. 23, n. 1, p. 351–369, 2019.
- 37 WANG, J. **Bayesian optimization with parallel function evaluations and multiple information sources: methodology with applications in biochemistry, aerospace engineering, and machine learning**. Tese (Doutorado) — Cornell University, 2017.

- 38 WANG, S.; CHEN, S. A novel bayesian optimization framework for computationally expensive optimization problem in tight oil reservoirs. In: ONEPETRO. **SPE Annual Technical Conference and Exhibition**. [S.l.], 2017.
- 39 QUEIPO, N. V.; HAFTKA, R. T.; SHYY, W.; GOEL, T.; VAIDYANATHAN, R.; TUCKER, P. K. Surrogate-based analysis and optimization. **Progress in aerospace sciences**, Elsevier, v. 41, n. 1, p. 1–28, 2005.
- 40 SIMPSON, T.; TOROPOV, V.; BALABANOV, V.; VIANA, F. Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come-or not. In: **12th AIAA/ISSMO multidisciplinary analysis and optimization conference**. [S.l.: s.n.], 2008. p. 5802.
- 41 WANG, G. G.; SHAN, S. Review of metamodeling techniques in support of engineering design optimization. **Journal of Mechanical Design**, v. 129, n. 4, p. 370–380, 2007.
- 42 BHOSEKAR, A.; IERAPETRITOU, M. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. **Computers & Chemical Engineering**, Elsevier, v. 108, p. 250–267, 2018.
- 43 ZITZLER, E. **Evolutionary algorithms for multiobjective optimization: Methods and applications**. [S.l.]: Citeseer, 1999. v. 63.
- 44 EMMERICH, M. T.; GIANNAKOGLU, K. C.; NAUJOKS, B. Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 10, n. 4, p. 421–439, 2006.
- 45 EMMERICH, M.; NAUJOKS, B. Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. In: **Adaptive computing in design and manufacture VI**. [S.l.]: Springer, 2004. p. 249–260.
- 46 PONWEISER, W.; WAGNER, T.; VINCZE, M. Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In: IEEE. **2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)**. [S.l.], 2008. p. 3515–3522.
- 47 PONWEISER, W.; WAGNER, T.; BIERMANN, D.; VINCZE, M. Multiobjective optimization on a limited budget of evaluations using model-assisted  $\mathcal{S}$ -metric selection. In: SPRINGER. **International Conference on Parallel Problem Solving from Nature**. [S.l.], 2008. p. 784–794.
- 48 EMMERICH, M. T.; DEUTZ, A. H.; KLINKENBERG, J. W. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In: IEEE. **2011 IEEE Congress of Evolutionary Computation (CEC)**. [S.l.], 2011. p. 2147–2154.
- 49 LTD., O. **Orcaflex**. [S.l.]: <<https://www.orcina.com/orcaflex/>>.
- 50 MOURELLE, M.; GONZALEZ, E.; JACOB, B. Anflex-computational system for flexible and rigid riser analysis. In: **Proceedings of the 9th International Symposium on Offshore Engineering, Brazil**. [S.l.: s.n.], 1995.

