

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

Metamodelagem Adaptativa para Otimização via Simulação a Eventos
Discretos

João Victor Soares do Amaral

Itajubá, Abril de 2025

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

João Victor Soares do Amaral

Metamodelagem Adaptativa para Otimização via Simulação a Eventos
Discretos

Tese submetida ao programa de Pós-Graduação em
Engenharia de Produção como parte dos requisitos
para obtenção do Título de Doutor em Ciências em
Engenharia de Produção.

Área: Engenharia de Produção

Orientador: Prof. Dr. José Arnaldo Barra Montevechi

Coorientador: Prof. Dr. Rafael de Carvalho Miranda

Abril de 2025

Itajubá

DEDICATÓRIA

A Deus, que nos dá o dom da vida, aos meus pais, Maria das Mercês e Juscelino, por todo apoio e ensinamentos, à minha noiva, Rafaela, e em especial ao meu querido avô, João Ribeiro, que sempre esteve ao meu lado.

AGRADECIMENTOS

A Deus, nunca terei como agradecer por todas as oportunidades que Ele sempre colocou no meu caminho. O Senhor é o meu pastor e nada me faltará.

Aos meus pais, Maria das Mercês Mota Soares e Juscelino da Silva Amaral, por todo apoio, amor e ensinamentos durante minha vida.

A minha noiva, Rafaela, por todo amor e compreensão nos momentos de ausência e por sempre me apoiar e incentivar a ir mais longe. Ao lado de um grande homem sempre há uma grande mulher.

A minha querida avó Efigênia, por todo carinho e amor durante toda minha vida, e ao meu avô Francisco, que aí do céu sempre olha por nós.

A minha avó Maria, por sempre cuidar, amar e orar por nós, mesmo hoje sendo uma estrelinha eu sei que sempre está ao meu lado.

Em especial, gostaria de agradecer ao meu avô João Ribeiro, ao qual convivi por poucos anos, que foram fundamentais para me tornar o homem que sou. Obrigado por sempre estar comigo, festejando nas vitórias e me carregando no seu colo nas dificuldades.

Ao meu orientador, José Arnaldo Barra Montevechi, pela orientação deste trabalho e me ensinar não apenas sobre engenharia, mas também que ser professor vai muito além de ter um título. É ter ética, ser justo e priorizar o conhecimento acima de qualquer coisa.

Ao meu co-orientador, Rafael de Carvalho Miranda, pelas contribuições valiosas a este trabalho e por sempre estar disposto a me ajudar e ensinar. Contigo aprendi que a engenharia não se resume apenas em otimizar processos, mas também serve como uma ferramenta valiosa para tentar melhorar o bem-estar das pessoas.

A todos professores e servidores da UNIFEI, em especial aos professores Renato Lima, Carlos Mello, José Henrique, Edson Pamplona, Pedro Paulo Balestrassi, Carlos Sanches, André Medeiros, Anderson Paulo, Fabiano Leal, José Antônio de Queiroz e todos outros que pacientemente transmitiram seus conhecimentos.

Aos professores Alexandre Pinho, Antonio F. Branco, Luiz Satoru, Marcone Souza e Cléber Rocco por aceitarem prontamente o convite para participarem de minha banca de doutorado, contribuindo para o desenvolvimento deste trabalho e de minha carreira como pesquisador.

Aos amigos da UFSJ, em especial aos professores Allexandre, Robson, Wilson, Kívia, Roberta, Flávio, Leandro, Guilherme, entre outros, por me guiarem e auxiliarem a desenvolver a base a qual estou construindo minha carreira.

Ao professor Anderson Rodrigo, o qual me recebeu com extrema atenção e apoio na NCSU, provendo uma experiência acadêmica e pessoal única em minha vida.

Aos amigos da NEAAD, em especial Carlos Henrique, Afonso, Alyson, Wesley, Guilherme, Serafim, André, Gustavo, Arthur e Milena.

Aos amigos da FlexSim Brasil, Michael, Flávio, Beatriz e Felipe, pelo apoio incondicional a esta pesquisa.

Aos amigos da Estratégia/RTC da Petrobras, em especial Giane, Victor, Danielle, Vanessa, Fernanda, Thiago, Marcia, Roberta, Patrícia, Murta, Fidellis e Laura.

A CAPES, FAPEMIG e CNPq pelo apoio e incentivo à pesquisa, em especial a essa.

A Universidade Federal de Itajubá, a quem sempre terei uma dívida eterna de gratidão.

A todos vocês, meu muito obrigado!

EPÍGRAFE

A verdadeira medida de um homem não é como ele se comporta em momentos de conforto e conveniência, mas como ele se mantém em tempos de controvérsia e desafio.

Martin Luther King Jr.

RESUMO

Pilar da Indústria 4.0, a Otimização via Simulação (OvS) é uma ferramenta poderosa utilizada em diversos campos, que permite a avaliação de um sistema sob diversas condições, facilita a sua análise de desempenho e torna a tomada de decisões mais eficiente. Por outro lado, a OvS pode ser demorada, especialmente quando se considera a otimização de modelos complexos. Nesse sentido, a metamodelagem emergiu como uma técnica promissora para OvS. A metamodelagem visa estabelecer e estimar uma relação entre as entradas e saídas de um modelo de simulação, criando um modelo simplificado utilizado para avaliar soluções potenciais durante o processo de otimização. Este estudo apresenta um novo *framework* de metamodelagem adaptativa, denominado de *Adaptive Metamodeling-based Simulation Optimization* (AMSO), para otimização de modelos de Simulação a Eventos Discretos complexos e computacionalmente intensivos. Esta abordagem combina aprendizado de máquina e técnicas meta-heurísticas para identificar as áreas mais promissoras do espaço de soluções que podem ser exploradas de forma mais eficiente para alcançar soluções de alta qualidade. O *framework* proposto é avaliado em três objetos de estudo reais da indústria: um problema de alocação de recursos em modelo de gêmeo digital de manufatura, um projeto de expansão de capacidades de uma usina de minério de ferro e um problema de alocação em um hospital temporário. Comparado ao método *Efficient Global Optimization*, o AMSO encontrou uma solução 8,1% melhor (em média) no primeiro objeto de estudo, 9,7% no segundo e 28,4% no terceiro, sem diferença significativa no tempo computacional gasto. Além disso, a AMSO encontrou soluções estatisticamente iguais ao método do Algoritmo Genético, mas que exigiram 83,6%, 90,6% e 65,2% menos tempo computacional no primeiro, segundo e terceiro caso, respectivamente. Os resultados encontrados mostram a aplicabilidade e robustez do *framework* proposto para otimização dos objetos de estudo analisados.

Palavras-chave: Otimização via simulação; Metamodelagem adaptativa; Simulação a Eventos Discretos; *Gradient-boosted Trees*; *Machine learning*.

ABSTRACT

A pillar of Industry 4.0, Simulation Optimization (SO) is a powerful tool used across several fields, which allows the evaluation of a system under different conditions, facilitates its performance analysis and makes decision-making more efficient. On the other hand, the SO might be time-consuming, particularly when considering complex model optimization. In this sense, metamodeling has emerged as a promising technique for simulation optimization. Metamodeling aims to establish and estimate a relationship between the inputs and outputs of a simulation model, creating a simplified model used to evaluate potential solutions during the optimization process. Metamodeling approaches can be classified as metamodeling with a fixed experimental design, in which a single experimental database is generated at the beginning of the project and the metamodel is trained exclusively on this basis, and adaptive metamodeling, which is based on the iterative construction of metamodels that are adjusted and refined throughout the optimization process. This paper presents a novel metamodeling framework, called Adaptive Metamodeling-based Simulation Optimization (AMSO), to optimize complex and expansive discrete event simulation models. The proposed approach combines machine learning and metaheuristic techniques to identify the most promising areas of the solution space that can be explored more efficiently to achieve high-quality solutions. The proposed framework is evaluated in three real-world industry case studies: a resource allocation problem in a manufacturing digital twin model, a capacity expansion project for an iron ore plant, and an allocation problem in a temporary hospital. Compared to the Efficient Global Optimization method, AMSO found a solution that was 8.1% better (on average) in the first case study, 9.7% better in the second, and 28.4% better in the third, with no significant difference in computational time spent. Additionally, AMSO found solutions statistically equivalent to the Genetic Algorithm method but required 83.6%, 90.6%, and 65.2% less computational time in the first, second, and third cases, respectively. The results found demonstrate the applicability and robustness of the proposed framework for optimizing the study objects analyzed.

Keywords: *Simulation optimization; Adaptive metamodeling; Discrete event simulation; Gradient-boosted Trees; Machine learning.*

LISTA DE FIGURAS

Figura 1: Processo de OvS baseada em modelos (a) e baseada em metamodelos (b).....	18
Figura 2: <i>Framework</i> geral para metamodelagem adaptativa	20
Figura 3: Etapas de um projeto de simulação a eventos discretos.....	28
Figura 4: Processo de OvS	29
Figura 5: Relação entre cenários do mundo real, modelos de simulação e metamodelos.....	31
Figura 6: Diagrama esquemático do <i>Bagging Regression</i>	35
Figura 7: Exemplos de arranjos LHD	42
Figura 8: Classificação da pesquisa.....	43
Figura 9: Sistema de resolução de problemas	44
Figura 10: Etapas da pesquisa	45
Figura 11: Procedimento de busca/triagem dos artigos.....	54
Figura 12: Dados coletados dos artigos.....	55
Figura 13: (a) Áreas de aplicação, (b) tipos de variáveis e (c) objetivos	60
Figura 14: Funções de Aquisição (a) e Método de otimização (b) mais usados.	68
Figura 15: Número de etapas utilizadas pelos trabalhos (a) e percentual de trabalhos que utilizaram cada etapa em seus <i>Frameworks</i> Utilizados/Propostos (b).	72
Figura 16: <i>Framework Adaptive Metamodeling-based Simulation Optimization</i>	81
Figura 17: Modelo computacional (OE1).....	88
Figura 18: Gráfico de convergência para EGO (a) e o AMSO (b) com 80 experimentos (OE1)	91
Figura 19: <i>Boxplot</i> para valor do lucro atingido após 80 experimentos (a) e tempo gasto na otimização (b) (OE1)	92
Figura 20: Gráfico de convergência para EGO (a) e o AMSO (b) com 160 experimentos (OE1)	93
Figura 21: <i>Boxplot</i> para valor do lucro atingido após 160 experimentos (a) e tempo gasto na otimização (b) (OE1)	94
Figura 22: Gráfico de convergência do AG para o OE1	95
Figura 23: Comparativo entre funções de aquisição para o OE1	96
Figura 24: Gráfico de convergência do OE1 para os três critérios de parada	97
Figura 25: Modelo computacional (OE2).....	99
Figura 26: Gráfico de convergência para EGO (a) e o AMSO (b) com 70 experimentos (OE2)	100

Figura 27: <i>Boxplot</i> para valor do lucro atingido após 70 experimentos (a) e tempo gasto na otimização (b) (OE2)	101
Figura 28: Gráfico de convergência do AG para o OE2	102
Figura 29: Comparativo entre funções de aquisição para o OE2	103
Figura 30: Modelo computacional (OE3).....	105
Figura 31: Gráfico de convergência para EGO (a) e o AMSO (b) com 100 experimentos (OE3)	106
Figura 32: <i>Boxplot</i> para valor de FO atingido após 100 experimentos (a) e tempo gasto na otimização (b) (OE3)	107
Figura 33: Gráfico de convergência do AG para o OE3	108
Figura 34: Comparativo entre funções de aquisição para o OE3	109

LISTA DE TABELAS

Tabela 1: Características dos problemas	56
Tabela 2: Métodos adotados	63
Tabela 3: Métricas e critérios de parada	74
Tabela 4: Variáveis de decisão e limites (OE1)	88
Tabela 5: Variáveis de decisão e limites (OE2)	98
Tabela 6: Variáveis de decisão e limites (OE3)	104

LISTA DE QUADROS

Quadro 1: Resumo das revisões sobre Metamodelagem Adaptativa	47
Quadro 2: Artigos por área versus tipo de problema.....	58
Quadro 3: Artigos por metamodelo adotado versus método de simulação	66

LISTA DE ABREVIÇÕES

AG	Algoritmo Genético
AMSO	<i>Adaptive Metamodeling-Based Simulation Optimization</i>
BEI	<i>Balanced Expected Improvement</i>
B-GBT	<i>Bagging-Based Gradient Boosted Trees</i>
DoE	<i>Design of Experiments</i>
EGO	<i>Efficient Global Optimization</i>
EI	<i>Expected Improvement</i>
FA	Função de Aquisição
FO	Função Objetivo
GBT	<i>Gradient Boosted Trees</i>
HP	Hiperparâmetros
LHD	<i>Latin Hypercube Design</i>
LOS	<i>Length of Stay</i>
MA	Metamodelagem Adaptativa
MAE	<i>Mean Absolute Error</i>
MAOvS	Metamodelagem Adaptativa para Otimização via Simulação
ML	Machine Learning
MSE	<i>Mean Square Error</i>
OHP	Otimização de Hiperparâmetros
OOB	<i>Out-of-Bag</i>
OvS	Otimização via Simulação
QP	Questões de Pesquisa
R	Coefficiente de Correlação
R^2	Coefficiente de Determinação
RBF	<i>Radial Basis Function</i>
RMSE	<i>Root Mean Square Error</i>
RNA	Redes Neurais Artificiais
RSL	Revisão Sistemática de Literatura
SED	Simulação a Eventos Discretos
SVM	<i>Support Vector Machine</i>

SUMÁRIO

1. INTRODUÇÃO	17
1.1. Contextualização.....	17
1.2. Objetivos.....	19
1.3. Motivação	20
1.4. Contribuições	22
1.5. Delimitação da Pesquisa	23
1.6. Estrutura do Trabalho	24
2. FUNDAMENTAÇÃO TEÓRICA.....	26
2.1. Simulação a Eventos Discretos.....	26
2.2. Otimização via Simulação (OvS).....	28
2.3. Definições Gerais de Metamodelagem	30
2.3.1. <i>Bagging</i>	35
2.3.2. <i>Gradient Boosted Trees</i>	37
2.4. Algoritmo Genético	39
2.5. <i>Latin Hypercube Design</i>	41
3. MÉTODO DE PESQUISA	43
3.1. Classificação da Pesquisa	43
3.2. Etapas da Pesquisa a Serem Desenvolvidas	45
4. REVISÃO SISTEMÁTICA DE LITERATURA.....	47
4.1. Trabalhos Relacionados	47
4.2. O Método de Revisão Sistemática de Literatura.....	50
4.3. Planejamento das Questões de Pesquisa.....	51
4.4. Busca e Triagem dos Artigos.....	52
4.5. Análise e Síntese	54
4.6. Resultados e Discussões da RSL	55
4.6.1. Características dos Problemas	55
4.6.2. Métodos Adotados.....	62
4.6.2.1. Método de Simulação.....	63
4.6.2.2. Metamodelos e <i>Design</i> Experimentais	65
4.6.2.3. Funções de Aquisição e Algoritmos de Otimização	68
4.6.2.4. Etapas do <i>Framework</i> de MAOvS Utilizado/Proposto	71

4.7.	Discussões e Futuras Direções.....	76
5.	FRAMEWORK DE METAMODELAGEM PROPOSTO	79
5.1.	Indicação, Requisitos e Benefícios do <i>Framework</i> Proposto	79
5.2.	Apresentação do <i>Framework</i>	80
5.2.1.	Fase de Inicialização	81
5.2.1.1.	Etapa 1: <i>Design</i> Inicial	81
5.2.1.2.	Etapa 2: Simulação do <i>Design</i> Inicial	82
5.2.1.3.	Etapa 3: Otimização dos Hiperparâmetros	82
5.2.2.	Fase Adaptativa	83
5.2.2.1.	Etapa 4: Treinamento do Metamodelo	83
5.2.2.2.	Etapa 5: Otimizar Função de Aquisição.....	83
5.2.2.3.	Etapa 6: Avaliar Solução no Modelo de Simulação.....	84
5.2.2.4.	Etapa 7: Incremento da base de treinamento.....	85
5.2.2.5.	Critério de Parada.....	85
5.3.	Implementação e Validação do <i>Framework</i>	85
6.	APLICAÇÃO DO FRAMEWORK PROPOSTO	87
6.1.	Objeto de Estudo 01: Manufatura Têxtil	87
6.1.1.	Resultados OE1: AMSO vs. EGO com 80 experimentos	90
6.1.2.	Resultados OE1: AMSO vs. EGO com 160 experimentos	92
6.1.3.	Resultados OE1: AMSO vs. AG	94
6.1.4.	Comparativo das Funções de Aquisição para o OE1	95
6.1.5.	Comparação dos Critérios de Parada para o OE1	96
6.2.	Objeto de Estudo 02: Mineração	97
6.2.1.	Resultados OE2: AMSO vs. EGO com 70 experimentos	99
6.2.2.	Resultados OE2: AMSO vs. AG	101
6.2.3.	Comparativo das Funções de Aquisição para o OE2	102
6.3.	Objeto de Estudo 03: Hospital Temporário	103
6.3.1.	Resultados OE3: AMSO vs. EGO com 100 experimentos	106
6.3.2.	Resultados OE3: AMSO vs. AG	107
6.3.3.	Comparativo das Funções de Aquisição para o OE3	108
7.	CONCLUSÕES.....	109
7.1.	Verificação dos Objetivos Específicos	109
7.2.	Considerações Finais	110

REFERÊNCIAS BIBLIOGRÁFICAS.....	114
ANEXO A – Detalhamento da Aplicação do <i>Framework</i>	138
APÊNDICE A - Publicações	148
i. Periódicos	148
ii. Capítulo de Livro	148
iii. Congressos	149

1. INTRODUÇÃO

1.1. Contextualização

A Simulação a Eventos Discretos (SED) é uma metodologia consagrada para representar computacionalmente sistemas complexos e dinâmicos, nos quais eventos individuais ocorrem em instantes discretos de tempo e interagem de forma estocástica, imitando o comportamento real do sistema (Basaglia et al., 2022; Law, 2013). Essa abordagem se mostra fundamental na análise de sistemas que operam em ambientes dinâmicos e incertos, proporcionando *insights* valiosos para a tomada de decisões. A SED é aplicável em uma variedade de contextos, como logística, mineração, saúde, sistemas militares, manufatura, entre outros (dos Santos *et al.*, 2022).

A SED permite que os pesquisadores e profissionais analisem o desempenho do sistema, identifiquem gargalos, avaliem estratégias de operação e tomem decisões fundamentadas com base em evidências. Além disso, a SED oferece a flexibilidade de experimentar diferentes cenários, condições e políticas sem interferência física no sistema de produção, contribuindo para uma compreensão mais abrangente e robusta do sistema em questão, reduzindo o risco na tomada de decisão (Banks *et al.*, 2014).

De acordo com Barlas e Heavey (2016), Rodič (2017) e Santos *et al.* (2023), embora amplamente adotada, nota-se que muitas organizações ainda possuem dificuldade em utilizar os benefícios da SED devido às limitações relacionadas ao tempo e à complexidade envolvida nos projetos de simulação. Neste caso, estes autores discutem que a simulação vem passando por mudanças significativas para se adaptar às exigências da indústria moderna, especialmente aos princípios da Indústria 4.0, que exige modelos de simulação capazes de apoiar tomadas de decisão mais ágeis e confiáveis.

A análise de sistemas de produção complexos muitas vezes envolve diversas variáveis de decisão e objetivos a serem alcançados, resultando na avaliação de inúmeros cenários. Neste caso, Wang *et al.* (2023) destacam a importância das técnicas de Otimização via Simulação (OvS) na identificação do melhor cenário. A OvS desempenha um papel crucial na otimização de sistemas complexos, nos quais a função objetivo (FO) e as restrições estão diretamente relacionadas a um modelo de simulação, possibilitando uma análise mais aprofundada e precisa do comportamento do sistema (Belgin, 2019).

A integração da otimização com a simulação oferece uma perspectiva valiosa para aprimorar o desempenho de sistemas dinâmicos e estocásticos. Essa abordagem é

especialmente útil quando a complexidade do sistema dificulta a formulação de modelos analíticos completos e precisos. A simulação fornece a estrutura para avaliar o sistema em diferentes cenários, enquanto a otimização permite a busca pelos melhores conjuntos de decisões ou parâmetros que maximizem ou minimizem a FO (Fu, 2002). A OvS possibilita a otimização de processos, o projeto eficiente de sistemas, a alocação de recursos e a tomada de decisões estratégicas, contribuindo para a maximização de eficiência, redução de custos e em melhorias substanciais no desempenho dos sistemas estudados (Sousa Junior et al., 2019a).

No entanto, a execução de simulações pode ser computacionalmente intensiva, principalmente quando se utilizam modelos complexos, o que pode tornar o processo de OvS moroso quando o tempo de tomada de decisão é uma restrição. Para superar esse problema, muitos autores sugerem elevar o problema a um nível superior de abstração, conhecido como metamodelagem (Chabanet; Bril El-Haouzi; Thomas, 2021; Kleijnen, 2017a; Parnianifard et al., 2023a).

A metamodelagem visa estabelecer e estimar uma relação entre as entradas e saídas de um modelo de simulação, criando um modelo simplificado utilizado para avaliar soluções potenciais durante o processo de otimização (Rojas-Gonzalez; Van Nieuwenhuysse, 2020). Uma vez treinado e validado, o metamodelo pode substituir o modelo de simulação original no processo OvS, reduzindo consideravelmente o tempo computacional para convergência da otimização (Parnianifard et al., 2020a). Na Figura 1 é apresentado o processo de OvS convencional (a) e o processo de OvS com a substituição do modelo de simulação pelo metamodelo (b).

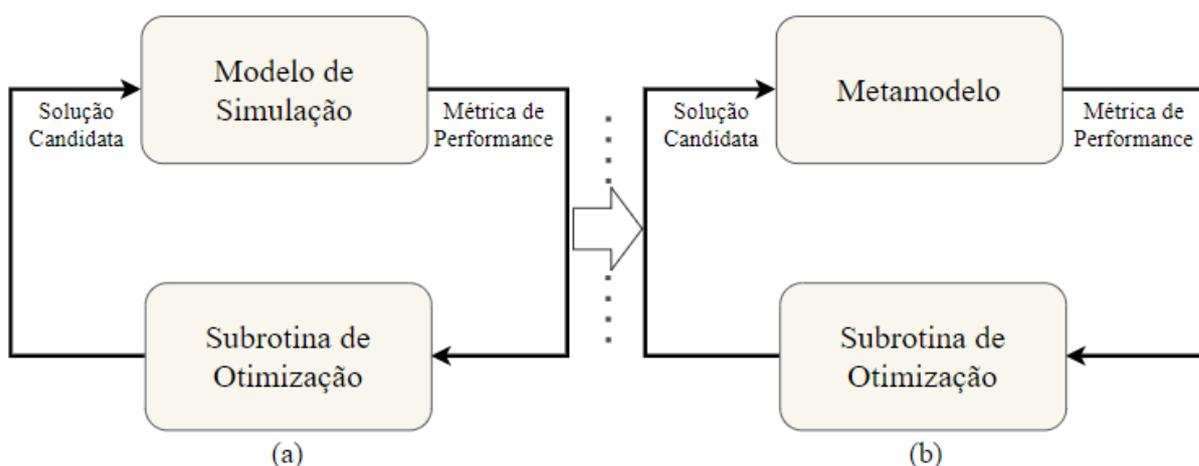


Figura 1: Processo de OvS baseada em modelos (a) e baseada em metamodelos (b)

Fonte: Adaptado de Amaral, Montevechi e Miranda (2022).

Neste contexto, diversos estudos propuseram o desenvolvimento de técnicas de metamodelagem mais precisas e eficientes que visam alcançar bons resultados com o mínimo

de execuções de simulação. Exemplos incluem os trabalhos de Moghaddam e Mahlooji (2017), Pang *et al.* (2022), Parnianifard *et al.* (2018), Amaral *et al.* (2022b) e Amaral *et al.* (2022d).

As abordagens de metamodelagem podem ser categorizadas como projeto experimental fixo (ou *one-shot*) ou metamodelagem adaptativa (MA) (ou *infill-search*) (Liu; Ong; Cai, 2018). No projeto fixo, uma única base experimental de tamanho w é escolhida no início do projeto, e o metamodelo é treinado exclusivamente neste conjunto (Amaral *et al.*, 2022a). No entanto, esta abordagem requer muitos experimentos para cobrir o espaço de soluções e treinar um metamodelo preciso. Por outro lado, considerando um *budget* de simulação limitado, a estratégia adaptativa utiliza mecanismos que visam alocar os experimentos em regiões mais promissoras.

A MA se baseia na construção iterativa de metamodelos, que se ajustam e refinam ao longo do processo de otimização. Inicialmente, são criados metamodelos simplificados a partir de uma quantidade limitada de experimentos. Conforme a otimização avança, mais experimentos são coletados e os metamodelos são aprimorados, incorporando informações adicionais e refinando sua representação do comportamento do modelo de simulação original. Esse ciclo iterativo de construção, avaliação e atualização dos metamodelos, o capacita a se adaptar às complexidades do sistema, melhorando a precisão das previsões e, conseqüentemente, a eficiência do processo de otimização (Rojas-Gonzalez; Van Nieuwenhuyse, 2020).

A estratégia de MA destaca-se por sua eficiência para otimização de modelos complexos, nos quais a execução das simulações demanda um elevado tempo computacional. Exemplos de aplicação pode ser observado em diversos trabalhos (Candelieri *et al.*, 2022; Coelho; Pinto, 2018; Pedrielli; Wang; Ng, 2020; Zheng; Li, 2023). Embora a estratégia de MA tenha apresentado resultados consistentes na literatura, a evolução no campo de *machine learning* (ML) abre espaço para o desenvolvimento de abordagens cada vez mais eficazes na solução de problemas de OvS, principalmente quando se trata de modelos de SED.

A fim de contribuir para esta problemática e promover maior eficiência na otimização de modelos de SED, esta tese propõe um novo *framework* de metamodelagem integrando, de forma inovadora, conceitos de ML, *Design of Experiments* (DoE) e meta-heurísticas .

1.2. Objetivos

Esta pesquisa tem como objetivo desenvolver, aplicar e validar um *framework* para OvS, denominado *Adaptive Metamodeling-based Simulation Optimization* (AMSO), que incorpora

de forma inovadora os conceitos de metamodelagem adaptativa, *Bagging*, *Gradient Boosted Trees* (GBT), otimização de hiperparâmetros (OHP) e meta-heurísticas para solução de problemas de otimização em modelos de SED, reduzindo o tempo computacional e permitindo a tomada de decisão com um *budget* restrito de simulações. Nesse contexto, os objetivos específicos desta tese abrangem:

- Conduzir uma revisão sistemática de literatura (RSL) para analisar o estado da arte em metamodelagem adaptativa em OvS, identificando as oportunidades e lacunas da literatura a fim de embasar a proposição do framework AMSO;
- Implementar o *framework* AMSO em um ambiente de programação (*Python*) em uma estrutura integrada ao *software* de simulação adotado (*FlexSim*[®]), aplicando-o em três problemas reais de SED.

1.3. Motivação

Em uma RSL publicada recentemente, Amaral *et al.* (2022a) observaram que cerca de 40% dos estudos utilizaram uma estratégia adaptativa na literatura de otimização de simulação baseada em metamodelagem. No *framework* geral de MA, assim como apresentado na Figura 2, uma pequena base experimental inicial é gerada e o metamodelo é treinado preliminarmente sobre esta base. Considerando a otimização de uma função de aquisição pré-definida, novos pontos são escolhidos, simulados e adicionados à base de treinamento. Após isto, o metamodelo é retreinado com a base experimental atualizada. Este ciclo se repete até que o critério de parada seja atendido (Liu; Ong; Cai, 2018).

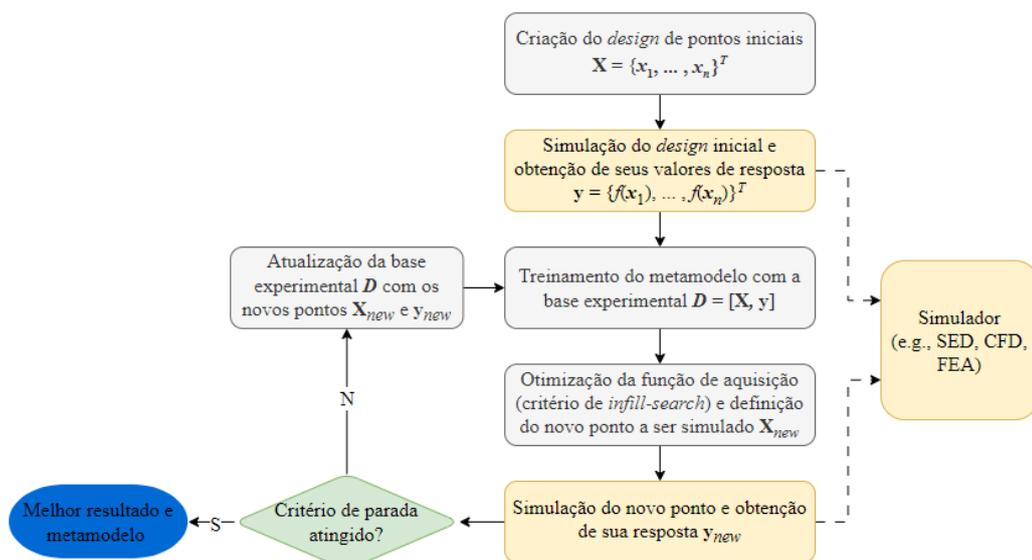


Figura 2: *Framework* geral para metamodelagem adaptativa

Fonte: Adaptado de Liu, Ong e Cai (2018)

Diversos *frameworks* para MA foram propostos na literatura visando maior eficiência do processo de otimização. Um dos *frameworks* precursores e mais utilizados de MA é o EGO. O EGO foi inicialmente proposto por Jones *et al.* (1998) e é baseado na metamodelagem por *Kriging* e utiliza principalmente a FA denominada *Expected Improvement* (EI). O EGO, comumente referenciado como *Bayesian Optimization*, ainda é considerado um dos principais algoritmos de MA atualmente e vem sendo utilizado em diversos estudos de OvS e ML (Briffoteaux *et al.*, 2020; Morales-Hernández; Van Nieuwenhuysse; Rojas Gonzalez, 2022; Zhan *et al.*, 2023).