- 51 NOCEDAL, J.; WRIGHT, S. **Numerical optimization**. [S.l.]: Springer Science & Business Media, 2006.
- 52 RASMUSSEN, C. E. Gaussian processes in machine learning. In: SPRINGER. **Summer school on machine learning**. [S.l.], 2003. p. 63–71.
- 53 SASENA, M. J. **Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations**. Tese (Doutorado) — University of Michigan, 2002.
- 54 SNOEK, J.; RIPPEL, O.; SWERSKY, K.; KIROS, R.; SATISH, N.; SUNDARAM, N.; PATWARY, M.; PRABHAT, M.; ADAMS, R. Scalable bayesian optimization using deep neural networks. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 2171–2180.
- 55 SOBOLOV, I. M. On the distribution of points in a cube and the approximate evaluation of integrals. **Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki**, Russian Academy of Sciences, Branch of Mathematical Sciences, v. 7, n. 4, p. 784–802, 1967.
- 56 MATÉRN, B. **Spatial variation**. [S.l.]: Springer Science & Business Media, 2013. v. 36.
- 57 FLETCHER, R. **Practical Methods of Optimization; (2nd Ed.)**. USA: Wiley-Interscience, 1987. ISBN 0471915475.
- 58 ZHU, C.; BYRD, R. H.; LU, P.; NOCEDAL, J. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. **ACM Transactions on Mathematical Software (TOMS)**, ACM New York, NY, USA, v. 23, n. 4, p. 550–560, 1997.
- 59 SCHONLAU, M. **Computer experiments and global optimization**. Tese (Doutorado) — University of Waterloo, 1997.
- 60 GARDNER, J. R.; KUSNER, M. J.; XU, Z. E.; WEINBERGER, K. Q.; CUNNINGHAM, J. P. Bayesian optimization with inequality constraints. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2014. v. 2014, p. 937–945.
- 61 GELBART, M. A.; SNOEK, J.; ADAMS, R. P. Bayesian optimization with unknown constraints. **arXiv preprint arXiv:1403.5607**, 2014.
- 62 ROSSUM, G. V.; JR, F. L. D. **Python tutorial**. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995. v. 620.
- 63 IERUSALIMSKY, R.; FIGUEIREDO, L. H. D.; FILHO, W. C. Lua—an extensible extension language. **Software: Practice and Experience**, Wiley Online Library, v. 26, n. 6, p. 635–652, 1996.
- 64 AUTHORS, T. G. **GPyOpt: A Bayesian Optimization framework in Python**. 2016. <<http://github.com/SheffieldML/GPyOpt>>.

- 65 KLUYVER, T.; RAGAN-KELLEY, B.; PÉREZ, F.; GRANGER, B.; BUSSON-NIER, M.; FREDERIC, J.; KELLEY, K.; HAMRICK, J.; GROUT, J.; CORLAY, S.; IVANOV, P.; AVILA, D.; ABDALLA, S.; WILLING, C.; TEAM, J. development. Jupyter notebooks - a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. Netherlands: IOS Press, 2016. p. 87–90. Disponível em: <<https://eprints.soton.ac.uk/403913/>>.
- 66 LUERSEN, M. A.; RICHE, R. L. Globalized nelder–mead method for engineering optimization. **Computers & structures**, Elsevier, v. 82, n. 23-26, p. 2251–2260, 2004.
- 67 SCHLUETER, M.; ERB, S.; GERDTS, M.; KEMBLE, S.; RUCKMANN, J. Midaco on minlp space applications. **Advances in Space Research**, Elsevier, v. 51, n. 7, p. 1116–1131, 2013.
- 68 FORTIN, F.-A.; De Rainville, F.-M.; GARDNER, M.-A.; PARIZEAU, M.; GAGNÉ, C. DEAP: Evolutionary algorithms made easy. **Journal of Machine Learning Research**, v. 13, p. 2171–2175, jul 2012.
- 69 NELDER, J. A.; MEAD, R. A simplex method for function minimization. **The computer journal**, Oxford University Press, v. 7, n. 4, p. 308–313, 1965.
- 70 SCHLUETER, M. **Nonlinear mixed integer based optimization technique for space applications**. Tese (Doutorado) — University of Birmingham, 2012.
- 71 SCHLUETER, M.; ERB, S. O.; GERDTS, M.; KEMBLE, S.; RÜCKMANN, J.-J. Midaco on minlp space applications. **Advances in Space Research**, Elsevier, v. 51, n. 7, p. 1116–1131, 2013.
- 72 SCHLUETER, M.; EGEA, J.; BANGA, J. Extended ant colony optimization for non-convex mixed integer nonlinear programming. **Computers and Operations Research**, Elsevier, v. 36, n. 7, p. 2217–2229, 2009.
- 73 SCHLUETER, M.; GERDTS, M. The oracle penalty method. **Journal of Global Optimization**, Springer, v. 47, n. 2, p. 293–325, 2010.
- 74 SCHLUETER, M.; MUNETOMO, M. Midaco solver-user manual. **IIC, Hokkaido University, Japan**, 2016.
- 75 DIJKSTRA, E. W. Cooperating sequential processes. In: **The origin of concurrent programming**. [S.l.]: Springer, 1968. p. 65–138.
- 76 DNV, G. Dnvg-l-st-f201: Dynamic risers. **Høvik, Norway**, 2018.