O sucesso do EGO levou diversos autores a focarem seus trabalhos no desenvolvimento de abordagens de MA, muitos deles adaptando *frameworks* existentes para tratativa de uma classe de problemas específica. Este é o caso de Luo *et al.* (2022), que propuseram uma abordagem de metamodelagem adaptativa bi-nível associada a *trust-region* e AG para um problema de contaminação de águas subterrâneas. Bamdad *et al.* (2020) desenvolveram uma estrutura de treinamento adaptativo para otimização da eficiência energética em edifícios. Nesta abordagem é criado um *Ensemble (Committee)* no qual L modelos de Redes Neurais Artificiais (RNA) são treinados com diferentes inicializações de seus pesos e cada um prevê o valor da FO para os demais pontos do espaço de soluções do problema, calculando-se a média e variância dessa previsão. Outros trabalhos também se destacaram por: adaptar *frameworks* de MA para otimização robusta (Ur Rehman; Langelaar, 2016; Wang; Yuan; Ng, 2020); propor novas FAs para considerarem a probabilidade de viabilidade dos novos pontos candidatos a serem avaliados pela simulação (Lee; Zhou; Chen, 2023; Wang; Ierapetritou, 2018); e estratégias para avaliar múltiplos pontos a cada iteração do algoritmo, potencializando o uso de paralelismo em máquinas *multi-core* (Yaohui, 2017).

Estes e outros trabalhos destacados ao longo do Capítulo 4 convergem para duas características comuns de problemas de OvS tratados pelas técnicas de MA: modelos que exijam um tempo computacional elevado para execução; e projetos com restrição de tempo (ou número de execuções). Ambas as características estão presentes nos estudos de SED, destacando a MA como uma abordagem promissora para este tipo de problema. Amaral *et al.* (2022a) ressaltam que, em sua RSL, foram encontrados apenas dez trabalhos que utilizaram uma estratégia de MA para otimização de modelos de SED, enquanto outros 31 artigos de SED utilizaram a abordagem *one-shot*. Isto resalta que, embora a MA possua diversas vantagens e potencialidades para OvS, ainda é pouco explorada na literatura de SED.

Os trabalhos de MA em SED utilizaram *Kriging*, RNAs e Regressão Polinomial para treinamento incremental de metamodelos para solução de problemas típicos de SED, como

alocação de recursos, controle de estoque e programação de produção (Chang; Hsieh, 2016; Coelho; Pinto, 2018; Cuckler; Chang; Hsieh, 2017; Moghaddam; Mahlooji, 2017; Mohammad Nezhad; Mahlooji, 2014; Quan *et al.*, 2015; Xiao; Lee; Chen, 2015; Zhang *et al.*, 2017).

Embora esses artigos tenham apresentado resultados promissores e consistentes, este ainda é um campo de pesquisa latente e em constante desenvolvimento. Nesse sentido, este trabalho propõe um novo *framework* de metamodelagem para problemas de otimização em SED. Esta abordagem se baseia em lacunas encontradas na literatura e técnicas do estado da arte em MA, evidenciadas na RSL apresentada no Capítulo 4. Desta forma, o *framework* proposto combina de forma inovadora as técnicas de *Latin Hypercube Design* (LHD), *Bagging*, GBT, otimização de hiperparâmetros, metaheurística e uma nova FA.

Neste *framework*, o LHD é o responsável pela geração da base experimental inicial, o metamodelo será constituído por um conjunto de modelos GBT treinados e agregados pela técnica de *Bagging*. A associação das técnicas de *Bagging* e GBT visa a redução da variância e vieses do metamodelo, respectivamente. Além disto, visando melhorar o ajuste do metamodelo à base experimental, a técnica OHP é utilizada para encontrar os hiperparâmetros (HP) ótimos para o GBT durante a inicialização do *framework*. Ademais, este trabalho ainda propõe uma nova FA, denominada de *Balanced Expected Improvement* (BEI), que estabelece uma alternância estocástica entre busca global e local (com base no ajuste do metamodelo) para guiar o processo de *infill-search*. Por fim, a otimização da FA será realizada via AG.

A fim de demonstrar sua aplicabilidade e desempenho, o *framework* proposto será aplicado para otimização de dois objetos de estudo reais. O primeiro é um modelo SED que funciona como um gêmeo digital para a alocação periódica de recursos em uma indústria têxtil. O segundo caso representa um problema de expansão de capacidades de uma planta de mineração/moagem de grande porte. Além disso, os métodos EGO e AG serão aplicados como *benchmarking* nos problemas apresentados.

1.4. Contribuições

Esta tese expressa sua contribuição à literatura em três pontos principais. A primeira contribuição relaciona-se à análise e discussão sistemática do conhecimento difundido em trabalhos específicos da área, na perspectiva e rigor científico da metodologia de RSL. Esta RSL traz novas perspectivas que subsidiarão pesquisas futuras e consequentemente ampliarão a literatura e aplicações de técnicas de metamodelagem adaptativa para otimização via simulação (MAOVs). Como resultado, os profissionais terão um guia com métodos e

abordagens que poderão ser adotados em seus projetos de MA, bem como uma linha de base para suas ações e decisões. Além disso, pesquisadores podem se beneficiar deste trabalho mediante uma visão holística dos principais conceitos que envolvem a MAOvS.

A segunda contribuição refere-se ao desenvolvimento de um *framework* inovador para metamodelagem adaptativa, que integra, de maneira pioneira, metamodelagem pelos algoritmos *Bagging* e GBT, otimização de hiperparâmetros, e Algoritmo Genético para a resolução de problemas de otimização em simulação a eventos discretos. Além disto, esta pesquisa propõe uma nova função de aquisição, chamada de *Balanced Expected Improvement*, que permite, a cada iteração do algoritmo, o balanceamento entre busca local e global com base na precisão do metamodelo.

A terceira contribuição é a implementação do *framework* proposto em um ambiente *open-source* com conexão direta ao ambiente de simulação (*software*), permitindo execuções do modelo em paralelo e maior eficiência no processo de OvS.

Ademais, esta tese mostra a eficácia do *framework* proposto por meio de sua aplicação em três problemas reais da indústria. Adicionalmente, este trabalho se destaca pela sua contribuição tecnológica ao apresentar um método passível de ser empregado na resolução de desafios complexos no contexto da indústria e serviços 4.0. Dessa forma, vislumbra-se que esta pesquisa estabeleça uma base sólida para que pesquisadores e profissionais possam explorar o potencial da integração entre metamodelagem adaptativa e simulação a eventos discretos.

1.5. Delimitação da Pesquisa

Nesta pesquisa será proposto um *framework* para MA em problemas de SED, no qual será incorporado conceitos de ML, meta-heurísticas, otimização de hiperparâmetros e DoE. Ressalta-se que não é escopo nem objetivo deste trabalho discutir sobre demais métodos de otimização empregados em OvS, tais como meta-heurísticas e heurísticas, detalhar, comparar ou incrementar a formulação matemática dos algoritmos de ML, analisar os diversos algoritmos para otimização de hiperparâmetros, comparar sistematicamente arranjos de DoE ou parâmetros envolvidos no AG.

Esta tese apresenta a aplicação e validação do *framework* proposto em problemas típicos de SED, envolvendo a otimização da alocação de recursos, o que representa uma importante classe de problemas na qual a SED é empregada (Sousa Junior *et al.*, 2020). Ressalta-se ainda que o *framework* AMSO foi desenvolvido para problemas com restrições ligadas diretamente às variáveis de decisão, logo, problemas que apresentem restrições relacionadas à outras

métricas de resposta (*outputs*) do modelo não são tratáveis pelo AMSO, sendo recomendado a utilização de outras abordagens de metamodelagem disponíveis na literatura (Lee; Zhou; Chen, 2023; Wang; Ierapetritou, 2018).

Além disto, destaca-se que os objetos de estudo utilizados nesta tese são casos reais desenvolvidos em projetos de pesquisa/consultoria. Portanto, não é objetivo desta tese discutir aspectos construtivos deles, tais como etapas de modelagem conceitual e computacional, validação, escolha das variáveis de decisão e definição da FO, já que esses aspectos foram definições conjuntas da equipe do projeto em acordo com premissas e escopo previamente definidos, não sendo passíveis de manipulação por parte do autor.

Vale ainda ressaltar que o *framework* foi implementado em ambiente *Python* e que, para sua utilização na otimização de modelos SED, o software de simulação precisa permitir comunicação entre o ambiente de modelagem e simulação e o ambiente *Python* (alteração dos valores das variáveis de decisão, leitura dos resultados da simulação, envio de comandos de controle, tais como executar e parar a simulação, entre outros). Esta pesquisa utilizou o *software* FlexSim para modelagem e simulação, todavia outros *softwares* que permitam esta comunicação modelo/*Python* podem ser utilizados pelo usuário, desde que com os devidos ajustes da interface.

É importante destacar que o AMSO é destinado a otimização de modelos de SED complexos (que demandam elevado tempo computacional para execução) e em situações de restrição de tempo e número de experimentos, o que representa a realidade da maioria dos projetos práticos de modelagem e simulação. Em situações em que o modelo seja rápido de se executar ou o tempo demandado para otimização não seja uma restrição do projeto, a literatura apresenta diversos métodos que apresentam bons resultados, tais como AG, *Particle Swarm Optimization*, *Tabu Search*, *Simulated Annealing*, entre outros (Amaran *et al.*, 2016).

Por fim, destaca-se que todas as aplicações apresentadas nesta tese foram desenvolvidas em um computador com processador Intel i7-3770, com 4 *cores*, 3.40 GHZ, 24GB de RAM, placa gráfica GPU NVIDIA GeForce GTX 660 e sistema operacional Windows, plataforma 64 bits.

1.6. Estrutura do Trabalho

O restante deste documento está dividido em cinco capítulos adicionais. O Capítulo 2 apresenta a base teórica sobre a qual esta pesquisa se baseia. O Capítulo 3 discute o método de pesquisa seguido por este trabalho. O Capítulo 4 aborda a RSL sobre o tema desta tese. O

Capítulo 5 descreve o *framework* proposto, enquanto o Capítulo 6 investiga sua aplicação nos objetos de estudo. Por fim, o Capítulo 7 apresenta as conclusões do trabalho e sugestões de pesquisas futuras.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão introduzidos os conceitos fundamentais relativos a SED, OvS e metamodelagem. É importante destacar que o embasamento teórico exposto neste Capítulo serve como o alicerce sobre o qual esta tese foi elaborada, sustentando os resultados e conclusões alcançados por meio deste estudo.

2.1. Simulação a Eventos Discretos

Conforme destacado por Banks et al. (2014), a SED é uma ferramenta de modelagem que se propõe a reproduzir sistemas complexos de forma computacional, permitindo a avaliação e análise de mudanças sem intervenções no mundo real. Além disto, a utilização da SED reduz o risco associado às decisões, já que é possível mensurar melhorias no sistema antes de qualquer investimento ser feito (Campos *et al.*, 2023a). Essa abordagem pode ser aplicada em diversos domínios, como manufatura, logística, assistência médica, cadeias de suprimentos e outros, com um objetivo comum: fornecer aos responsáveis pela tomada de decisões uma compreensão realista dos processos (Taleb *et al.*, 2023).

De acordo com a definição de Banks et al. (2014), a SED se refere à modelagem de um sistema que progride ao longo do tempo, no qual mudanças de estado das variáveis ocorrem em momentos discretos. Desde sua criação nos anos 1950, a SED tem sido empregada com êxito na otimização de processos, oferecendo uma ferramenta de modelagem e análise para sistemas complexos que são difíceis de se tratar analiticamente (Greasley; Owen, 2018; Miranda *et al.*, 2017).

Banks *et al.* (2014) afirmam que um modelo computacional, após ser validado, capacita os tomadores de decisões a examinarem o comportamento do sistema ao longo do tempo, possibilitando responder a questões do tipo: E se.... Esses autores também citam como vantagens da SED a capacidade de obter análises de partes específicas do sistema, o que pode resultar na simplificação de sua complexidade.

Law (2013) enfatiza que a SED é uma alternativa de custo reduzido em comparação com experimentações reais, evitando interrupções na produção e custos de implementação no ambiente de produção. Além disso, a SED permite que analistas explorem diversos cenários, buscando a melhor configuração do sistema sob estudo. Entretanto, ao abordar sistemas complexos com múltiplas opções de configuração, torna-se imperativo incorporar técnicas de otimização em conjunto com a simulação, visando a identificação do *design* mais eficaz para o sistema (Amaral et al., 2022b).

Segundo Hoad *et al.* (2010), o conceito de replicação em SED é essencial para garantir a validade e a precisão dos resultados obtidos. Em uma simulação, eventos ocorrem em pontos específicos no tempo, e o comportamento do sistema é analisado ao longo dessas ocorrências. Devido à natureza aleatória e variabilidade intrínseca dessas simulações, uma única execução pode não fornecer uma visão completa ou precisa do desempenho do sistema. Portanto, realizar múltiplas replicações, ou seja, executar a simulação várias vezes sob as mesmas condições iniciais, é crucial para capturar essa variabilidade e obter estimativas confiáveis das métricas de interesse. A importância das replicações reside na capacidade de avaliar a variabilidade dos resultados, identificar tendências consistentes e calcular intervalos de confiança, proporcionando uma base sólida para a tomada de decisões.

De acordo com Law (2013), o cálculo envolve a execução repetida da simulação e a análise estatística dos resultados para determinar medidas de tendência central (como a média) e dispersão (como o desvio padrão), permitindo uma compreensão mais robusta do comportamento do sistema simulado. A resposta final da simulação, $f(\mathbf{x})$, é normalmente calculada sobre a média das respostas encontradas nas r replicações, assim como apresentado na Eq. (1).

$$f(\mathbf{x}) = \frac{1}{r} \sum_{i=1}^r f_r(\mathbf{x}) \quad (1)$$

Montevecchi *et al.* (2010) delineiam as etapas de um projeto de SED, conforme ilustrado na Figura 3. A primeira fase envolve a concepção do modelo conceitual, na qual se define e descreve o objeto de estudo. A segunda fase compreende a implementação e validação do modelo computacional, e, por fim, a etapa denominada Análise consiste na experimentação e avaliação dos resultados do modelo, frequentemente recorrendo ao uso de técnicas de OvS.

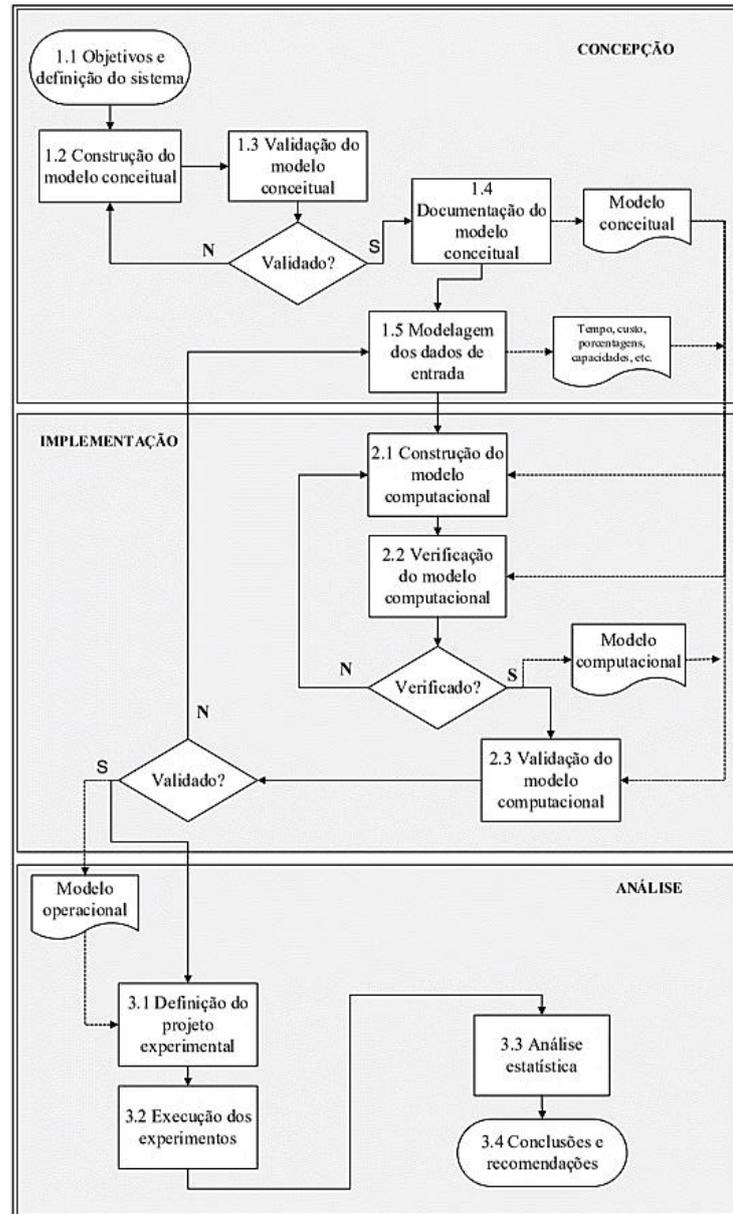


Figura 3: Etapas de um projeto de simulação a eventos discretos

Fonte: Adaptado de Montevechi *et al.* (2010)

2.2. Otimização via Simulação (OvS)

De acordo com Wang e Demeulemeester (2022), a OvS emerge como uma ferramenta poderosa no âmbito da tomada de decisões. Esta abordagem tem a capacidade de capturar as interações entre diversas variáveis em sistemas complexos, resultando na identificação da solução mais adequada para desafios em questão. A OvS é amplamente empregada para guiar os tomadores de decisão no investimento e alocação eficaz de recursos, especialmente em sistemas de produção, sejam eles projetados ou já existentes (Sousa Junior *et al.*, 2019a).

A OvS pode ser definida como o processo de encontrar o valor ótimo de uma FO sujeita a restrições, cuja avaliação é executada por meio de simulações estocásticas, ou seja, visa identificar os valores de entrada ótimos para variáveis de decisão em um sistema simulado. Assim como ilustrado na A Figura 4, esse procedimento é viabilizado por um ciclo iterativo que interliga o algoritmo de otimização e o modelo de simulação (Wang; Demeulemeester, 2022).

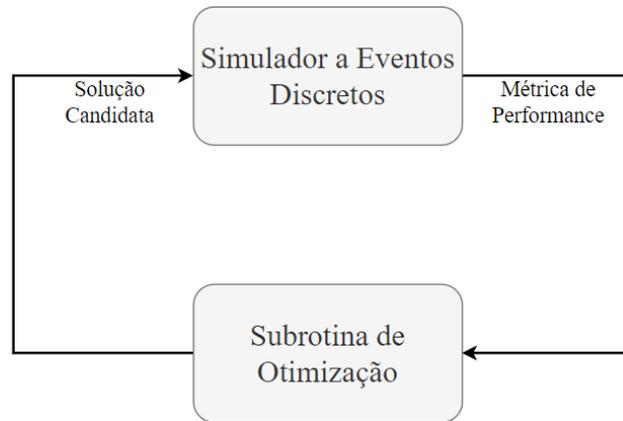


Figura 4: Processo de OvS

Fonte: Adaptado de Fu (2002)

Nos últimos anos, o campo de estudo da OvS passou por avanços significativos. Diversos algoritmos, softwares e aplicações foram desenvolvidos, impulsionando sua evolução. Nesse sentido, Amaran *et al.* (2016) salientam que a formulação geral de um problema de OvS é representada pela Eq. (2).

$$\begin{aligned}
 & \text{Min } \mathbb{E}_{\omega}[f(\mathbf{x}, \mathbf{y}, \omega)] \\
 & \text{s. a. } \mathbb{E}_{\omega}[g(\mathbf{x}, \mathbf{y}, \omega)] \leq 0 \\
 & \quad h(\mathbf{x}, \mathbf{y}) \leq 0 \\
 & \quad \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \\
 & \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{D}^m
 \end{aligned} \tag{2}$$

No contexto apresentado, \mathbb{E}_{ω} denota o valor médio esperado da função f , cuja avaliação ocorre mediante simulação utilizando entradas contínuas \mathbf{x} ou discretas \mathbf{y} . Essas entradas são submetidas a um conjunto de valores aleatórios (sementes) representados por ω . Similarmente, as limitações impostas são delineadas pelos valores da função g , que são avaliados a cada iteração da simulação. Além disso, o problema pode conter restrições adicionais (representadas por h) que não estão vinculadas a variáveis aleatórias, bem como restrições associadas às variáveis de decisão.

De acordo com Moghaddam e Mahlooji (2017), os métodos OvS podem ser classificados como baseados em modelos ou baseados em metamodelos. Na abordagem baseada em modelo, a otimização escolhe quais soluções precisam ser avaliadas, e o modelo de

simulação as avalia. Por outro lado, na abordagem baseada em metamodelo, o modelo de simulação é substituído por um terceiro componente, denominado metamodelo. Segundo Sousa Junior *et al.* (2019a), meta-heurísticas e metamodelagem são os métodos mais utilizados em problemas de otimização de SED. Embora as meta-heurísticas sejam métodos eficientes e generalizados, elas geralmente demandam um alto tempo computacional para serem executadas (Miranda *et al.*, 2014). No entanto, os gestores precisam de utilizar ferramentas para apoiar as suas decisões em geral de forma rápida, por vezes em tempo real ou quase real (dos Santos *et al.*, 2022). Neste caso, as meta-heurísticas em otimização baseadas em modelos podem não convergirem em bons resultados em tempo hábil, sendo recomendado o uso de estratégia de metamodelagem (Parnianifard *et al.*, 2023b).

2.3. Definições Gerais de Metamodelagem

No processo de OvS, quanto maior a complexidade do sistema sob estudo, mais tempo é requerido para realizar a otimização. Neste caso, para obter resultados de qualidade dentro de prazos adequados, é necessário elevar o problema a um nível mais abstrato, conhecido como metamodelo (Parnianifard *et al.*, 2023b). A metamodelagem é uma abordagem frequentemente empregada para desenvolver um modelo representativo de um modelo de simulação. O metamodelo consegue capturar a relação entre os valores das variáveis de decisão e os resultados da simulação, oferecendo uma estimativa da FO em um tempo de execução significativamente menor que a simulação em si.

Metamodelos são aproximações matemáticas construídas com base em dados gerados por modelos de simulação complexos, com o objetivo de representar, de forma simplificada e eficiente, o comportamento do modelo de simulação original (Amaral *et al.*, 2022b; Liu *et al.*, 2022; Rojas-Gonzalez; Van Nieuwenhuyse, 2020). Logo, assim como apresentado na Figura 5, um modelo de simulação é uma abstração de um sistema real e, da mesma forma, um metamodelo é uma abstração de um modelo de simulação (um modelo de outro modelo), tornando as avaliações mais simples e rápidas, embora representem sistemas reais com menos precisão.

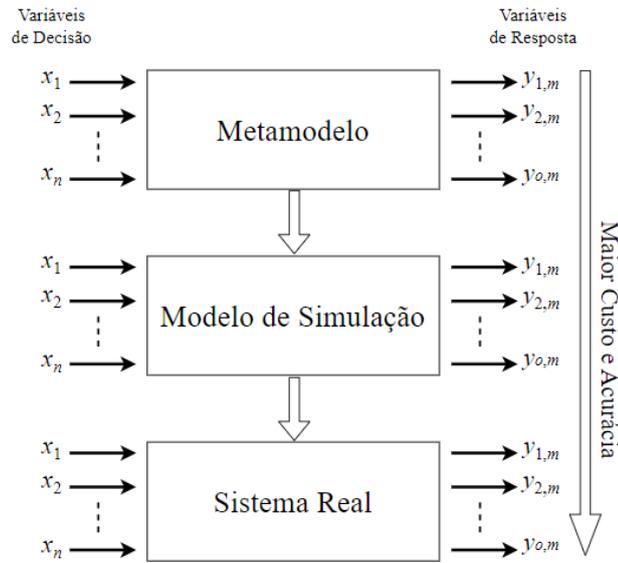


Figura 5: Relação entre cenários do mundo real, modelos de simulação e metamodelos

Fonte: Adaptado de Li *et al.* (2010)

Segundo Amaral *et al.* (2022a), os *frameworks* para metamodelagem podem ser divididos em duas categorias: a estratégia de *design* experimental fixo (*one-shot*) e estratégia de *design* incremental ou adaptativo (*infill-search*). A estratégia de experimentação fixa consiste em retirar um único conjunto de amostras de tamanho w no início do projeto, no qual o metamodelo é treinado exclusivamente com essas amostras. Exemplos de aplicação de metamodelagem *one-shot* podem ser encontrados em diversos trabalhos (Amaral *et al.*, 2022b; Lal; Datta, 2020; Parnianifard; Azfanizam, 2020; Sharif; Hammad, 2019).

Por outro lado, a estratégia adaptativa começa com uma amostra inicial de dados de tamanho ξ , na qual o metamodelo é treinado e otimizado para determinar a precisão e a direção da região em que se encontra o ótimo. Depois disso, a base de dados é incrementada com um novo conjunto de amostras de tamanho δ_i em cada iteração do algoritmo. O metamodelo é treinado novamente com a nova base experimental de tamanho $\xi_{i+1} = \xi_i + \delta_i$, onde i é o número de iterações do algoritmo (Razali *et al.*, 2020). A escolha dos pontos a serem simulados e acrescentados em cada iteração é definida pela otimização da FA. Mais detalhes sobre esta abordagem serão apresentados no Capítulo 3.

Em ambas as abordagens, o sucesso da aplicação está diretamente ligado as técnicas que serão utilizadas para construção do metamodelo. Dentre elas podem-se citar a escolha do arranjo experimental adequado, tamanho da base de treinamento, método de busca (algoritmo de otimização), e, principalmente, o algoritmo de metamodelagem (algoritmo de ML) e seus hiperparâmetros (Amaral *et al.*, 2022a). Neste sentido, diversos algoritmos de metamodelagem surgiram na literatura, visando criar metamodelos mais precisos, tais como Regressão Polinomial (Hassannayebi *et al.*, 2019; Raponi *et al.*, 2021), *Kriging* (Li *et al.*, 2021; Zhang *et*

al., 2022b), *Radial Basis Function* (RBF) (Christelis; Mantoglou, 2016; Urquhart; Ljungskog; Sebben, 2020), e RNAs (Dunke; Nickel, 2020; Renganathan; Maulik; Ahuja, 2021). Muitos artigos publicados nesta área utilizam um ou mais algoritmos de metamodelagem para resolver problemas específicos.

Como exemplo, Sousa Junior *et al.* (2019b) integraram 33 algoritmos de ML, duas meta-heurísticas e processamento paralelo para alocação ideal de recursos no chão de fábrica. Storti *et al.* (2019) utilizaram RNA combinada com AG para otimizar o projeto de turbinas eólicas. Osorio e Chong (2015) estudaram o controle de semáforos por meio de metamodelagem por Regressão Polinomial quadrática e algoritmo de *derivate-free trust region*. Coelho e Pinto (2018) aplicaram *Kriging* para minimizar o tempo médio de resposta em um sistema médico de emergência. Jung *et al.* (2017) usaram metamodelos *Kriging* para otimizar os parâmetros de processo de um moinho a quente e maximizar o seu rendimento.

Devido à grande variedade de algoritmos utilizados na literatura, alguns trabalhos tiveram como foco comparar e analisar a performance destes algoritmos para metamodelagem em problemas de OvS. Como exemplo, Li *et al.* (2010) compararam sistematicamente cinco metamodelos em termos de métricas de erro. Hachicha (2011) estudou os metamodelos RNA, Regressão Polinomial e DoE Fatorial para o problema de dimensionamento de lotes em um problema de cadeia de suprimentos sob encomenda. Villareal-Marroquín *et al.* (2013) testaram dois métodos de metamodelagem no problema de moldagem por injeção. A primeira metodologia utiliza metamodelos de Regressão Linear e Análise Envoltória de Dados, enquanto a segunda utiliza metamodelo *Kriging* em uma abordagem de MA. Østergard *et al.* (2018) realizam uma comparação de seis metamodelos treinados com diferentes tamanhos de amostra para redução de custos e de energia em um simulador de desempenho predial.

Mirfenderesgi e Mousavi (2016) compararam o valor da FO, o erro e as métricas de ajuste de quatro metamodelos aplicados no problema de alocação hídrica. Luo *et al.* (2019) analisam estratégias de amostragem baseadas em variações do LHD para metamodelagem usando *Kriging* em um problema de remediação de águas subterrâneas. Para o mesmo problema, Hou *et al.* (2019) estudaram modelos de conjunto de *Kriging*, *Support Vector Machine* (SVM) e *Extreme Learning Machine* para identificação de fontes de contaminação em águas subterrâneas. No mesmo contexto, Lal e Datta (2020) avaliaram o desempenho de metamodelos autônomos e suas combinações em modelos de conjuntos para otimização de bombeamento. Além disso, Parnianifard *et al.* (2020a) compararam quatro técnicas de amostragem e três metamodelos para problemas de otimização robusta de baixa dimensão (uma ou duas variáveis de decisão).

Em RSL realizada por Amaral *et al.* (2022b), *Kriging*, Regressão Polinomial, RNA, GBT, *Random Forest*, RBF e SVM foram identificados como os principais metamodelos utilizados em OvS. Com foco em metamodelagem com *design* experimental fixo para otimização de modelos SED, Amaral *et al.* (2022b) e Amaral *et al.* (2022c) compararam algoritmos de metamodelagem em diversos problemas de manufatura e concluíram que o algoritmo GBT obteve o melhor desempenho e flexibilidade, apresentando melhor relação entre métricas de erro, valor ótimo atingido, e tempo de treinamento.

Considerando estes resultados promissores do algoritmo GBT para metamodelagem em SED (considerando a abordagem de *design* experimental fixo) e que não foram encontrados artigos que utilizem este algoritmo associado a uma abordagem adaptativa (assim como discutido na RSL apresentada no Capítulo 4, este trabalho utilizará o GBT como algoritmo de metamodelagem no *framework* proposto.

Quando se trata de abordagens adaptativas, a definição da FA é um fator decisivo no sucesso da aplicação. Grande parte das FAs encontradas na literatura almejam o balanceamento eficaz entre busca local e busca global, ou seja, explorar regiões com alta probabilidade de conterem o ótimo (busca local) e regiões onde o metamodelo apresenta alta incerteza (busca global). Neste sentido é necessário que o metamodelo estime não somente o valor previsto da FO para o ponto avaliado ($\hat{f}(\mathbf{x})$) mas também a incerteza associada a esta previsão (muitas vezes determinada pelo desvio padrão de n previsões para o ponto \mathbf{x} , ou seja, $\sigma(\mathbf{x})$).

No caso de metamodelos estocásticos, como o *Kriging*, é fornecido uma estimativa pontual do valor para o ponto desejado e informações sobre a incerteza associada a essa estimativa por meio do desvio padrão da estimativa (Østergård; Jensen; Maagaard, 2018). De acordo com Bertolini *et al.* (2022), a estimativa da incerteza de previsão também pode ser calculada via métodos de *resampling*. Os métodos de *resampling* são abordagens estatísticas não paramétricas empregadas para oferecer estimativas empíricas da distribuição amostral de um parâmetro estatístico, considerando a incerteza e variabilidade que não podem ser adequadamente quantificadas por meio de um único modelo estatístico. Nestes métodos, uma amostra de m conjuntos de dados é gerada a partir da base de dados original e o valor de uma estatística (por exemplo, mediana, média, desvio padrão) é calculado para cada conjunto. Segundo os autores, os principais métodos de *resampling* existentes na literatura são o *Jackknife Resampling*, *Monte Carlo Resampling*, e *Bootstrapping Resampling*.

O método *Jackknife Resampling* é uma técnica estatística usada para estimar o viés e a variância de um estimador estatístico. Envolve deixar sistematicamente de fora um ponto de cada vez do seu conjunto total de dados, calcular o valor do estimador para cada subconjunto

e, em seguida, usar esses v (Parnianifard *et al.*, 2021) e a variância (Parnianifard *et al.*, 2021). O método de *Monte Carlo Resampling* é uma técnica estatística que gera repetidamente amostras dos dados originais, sem reposição, para avaliar a incerteza em estimativas ou modelos. Isso permite criar distribuições de resultados que refletem a variabilidade nos dados, útil para estimar intervalos de confiança e analisar situações complexas (Bertolini; Finch; Nehm, 2022; Nakatsu, 2021).

O *Bootstrapping Resampling* é uma técnica estatística usada para estimar a distribuição amostral de uma estatística por meio da geração repetida e aleatória de amostras com reposição a partir de um conjunto de dados. Esse processo gera múltiplos conjuntos de dados simulados, cada um com o mesmo tamanho do original, mas contendo alguns dados duplicados e ausentes (Mairinque *et al.*, 2022). O algoritmo de *Bootstrapping* gera m subconjuntos de *bootstrap* extraído aleatoriamente amostras independentes dos dados de treinamento originais D , com reposição. Cada amostra extraída mantém o tamanho do conjunto de dados original, preservando assim as propriedades estocásticas dos dados originais D .

As técnicas de *resampling* possuem diversas aplicações, que vão desde a simples estimativa de intervalo de confiança até desafios mais complexos, como regressão, análise de séries temporais, estatísticas Bayesianas, *cross-validation* em métodos de ML, entre outros. Estas técnicas também podem ser associadas ao conceito de *Ensembles*, resultando no algoritmo conhecido como Bagging, ou *Bootstrap Aggregating*, no qual cada subconjunto (gerado via *Bootstrapping Resampling*) é utilizado para treinar um modelo independente (chamado de *base-learner*). O valor previsto estimado final é composto pela agregação das previsões individuais de cada modelo (Tahraoui *et al.*, 2022). Pérez-Rodríguez, Fernández-Navarro e Ashley (2023) destacam que o *Bagging* tem o objetivo de aumentar a capacidade de predição dos modelos, promovendo uma redução significativa na variância do modelo agregado.

Com base nos pontos destacados acima, resultados alcançados em estudos prévios (Amaral *et al.*, 2022b, 2022c) e nas discussões promovidas na RSL (Capítulo 3), a abordagem apresentada neste trabalho utiliza o conceito de *Bagging* associado ao algoritmo GBT (atuando como *base-learner*), em uma abordagem conhecida como *Bagging-based Gradient Boosted Trees* (B-GBT).

Louk e Tama (2023) sugerem que o uso de GBT como *base-learner* para *Bagging* pode fornecer diversos benefícios. Em primeiro lugar, o *Bagging* pode ajudar a diminuir a variância geral do modelo, levando a uma maior precisão da previsão. Além disso, a natureza sequencial do GBT permite que os modelos subsequentes se concentrem na correção de erros cometidos por modelos anteriores, o que pode minimizar o viés de previsão. Por último, o *Bagging* pode

aumentar a robustez do modelo, mitigando o impacto de valores discrepantes ou dados desequilibrados no conjunto de dados original. Como resultado, a combinação destas duas técnicas pode reduzir a variância e o viés do modelo (Ganjisaffar; Caruana; Lopes, 2011).

2.3.1. *Bagging*

O algoritmo *Bagging* (*Bootstrap Aggregating*) é uma técnica *Ensemble* homogênea que combina vários modelos de ML, chamados de *base-learners*, treinados em conjuntos de dados amostrados aleatoriamente usando a técnica de *Bootstrapping* (Breiman, 1996). Segundo Chen *et al.* (2021), os modelos *Bagging* geralmente possuem melhor capacidade de predição do que seus *base-learners* individuais. Essa melhoria é atribuída à incorporação do conceito de decomposição, no qual o erro quadrado do modelo é dividido em três componentes: viés, variância e covariância. No *Bagging*, o erro quadrado é reduzido por meio do termo de covariância com a contribuição colaborativa de vários *base-learners* individuais que possuem baixa correlação entre si, reduzindo assim a taxa de erro global. A Figura 6 ilustra o diagrama esquemático do algoritmo *Bagging*.

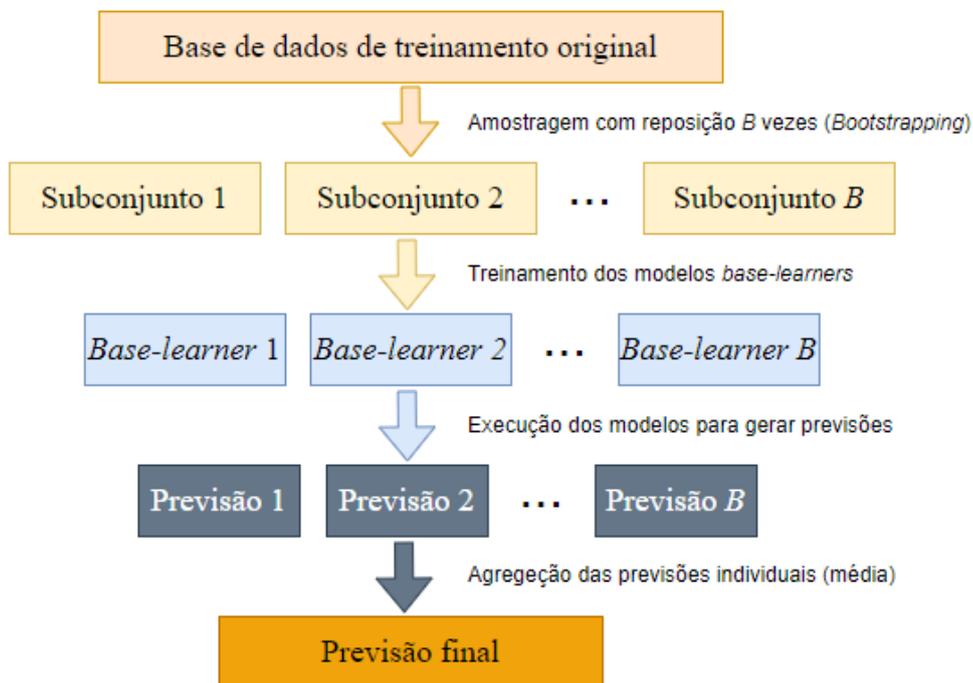


Figura 6: Diagrama esquemático do *Bagging Regression*

Fonte: Adaptado de Fan *et al.* (2022)

Segundo Fan *et al.* (2022), o procedimento geral do *Bagging* segue as seguintes etapas:

- 1) O conjunto de dados de treinamento original, D , é amostrado B vezes com reposição (via *Bootstrapping*), gerando B subconjuntos de treinamento independentes, cada um composto por n amostras;
- 2) Para cada subconjunto é treinado um modelo *base-learner*, denominado B_b ($b = 1, \dots, B$). Portanto, o modelo *Bagging* é composto por B modelos *base-learners*;
- 3) Após o treinamento, as previsões finais do modelo *Bagging* $\hat{f}(\mathbf{x})$ são determinadas agregando pela média as previsões de modelo *base-learner*, $\hat{f}^{*b}(\mathbf{x})$.

Hastie, Tibshirani e Friedman (2008) salientam que este processo gera uma previsão final que normalmente é mais precisa e estável do que as previsões feitas por modelos individuais. As Equações (3) e (4) definem a estimativa da previsão final e o desvio padrão da previsão do modelo *Bagging*.

$$\hat{f}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(\mathbf{x}) \quad (3)$$

$$\sigma = \frac{1}{B} \sum_{b=1}^B \left(\hat{f}(\mathbf{x}) - \hat{f}^{*b}(\mathbf{x}) \right)^2 \quad (4)$$

A capacidade de previsão de um modelo de ML pode ser avaliada de inúmeras formas, tais como separação da base de dados entre base de treino e validação, *k-fold cross-validation*, *leave-one-out cross-validation*, utilização de novos dados, entre outros (Amaral et al., 2022a). No caso do algoritmo *Bagging*, uma primeira camada de validação pode ser obtida utilizando a técnica de *Out-of-Bag* (OOB) (Ramosaj; Pauly, 2019; Schonlau; Zou, 2020). O OOB é uma maneira de estimar o desempenho do modelo *Bagging* sem a necessidade de um conjunto de validação separado.

Como detalhado anteriormente, ao criar cada subconjunto dos dados de treinamento, alguns pontos de dados são selecionados aleatoriamente várias vezes (com reposição), enquanto outros podem não ser selecionados em determinado subconjunto. Em média, cerca de 63,2% dos pontos de dados originais são incluídos em cada subconjunto, e os restantes aproximadamente 36,8% são deixados de fora. Para cada subconjunto, um modelo *base-learner* é treinado usando os pontos de dados selecionados no conjunto. Como alguns pontos de dados não estão incluídos em um subconjunto específico, eles são considerados *Out-of-Bag* para esse modelo. O erro OOB é calculado as previsões feitas por cada modelo nos pontos de dados *Out-of-Bag* com os valores-alvo reais (Cho; Jung; Hwang, 2019). O erro OOB é normalmente calculado como uma medida agregada de precisão das previsões em todos os modelos *base-*

learners, como o *Mean Square Error* (MSE), *Mean Absolute Error* (MAE), e o Coeficiente de Determinação (R^2) (Chen *et al.*, 2021).

2.3.2. Gradient Boosted Trees

O *Boosting* foi inicialmente proposto por Schapire (1990) e tem como objetivo construir um modelo de alta precisão a partir da combinação de algoritmos de baixa capacidade de aprendizado, especialmente no caso do GBT, que utiliza árvores de decisão. A cada iteração do algoritmo, as árvores são treinadas sequencialmente usando uma amostra aleatória \tilde{N} dos dados de treinamento, sem reposição, na qual \tilde{N} é menor que N , sendo N o tamanho da base de dados de treinamento (Friedman, 2002). O algoritmo GBT é considerado um *Ensemble*, pois combina um conjunto de árvores de decisão para criação de um modelo de alta precisão. Todavia, o GBT se diferencia dos métodos de *Bagging*, uma vez que constrói modelos (árvores) sequencialmente, enquanto o *Bagging* é composto de modelos construídos em paralelo (Xia *et al.*, 2017).

Essa abordagem se baseia em conjuntos de árvores de decisão, na qual a cada iteração, uma nova árvore é gerada com base nos resíduos das árvores anteriores (Krauss; Do; Huck, 2017). Considerando um conjunto de dados $\{x_i, y_i\}$ de tamanho N , em que x_i representa os atributos de entrada e y_i é a variável de resposta, o objetivo do algoritmo é determinar a função $F(x)$ que descreve a relação entre x e y , minimizando a função de perda definida na Eq. (5). Segundo Xia *et al.* (2017), essa relação é aproximada de forma aditiva pela Eq. (6).

$$F(x) = \arg \min_f \Psi(y, F(x)) \quad (5)$$

$$F_m^*(x) = \sum_{m=0}^M \beta_m h(x, \theta_m) \quad (6)$$

Sendo β_m e θ_m os coeficientes e parâmetros ótimos para o modelo de base representado por $h(x, \theta_m)$, respectivamente. De acordo com Xia *et al.* (2017), Friedman (2002) e Nie *et al.* (2021), o procedimento de treinamento do GBT pode ser definido pelos seguintes passos:

- 1) Inicialização: O algoritmo começa com um modelo inicial simples, geralmente uma única árvore de decisão, que serve como *learner* fraco inicial. Este *learner* fraco é representado por uma função de previsão, geralmente denotada como $F_0(x)$;
- 2) Calcular os resíduos: Para cada exemplo de treinamento i , calcula-se os resíduos entre a resposta real y_i e a previsão atual do modelo $F_{m-1}(x_i)$. Os resíduos são denotados como $r_{im} = y_i - F_{m-1}(x_i)$, no qual m é o número da iteração atual;

- 3) Treinar um *learner* fraco: Treina-se um novo *learner* fraco (geralmente uma árvore de decisão pequena) para prever os resíduos r_{im} em relação às características de entrada \mathbf{x}_i . Denota-se esta árvore como $h(\mathbf{x}, \theta_m)$;
- 4) Determinar os coeficientes ótimos: Utiliza-se um procedimento de otimização (geralmente o *Gradient Descent*) para encontrar o melhor coeficiente β_m que multiplica as previsões do *learner* fraco $h(\mathbf{x}, \theta_m)$. Este coeficiente é calculado minimizando uma função de perda, que pode variar dependendo do problema (por exemplo, MSE para regressão ou *cross-entropy* para classificação), que pode ser descrita por $\beta_m = \operatorname{argmin}_{\beta} \sum_{i=1}^N L(y_i, F_{m-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i, \theta_m))$. Sendo L a função de perda e N é o tamanho da base de treinamento;
- 5) Atualizar o modelo global: Com o coeficiente β_m determinado, atualiza-se o modelo global F_m somando o produto do *learner* fraco $h(\mathbf{x}, \theta_m)$ com β_m ao modelo anterior, $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \beta_m h_m(\mathbf{x}, \theta)$;
- 6) Recorrência: Repita os passos 2 a 5 por um número definido de iterações (ou até que um critério de parada seja alcançado). Cada nova árvore é treinada para capturar os resíduos não explicados pelo modelo anterior, refinando assim a previsão global.

A previsão final do algoritmo GBT é dado por $F_m(\mathbf{x})$, que é uma combinação ponderada das previsões dos *learners* fracos treinados em todas as iterações do algoritmo. O processo de adição sequencial de *learners* fracos (árvores de decisão neste caso), otimização dos coeficientes e atualização do modelo global permite que o GBT capture relações complexas nos dados, tornando-o uma técnica poderosa para uma variedade de problemas de ML.

Durante o treinamento do modelo (ou metamodelo) GBT, a seleção de hiperparâmetros desempenha um papel crucial. Segundo Amaral *et al.* (2022b), os principais hiperparâmetros a serem ajustados no GBT estão comumente relacionados ao número de estimadores (1 – 100), à taxa de aprendizagem (0,05 – 0,30) e às amostras mínimas por folha (0,12 – 0,50). Devido à impraticabilidade de definir manualmente esses parâmetros, o emprego de técnicas de otimização é essencial para determinar a configuração ideal que maximiza o desempenho do algoritmo. Esta etapa é conhecida como otimização de hiperparâmetros e será melhor discutida na seção seguinte.

Há uma variedade de métodos de otimização disponíveis para a seleção de hiperparâmetros, incluindo AG, *Grid Search*, Otimização Bayesiana, *Randon Search*, *Hyperband*, *Particle Swarm*, entre outros (Yang; Shami, 2020). Neste trabalho, optou-se por utilizar o AG devido à sua rapidez de execução, capacidade de adaptação a diferentes tipos de

variáveis e não necessitar da definição de um ponto de partida inicial. Além disto, o AG foi o método de busca mais utilizado por artigos da área, conforme destacado na RSL do Capítulo 4.

2.4. Algoritmo Genético

Neste trabalho o AG será utilizado para dois propósitos principais: para OHP do metamodelo na inicialização do *framework* proposto; e como algoritmo de busca para otimização da FA. Estes propósitos serão melhor discutidos no Capítulo 5.

O AG é uma metaheurística que se inspira na teoria biológica da seleção natural. Sua concepção inicial é atribuída aos pesquisadores De Jong (1975) e Holland (1975). O AG é fundamentado no princípio da sobrevivência do mais forte, conceito evolutivo descrito por Darwin. Essa abordagem tem sido aplicada com êxito em uma variedade de problemas, abrangendo tanto a programação linear como a não linear (Apornak *et al.*, 2020; Giri *et al.*, 2013).

No AG, o espaço de busca é constituído por soluções candidatas, cada uma delas representada por um cromossomo. Cada cromossomo é composto por genes, os quais codificam os parâmetros de uma solução candidata, além de carregar um valor de aptidão, representada pela função de avaliação, que expressa a qualidade da solução em relação ao problema em questão. O principal objetivo do AG é facilitar a combinação dos genes de maneira a gerar novos cromossomos com valores da função de avaliação superiores (Sohail, 2023).

Segundo Storti *et al.* (2019), os operadores fundamentais do algoritmo genético são três: reprodução, cruzamento e mutação. A reprodução consiste no processo em que os parâmetros ou genes de um indivíduo são copiados, levando em consideração seus valores da função de avaliação. Nesse sentido, indivíduos com o valor da função de avaliação mais elevada tem uma probabilidade maior de gerar um ou mais descendentes para a próxima geração. Quando um parâmetro é selecionado para reprodução, ele cria uma réplica idêntica de si mesmo. Posteriormente, o operador de cruzamento assume a responsabilidade de combinar aleatoriamente os parâmetros recém-gerados. Por último, a mutação desempenha um papel secundário, uma vez que a taxa de mutação geralmente é baixa (Alabas; Altiparmak; Dengiz, 2002; Storti *et al.*, 2019). O pseudocódigo do AG clássico é apresentado no Algoritmo 1.

 Algoritmo 1: Algoritmo Genético Clássico

Input:

Tamanho da população, n
 Número máximo de iterações, MAX

Output:

Melhor solução global, \mathbf{x}_{best}

Inicialização:

Gerar população inicial de n cromossomos \mathbf{x}_i ($i = 1, 2, \dots, n$)
 Calcular o valor da função de avaliação para cada cromossomo \mathbf{x}_i na população inicial
 Definir contador $t = 0$

Enquanto ($t < MAX$)

 Selecione os cromossomos pais baseado no valor da função de avaliação
 Aplique a operação de *crossover* nos cromossomos pais selecionados (dada a probabilidade de *crossover*)
 Aplique a operação de mutação aos cromossomos filhos (dada a probabilidade de mutação)
 Calcule o valor da função de avaliação para cada cromossomo filho
 Substitua a população anterior pela nova população gerada
 $t = t + 1$

Fim do loop

Retorne a melhor solução, \mathbf{x}_{best}

Fim

 Fonte: Adaptado de Katoch, Chauhan e Kumar (2021) e Srinivas e Patnaik (1994)

O emprego de AG na otimização de hiperparâmetros tem sido evidenciado em vários estudos (Hou *et al.*, 2019; Hüllen *et al.*, 2020). A OHP envolve a exploração de diversas combinações potenciais dos parâmetros $\{a_1, \dots, a_i\}$ de um algoritmo de ML. O AG pode ser empregado como método de otimização para encontrar os parâmetros ótimos que minimizam o erro de previsão do metamodelo, que é calculado pela técnica de *cross-validation*.

A *k-fold cross-validation* é considerada uma das técnicas mais relevantes para avaliar métodos de regressão e classificação. Esse método divide de forma igual e aleatória os dados de treinamento em k partes, utilizando $k-1$ partes para treinar o algoritmo e a parte restante para testá-lo. Cada conjunto de treinamento e teste é repetido k vezes, variando a parte escolhida para teste em cada iteração (representada por k'_i , onde $i = \{1, \dots, k\}$). Posteriormente, a performance é calculada com base na média dos erros obtidos nas k iterações de validação, proporcionando uma métrica mais robusta do que aquela calculada com base em uma única observação (Bergmeir; Benítez, 2012).

2.5. *Latin Hypercube Design*

Uma etapa fundamental para criação de metamodelos é a definição da base de dados de treinamento inicial. Para isto é necessário definir as variáveis de decisão (*inputs*) e resposta (*outputs*), planejar o experimento (definição dos valores que cada variável de decisão assumirá em cada experimento), e executar os experimentos (realizar simulações com os valores das variáveis de decisão e obter o valor de resposta) (Kleijnen, 2017b).

Conforme destacado por Montevechi, Miranda e Daniel (2012), um experimento pode ser compreendido como a execução de uma série de testes nos quais são realizadas alterações propositalmente nas variáveis de entrada. O objetivo é identificar e analisar os efeitos dessas mudanças nas variáveis de resposta do sistema. Nesse contexto, o DoE representa um conjunto de métodos voltados para o planejamento e condução de experimentos.

A literatura oferece diversas técnicas de DoE, tais como o *orthogonal array* (Taguchi; Tsai, 1995) o *Central Composite Design* (Villarreal-Marroquín *et al.*, 2013), o planejamento fatorial (Dengiz; Bektas; Ultanir, 2006), entre outras abordagens (Montgomery; Runger, 2003). Com base na RSL apresentada no Capítulo 4, nota-se que o método de DoE mais comumente utilizado em projetos de metamodelagem é o *Latin Hypercube Design*. De acordo com Mckay, Beckman e Conover (2000), o LHD é uma técnica de DoE que se fundamenta no conceito de *space-filling*, onde, para n variáveis de entrada, são obtidos aleatoriamente m pontos amostrais dentro de m intervalos de igual probabilidade.

Conforme detalhado por Parnianifard *et al.* (2020a), os procedimentos do LHD podem ser descritos da seguinte maneira:

- 1) As faixas de valores para cada variável são subdivididas em m subfaixas de magnitude equiprováveis. O número de pontos amostrais m é uma escolha do pesquisador e, em geral, é maior do que o total de pontos gerados pelo CCD.
- 2) O próximo passo consiste em alocar os m pontos entre os limites inferiores e superiores de cada intervalo. Cada ponto, numerado de 1 a m , aparece apenas uma vez em uma linha e coluna do espaço amostral.

A Figura 7 apresenta quatro possibilidades (a – d) de arranjos LHD com 2 variáveis de decisão e sete pontos amostrais.

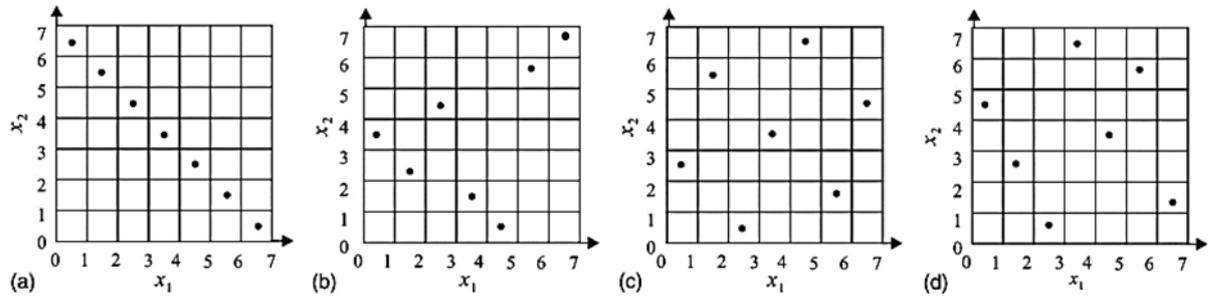


Figura 7: Exemplos de arranjos LHD

Fonte: Adaptado de Luo, Ji e Lu (2019)

O LHD é uma abordagem estatística eficaz e eficiente para a seleção de amostras representativas em experimentos. Quando aplicado à metamodelagem, o LHD oferece vantagens significativas. Em primeiro lugar, ele proporciona uma distribuição uniforme e estratégica das amostras, otimizando a representação do espaço de soluções e, conseqüentemente, melhorando a precisão do metamodelo. Além disso, a organização eficiente das amostras também facilita a análise estatística, contribuindo para a interpretação precisa e confiável dos resultados do modelo (Kleijnen, 2017b).

3. MÉTODO DE PESQUISA

3.1. Classificação da Pesquisa

De acordo com Miguel *et al.* (2018), pesquisas científicas no âmbito da engenharia de produção podem ser classificadas quanto às dimensões natureza, objetivos, abordagem e método, assim como ilustrado na Figura 8. No caso desta pesquisa, ela se enquadra na categoria de pesquisa aplicada, pois possui interesse prático, ou seja, resultados passíveis de aplicação para solução de problemas reais (Turrioni; Mello, 2012).

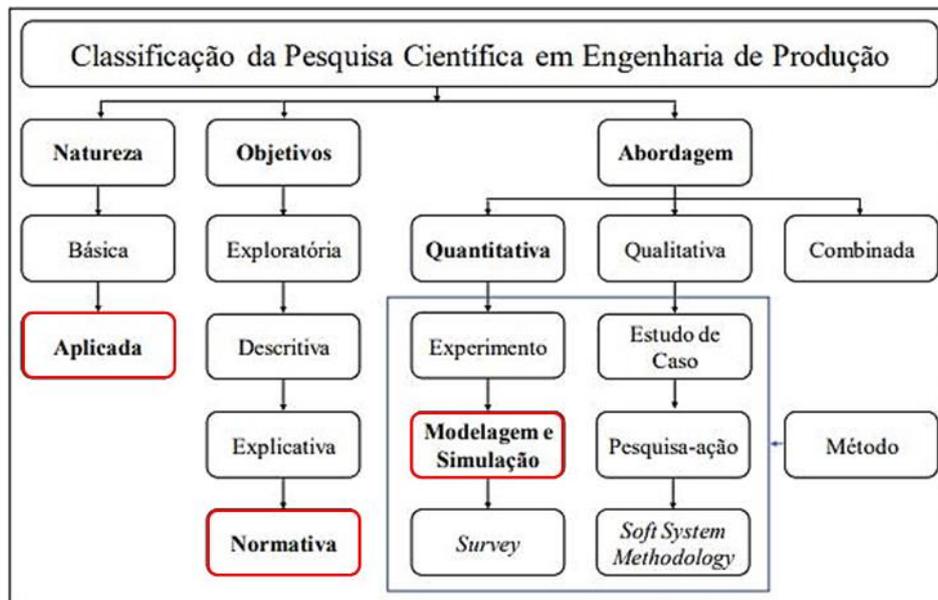


Figura 8: Classificação da pesquisa

Fonte: Adaptado de Miguel *et al.* (2018)

No que tange aos objetivos da pesquisa, estes podem ser classificados como exploratórios, descritivos, explicativos ou normativos. Seguindo a tipologia apresentada por Bertrand, Fransoo e Udenio (2023), a pesquisa em questão adota uma abordagem normativa. Tal abordagem busca encontrar soluções ótimas para novas definições ou visa o desenvolvimento de políticas e estratégias com vista à melhoria dos resultados já existentes na literatura. Portanto, o objetivo desta pesquisa é de natureza normativa, uma vez que visa propor, implementar e avaliar um *Framework* para MAOvS com o intuito de atingir resultados ótimos em modelos complexos de SED.

Adicionalmente, este trabalho adota uma abordagem quantitativa que parte da teoria para formular hipóteses, recolher dados por meio de observações, analisá-los e produzir resultados. Neste contexto, diversas metodologias podem ser empregadas. Neste estudo, a escolha recaiu sobre o método de modelagem e simulação, que se destaca por possibilitar a

manipulação de variáveis em um modelo representativo da realidade, sem interferir no ambiente real durante essa manipulação (Miguel et al., 2018).

Mitroff *et al.* (1974) propuseram uma metodologia científica de resolução de problemas voltadas para área de pesquisa operacional, e em especial para modelagem e simulação. Assim como ilustrado na Figura 9, os autores propuseram a subdivisão do projeto de simulação em quatro etapas, a saber: conceitualização, modelagem, solução do modelo, e implementação. Esta tese adotará o este método pois Mitroff *et al.* (1974) é um método clássico, robusto e amplamente discutido pela literatura de pesquisa operacional, e ainda considerado um método válido e aplicável nos tempos atuais (Bertrand; Fransoo; Udenio, 2023; Wollmann; Tortato, 2019).

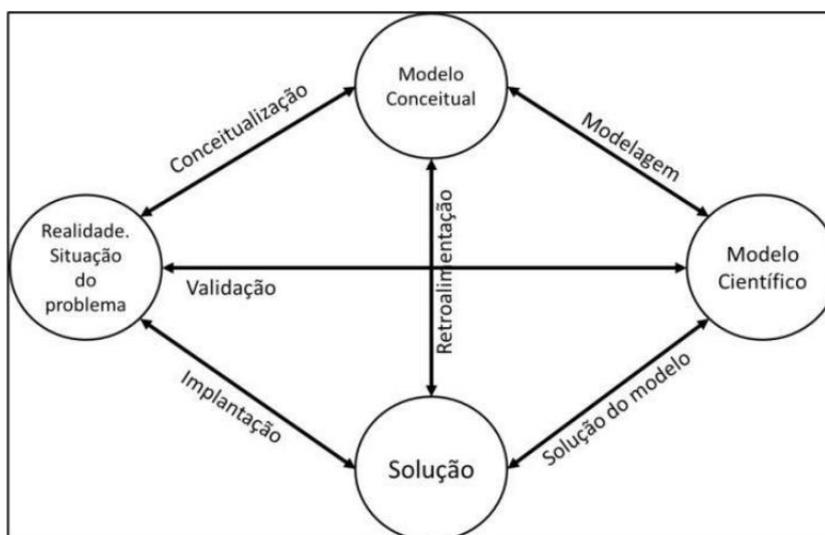


Figura 9: Sistema de resolução de problemas

Fonte: Adaptado de Mitroff *et al.* (1974)

Bertrand, Fransoo e Udenio (2023) propõem a classificação das pesquisas quantitativas com base nas etapas da metodologia de Mitroff *et al.* (1974), sendo divididas em quatro tipos, a depender do ciclo do projeto, a saber:

- Empírica descritiva: projeto em que o pesquisador envolve o ciclo conceitualização, modelagem e validação;
- Empírica prescritiva: engloba as ações de conceitualização, modelagem, solução do modelo e implantação;
- Axiomática descritiva: envolve a ação de modelagem, passando do modelo conceitual para o modelo científico;
- Axiomática prescritiva: envolve as ações de modelagem e solução do modelo.

Dada a classificação proposta por Bertrand, Fransoo e Udenio (2023), esta pesquisa pode ser entendida como axiomática prescritiva, ao passo que não pretende basear-se na

observação de um sistema real, mas a partir de um modelo conceitual pré-existente desenvolver o modelo científico e prover uma solução para o problema a partir otimização de modelos de SED com o uso de metamodelagem.

3.2. Etapas da Pesquisa a Serem Desenvolvidas

A Figura 10 apresenta as etapas a serem seguidas na elaboração desta pesquisa.

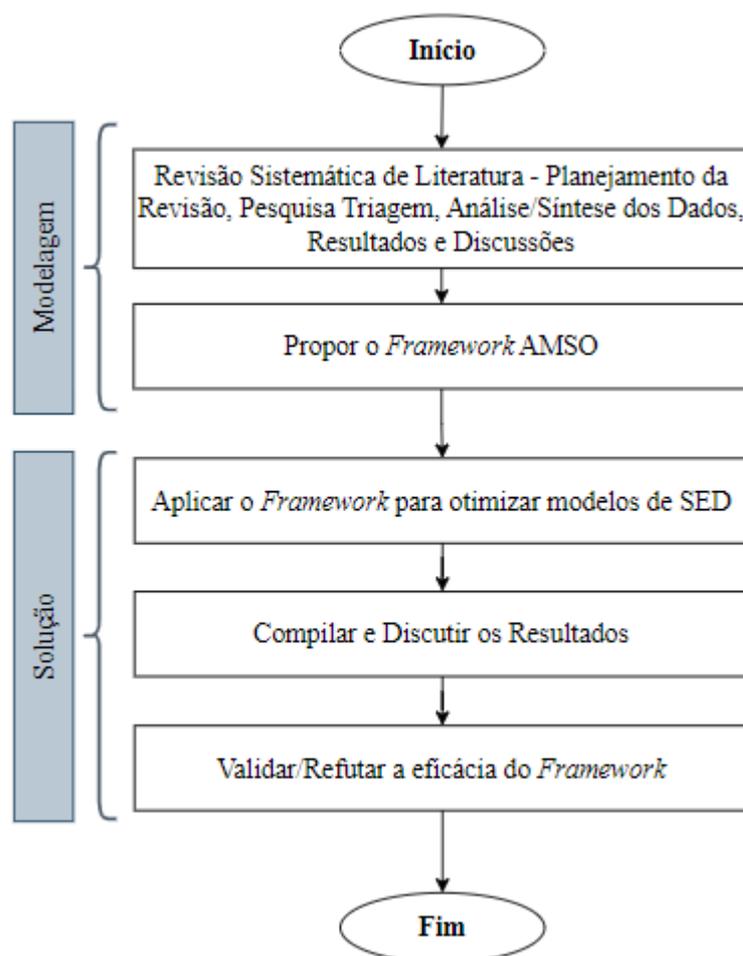


Figura 10: Etapas da pesquisa

A pesquisa adotará as seguintes etapas:

1. Modelagem (Capítulo 4 e Capítulo 5): Esta etapa iniciará com uma RSL para identificar o contexto de aplicação, técnicas e algoritmos relevantes em MAOVs. Com base nos resultados da RSL, será proposto um *framework* de metamodelagem adaptativa, visando aprimorar a qualidade das respostas (valor ótimo alcançado);
2. Solução do Modelo (Capítulo 6): O *framework* proposto será aplicado à otimização de modelos de SED, sendo validado por meio de comparação dos resultados com dois algoritmos de OvS amplamente utilizados: AG e EGO.

Dessa forma, o estudo seguirá um processo estruturado, da revisão da literatura à validação do *framework*, garantindo uma abordagem abrangente e bem fundamentada.

4. REVISÃO SISTEMÁTICA DE LITERATURA

Este Capítulo apresenta uma RSL sobre o tema proposto nesta tese, MAOVs, buscando o entendimento e análise da sua respectiva literatura e a fronteira do conhecimento no qual este trabalho está inserido. Este Capítulo faz parte da primeira etapa do método de pesquisa utilizado nesta tese, a contextualização do problema.

4.1. Trabalhos Relacionados

A fim de explorar trabalhos de revisão de literatura que abordaram temas semelhantes aos propostos nesta RSL, conduziu-se uma busca exploratória nas bases de dados científicas *Scopus*, *Web of Science*, *IEEE Xplore*, *ACM Library*, *Taylor & Francis* e *Science Direct*, abrangendo artigos de periódicos publicados até abril de 2023. Nesta etapa, buscou-se revisões relacionadas que abordem a lógica booleana (*AND/OR*) aos termos correlatos a metamodelagem adaptativa (*Adaptive sample*, *Adaptive sampling*, *Adaptive metamodel*, *Adaptive search*, *Adaptive optimization*, *Infill search*, *Infill sampling*, *Bayesian Optimization*, *Sequential optimization*, *Sequential search*, *Efficient Global Optimization*), Otimização via Simulação (*Simulation optimization*, *Simulation-based optimization*, *Optimization via simulation*), metamodelagem (*Metamodel*, *Meta-model*, *metamodeling*, *Surrogate*, *Emulator*, *Substitute model*, *Response surface*) e revisão de literatura (*Literature review*, *State of the art*, *Overview*, *Survey*).

Durante a busca exploratória, foram encontrados e avaliados nove artigos, conforme resumido na Quadro 1. Ressalta-se que não é objetivo desta seção apresentar todas as revisões existentes na área, mas fornecer uma compreensão dos estudos anteriores e *insights* sobre o tema, que são utilizados para a definição do protocolo de pesquisa.

Quadro 1: Resumo das revisões sobre Metamodelagem Adaptativa

Artigo	Método de Pesquisa	Foco	Metamodelo	Estratégia de Amostragem
Kleijnen (2009)	Revisão de Literatura	Premissas e Formulações do <i>Kriging</i>	<i>Kriging</i>	Fixa (LHD) e adaptativa (EI)
Kleijnen (2014)	Survey	OvS via <i>Kriging</i> e <i>Boostrapping</i>	<i>Kriging</i>	Fixa e adaptativa
Kleijnen (2017b)	Revisão de Literatura	Design e análise de experimentos em simulação	<i>Kriging</i> e RP	Fixa e adaptativa (Response Surface Methodology e EGO)
Vu <i>et al.</i> (2017)	Survey	Metamodelagem para otimização <i>black-box</i> contínua	Ensembles, SVM, <i>Kriging</i> , RBF e RP	Adaptativa

Liu, Ong e Cai (2018)	Revisão de Literatura	Metamodelos globais para projetos de engenharia	Não Limitado	Adaptativa
Rojas-Gonzalez e Van Nieuwenhuyse (2020)	Survey	OvS Multiobjetivo via <i>Kriging Infill Search</i> em metamodelagem por <i>Kriging</i> e RBF	Kriging	Adaptativa
Razali <i>et al.</i> (2020)	Revisão de Literatura	Aplicações de ML em meta-heurísticas	Kriging e RBF	Adaptativa
Talbi (2021)	Survey e Taxonomia	Metamodelagem em OvS	Cita o uso de <i>Kriging</i>	Adaptativa (EGO)
Amaral <i>et al.</i> (2022a)	RSL		Não limitado	Fixa e Adaptativa (Aborda, mas não é o foco do trabalho)

Os primeiros trabalhos encontrados foram desenvolvidos por Jack P. C. Kleijnen, com revisões focadas no uso de *Kriging* e Regressão Polinomial para metamodelagem. Em Kleijnen (2009), o autor revisa as premissas e formulações básicas envolvidas no *Kriging*, discutindo também conceitos de DoE e procedimentos de metamodelagem adaptativa. Além disto, foram discutidas questões de análise de sensibilidade, *designs* sequenciais e adaptativos, e *Bootstrapping*.

Em uma *survey*, Kleijnen (2014) apresentou os principais conceitos matemáticos e estatísticos envolvendo *Kriging* e *Bootstrapping* em OvS. Embora o foco do artigo não tenha sido o metamodelagem adaptativa, Kleijnen (2014) promoveu discussões e tratou de como o *Kriging* e *Bootstrapping* se relacionam no contexto de EGO em OvS determinística, estocástica e otimização robusta. Sobre o algoritmo EGO, o autor discutiu como a abordagem de *Bootstrapping* pode melhorar a estimação da variância do *Kriging*, promovendo uma melhoria em toda a otimização.

Em Kleijnen (2017b) é apresentado uma revisão sobre *design* e análise de experimentos em simulação. O trabalho abordou dois tipos de metamodelos (*Kriging* e Regressão Polinomial), DOEs e técnicas correlatas, como triagem de fatores, análise de sensibilidade, risco e incertezas. Neste contexto, o autor aborda *designs* sequenciais associados aos metamodelos, discutindo conceitos de *Response Surface Methodology* para Regressão Polinomial e EGO para *Kriging*.

No mesmo ano, Vu *et al.* (2017) abordaram técnicas de metamodelagem para solução de problemas *Black-box*. Nesta revisão, os autores apresentaram e discutiram os principais conceitos sobre técnicas DoE (*design* Geométrico e Estatístico), metamodelos (*Kriging*, Regressão Polinomial, SVM, RBF), FAs e seleção de regiões de busca. No ano seguinte, Lui, Ong e Cai (2018) revisaram, categorizaram e analisaram abordagens de metamodelagem

adaptativa mono e multiobjetivo para metamodelagem em projetos de engenharia baseados em simulação. O foco dos autores foram métodos para desenvolvimento de metamodelos com alta precisão e não aplicações de otimização, embora os autores discutam brevemente este tópico. Os autores promoveram uma extensa discussão dos principais tópicos que afetam o sucesso de abordagens adaptativas.

Com foco em problemas multiobjetivos, Rojas-Gonzalez e Van Nieuwenhuyse (2020) apresentaram um *survey* sobre métodos de metamodelagem adaptativas baseados em *Kriging*. Os autores discutiram os principais trabalhos na área, apontando as peculiaridades de cada algoritmo proposto em termos de formulação matemática e aplicabilidade. Além disto, foram apresentadas duas estruturas gerais para métodos de otimização multiobjetivo baseado em *Kriging* com critério *Infill* (Funções de Aquisição) mono e multiobjetivo, respectivamente. No mesmo ano, Razali *et al.* (2020) discutiram os desafios e oportunidades envolvendo amostragem adaptativa em metamodelagem por *Kriging* e RBF. Adicionalmente, os autores propuseram um método baseado em *Ensembles* não homogêneos para obtenção de novos pontos a serem experimentados (Função de Aquisição).

O trabalho de Talbi (2021) investigou diversas oportunidades do uso de ML na melhoria de desempenho de meta-heurísticas. Nesta pesquisa, o autor promoveu um detalhado estudo taxonômico, distinguindo três caminhos, nos quais o ML pode auxiliar meta-heurísticas: *High-level*, no qual o ML apoia a seleção e geração das meta-heurísticas; *Low-level*, no qual o ML atua na seleção dos hiperparâmetros estruturais da meta-heurísticas; e *Problem-level*, o qual auxilia na modelagem e processo de solução do problema. Neste último nível, *problem-level*, o autor discute a metamodelagem com o algoritmo EGO.

O trabalho mais recente, Amaral *et al.* (2022a), desenvolveu uma RSL sobre metamodelagem em problemas de OvS, analisando aspectos como o tipo e estrutura dos metamodelos utilizados em problemas de OvS, características dos problemas estudados, métodos de simulação, técnicas de DoE, e método de otimização empregados. Embora abordou-se trabalhos de metamodelagem adaptativa, as peculiaridades deste método não era o foco do artigo, restando diversas questões a serem respondidas sobre as especificidades envolvidas nesta temática.

Ressalta-se que os artigos mencionados anteriormente foram a base para o desenvolvimento desta RSL. No entanto, eles possuíram foco em dimensões específicas de MA, não sendo encontrados trabalhos que tratem a MA de maneira abrangente. Desta forma, diferentemente deles, este estudo não se limita ao tipo de metamodelos, simulação ou características do problema, conforme apresentado na Quadro 1. Portanto, a contribuição desta

revisão é a análise abrangente sobre a literatura existente de MA, a partir da perspectiva e rigor científico da RSL e da metodologia CIMO (Xiao; Watson, 2019).

Vale ressaltar que esta RSL, embora não tenha a pretensão de abranger toda a literatura sobre MA, analisa trabalhos indexados nas principais bases de dados científicas, fornecendo uma visão holística sobre o assunto, consolidando e categorizando a literatura existente. Nesse caso, pretende-se oferecer uma melhor delimitação da fronteira do conhecimento na área, identificando e esclarecendo as oportunidades e desafios relacionados ao tema.

4.2. O Método de Revisão Sistemática de Literatura

A RSL é uma metodologia que envolve a coleta, compreensão, síntese e avaliação de trabalhos científicos para criar a base teórico-científica necessária para compreender um determinado tópico (Levy; Ellis, 2006; Xiao; Watson, 2019). De acordo com Denyer, Tranfield e Van Aken (2008), a RSL auxilia os pesquisadores em todas as etapas metodológicas, incluindo definição do problema, seleção do método, coleta e análise de dados, ajudando a minimizar falhas nas conclusões do trabalho e fornecendo maior clareza sobre os procedimentos de pesquisa utilizados. Ao contrário das revisões de literatura tradicionais, a RSL é um projeto de pesquisa que segue um procedimento predefinido para responder a perguntas específicas utilizando a literatura existente.

Oliveira, Lima e Montevechi (2016) enfatizam que as revisões de literatura exploram diversos campos de estudo, mas nem sempre seguem uma abordagem sistemática. Por outro lado, a RSL é sistemática, envolvendo uma crítica durante todas as etapas da pesquisa para evitar viés e distorção de dados. Booth, Sutton e Papaioannou (2016) afirmam que a RSL deve ser explícita, transparente, metodológica, objetiva, padronizada, estruturada e reproduzível para manter sua natureza sistemática. Segundo Kitchenham *et al.* (2009), as RSLs diferem das revisões de literatura convencionais nas seguintes características:

- As RSLs iniciam com a definição de um protocolo de revisão, que estabelece as perguntas de pesquisa e a metodologia a ser utilizada;
- As RSLs possuem uma estratégia de busca bem definida e documentam a abordagem de pesquisa para garantir transparência e precisão;
- As RSLs estabelecem critérios explícitos de inclusão/exclusão dos artigos;
- As RSLs especificam as informações a serem obtidas de cada estudo, o que é necessário para realizar análises quantitativas.

A principal motivação deste estudo decorre da combinação do aumento significativo de publicações sobre MA em OvS, sua aplicação em diversas áreas, e a escassez de trabalhos que reúnam e discutam de forma ampla e sistemática o conhecimento sobre esse tema. Espera-se que os resultados desta pesquisa possam ser úteis em todas as fases de projetos nessa área, além de apresentar o estado atual do conhecimento em MAOvS, compilando o que já foi estudado e, especialmente, identificando oportunidades para novas pesquisas. Portanto, os objetivos desta RSL são os seguintes:

- a) Identificar e examinar as características dos metamodelos (tipos, parâmetros, medidas de desempenho), tipos de simulação, técnicas de DoE, funções de aquisição, métodos de otimização e aplicações relacionadas a estudos de MAOvS;
- b) Analisar e resumir os resultados, respondendo às QPs listadas na Seção 4.3.
- c) Discutir os resultados, identificando a fronteira do conhecimento, lacunas e tendências a serem exploradas em MAOvS.

A metodologia adotada neste trabalho é baseada nos estudos de Amaral *et al.* (2022a), Santos *et al.* (2022) Sousa Junior *et al.* (2019), e está dividida em quatro etapas: planejamento de questões de pesquisa; busca/triagem; análise/síntese; e resultados e discussões. O protocolo de pesquisa deste trabalho, apresentado nas seções seguintes, foi desenvolvido por meio de um painel de especialistas. O painel contou com 4 doutores com amplo conhecimento em simulação, otimização e metamodelagem, além de estudantes de doutorado que atuam em temas associados. O objetivo do painel foi o planejamento e definição das questões de pesquisa e procedimentos de busca/triagem a serem empregados.

4.3. Planejamento das Questões de Pesquisa

Para alcançar os objetivos mencionados anteriormente, a formulação de perguntas de pesquisa é um passo crucial para extrair e analisar o conhecimento inerente ao tópico em questão. Este trabalho utiliza o *framework* lógico CIMO proposto por Denyer, Tranfield e Van Aken (2008) e aplicado por diversos autores (Aouadni; Aouadni; Rebaï, 2019; Bin Makhshen *et al.*, 2020; Kochan; Nowicki, 2018). Esse *framework* envolve a divisão das perguntas de pesquisa em quatro elementos, utilizando o método CIMO, que são: contexto, intervenção, mecanismo e resultados. O contexto envolve os problemas e áreas em que MA é aplicada. A intervenção aborda as técnicas e ferramentas utilizadas nos desenvolvimentos das soluções. O mecanismo descreve como essas técnicas são aplicadas para melhorar o desempenho do

sistema. E, por fim, os resultados englobam as vantagens e desafios associados a essa abordagem.

Dessa forma, esta pesquisa integra o método de RSL com a metodologia CIMO para examinar a literatura de metamodelagem adaptativa. As QPs desta revisão são:

- QP1. Quais os principais objetivos e campos de aplicação envolvidos nos estudos de MAOVs? (Contexto);
- QP2. Quais são os tipos de simulação que estão relacionados ao uso da metamodelagem? (Mecanismo);
- QP3. Quais metamodelos são mais frequentemente utilizados em MAOVs? (Intervenção);
- QP4. Quais designs experimentais são utilizados? (Intervenção);
- QP5. Quais funções de aquisição e métodos de otimização adotados são utilizados para definir o próximo ponto a ser avaliado? (Intervenção);
- QP6. Como estão estruturados os métodos/*frameworks* seguidos pelos pesquisadores? (Resultados).

4.4. Busca e Triagem dos Artigos

Para esta RSL, o procedimento de busca e triagem dos artigos seguiu três etapas: Busca Exploratória, Triagem, e Avaliação de Qualidade (Amaral et al., 2022a).

Na etapa de Busca Exploratória pesquisou-se artigos científicos das bases Scopus, Web of Science, IEEE Xplore, ACM Digital Library, Taylor & Francis e Science Direct. Com base no painel de especialistas e nas revisões apresentadas na Seção 4.1, foram definidas as palavras-chave de pesquisa como: (*Adaptive sample OR Adaptive sampling OR Adaptive metamodel OR Adaptive search OR Adaptive optimization OR Infill search OR Infill sampling OR Bayesian optimization OR Sequential optimization OR Sequential search OR Efficient Global Optimization*) AND (*Simulation optimization OR Simulation-based optimization OR Optimization via simulation*) AND (*Metamodel OR Meta-model OR Metamodeling OR Surrogate OR Emulator OR Substitute model OR Response Surface*). A meta-busca por artigos publicados até abril de 2023 retornou 43 artigos da Scopus, 51 da *Web of Science*, 7 da *IEEE Xplore*, 110 da *ACM Digital Library*, 39 da *Taylor & Francis* e 17 da *Science Direct* (total de 267 artigos).

Na etapa de Triagem, a RSL utilizou o seguinte conjunto de critérios em cada busca para determinar se um artigo era elegível: (i) apenas artigos completos publicados em revistas científicas revisadas por pares; (ii) apenas artigos escritos em inglês; (iii) disponíveis para

download; (iv) os termos de pesquisa deviam estar presentes no título, resumo ou palavras-chave do artigo.

Durante a Triagem, 131 artigos foram excluídos da busca, pois não atendiam pelo menos um dos critérios previamente definidos: 122 não foram publicados em revistas científicas revisadas por pares, 6 não estavam disponíveis para download e 3 não possuíam os termos de busca no título, resumo ou palavras-chave.

Após esta etapa restaram 29 artigos da *Scopus*, 39 da *Web of Science*, 1 da *IEEE Xplore*, 19 da *ACM Digital Library*, 31 da *Taylor & Francis* e 17 da *Science Direct*. Por fim, 34 artigos duplicados foram removidos da base, totalizando 103 artigos a serem considerados na fase de Avaliação da Qualidade.

Posteriormente, os artigos selecionados passaram por uma etapa de Avaliação de Qualidade, na qual o título, resumo, introdução, método e conclusão foram cuidadosamente examinados para identificar aqueles alinhados com os objetivos da RSL. Quaisquer artigos que não atendessem aos critérios de pesquisa estabelecidos anteriormente ou que não abordassem a aplicação de MAOvS foram excluídos. Durante esse processo, cada documento foi examinado por dois pesquisadores, e caso houvesse discordância sobre a inclusão/exclusão de um artigo, um terceiro pesquisador deveria avaliá-lo.

Durante esta etapa, 35 artigos foram excluídos da análise pelos seguintes motivos: 12 (34,3%) utilizam metamodelagem em OvS, mas não uma abordagem adaptativa; 10 (28,6%) não abordam o tema estudado, 7 (20,0%) usaram metamodelagem adaptativa para outros propósitos que não OvS, 3 (8,6%) tratavam de OvS mas não usavam metamodelagem e 3 (8,6%) eram artigos de revisões. Após essa etapa, 68 artigos foram considerados adequados para leitura completa e extração dos dados. A Figura 11 resume os procedimentos de pesquisa/triagem.

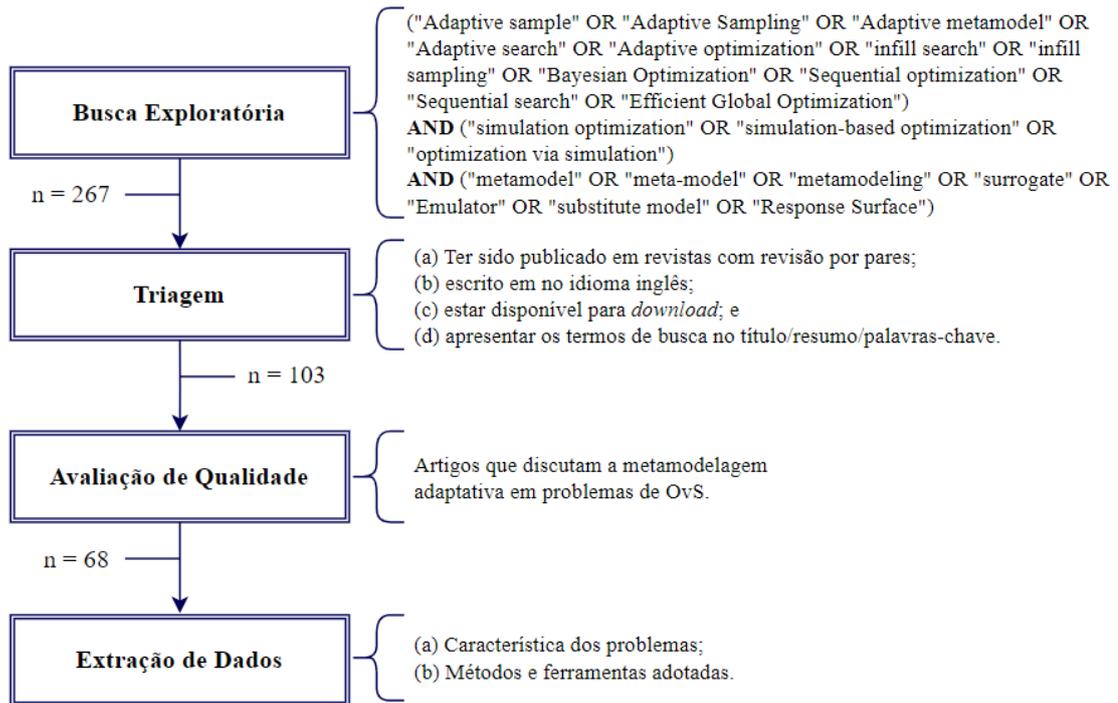


Figura 11: Procedimento de busca/triagem dos artigos.

4.5. Análise e Síntese

Para abordar as questões de pesquisa, foi realizado um processo de avaliação dos 68 artigos com base em oito critérios distintos, conforme ilustrado na Figura 12. Um dos pesquisadores ficou responsável por extrair os dados relevantes de cada artigo, enquanto outro verificou a precisão dos dados extraídos para evitar qualquer influência pessoal. Caso houvesse alguma divergência, um painel composto por todos os pesquisadores foi convocado para discussão e resolução de possíveis problemas. Após a etapa de extração de dados, as informações coletadas foram submetidas a análises estatísticas descritivas, conforme detalhado na Seção 4.6. Os dados foram organizados em uma planilha do MS Excel[®] e posteriormente analisados utilizando a linguagem *Python*.

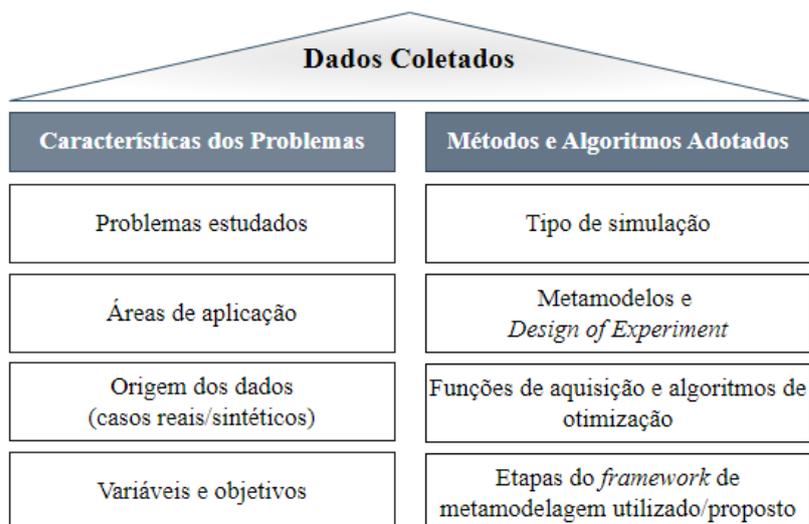


Figura 12: Dados coletados dos artigos.

Com base na coleta e análise dos dados elencados na Figura 12 será possível responder as questões de pesquisa listadas na Seção 4.3. A QP1 será respondida com base nos dados referentes às Características dos Problemas (problemas estudados, áreas de aplicação, origem dos dados, variáveis e objetivos). O macro tópico Métodos e Algoritmos Adotados será a base para resposta das QPs 2 a 6. A QP2 se relaciona à análise do item Método de simulação, os dados referentes a Metamodelos e *design* experimental servirão de base para resposta à QP3 e QP4, o tópico Funções de aquisição e métodos de otimização está relacionado a QP5 e o tópico Etapas do *framework* de MA utilizado/proposto refere-se a QP6.

Neste último tópico serão elencadas etapas que estiveram presentes em pelo menos dois trabalhos. Por fim, estas etapas serão classificadas em etapas fundamentais para *frameworks* de MAOvS (presentes em pelo menos 90% dos trabalhos), etapas recomendadas, mas não fundamentais (presentes em pelo menos 10% dos trabalhos), e etapas opcionais (menos de 10% dos trabalhos).

4.6. Resultados e Discussões da RSL

4.6.1. Características dos Problemas

O objetivo desta seção é discutir o pilar “Características dos Problemas”, que está diretamente relacionado à QP1 (Quais os principais objetivos e campos de aplicação envolvidos nos estudos de MAOvS?). A

Tabela 1 apresenta os principais aspectos relacionados aos problemas, bem como suas especificidades, como os objetivos e tipos de variáveis envolvidas.

Tabela 1: Características dos problemas

Tipo de problema	Mono-objetivo	Multiobjetivo	Total	Acum.
Funções sintéticas	26 (28,0%)	3 (3,2%)	29 (31,2%)	31,2%
<i>Design</i> de produto	14 (15,1%)	6 (6,5%)	20 (21,5%)	52,7%
Parâmetros de processo	11 (11,8%)	2 (2,2%)	13 (14,0%)	66,7%
Alocação de recursos	5 (5,4%)	2 (2,2%)	7 (7,5%)	74,2%
Controle de tráfego urbano	4 (4,3%)	3 (3,2%)	7 (7,5%)	81,7%
Controle de estoque	5 (5,4%)	1 (1,1%)	6 (6,5%)	88,2%
Outros	9 (9,7%)	2 (2,2%)	11 (11,8%)	100,0%
Setor de Aplicação				
Gestão hídrica	8 (11,8%)	4 (5,9%)	12 (17,6%)	17,6%
Transporte/logística	5 (7,4%)	3 (4,4%)	8 (11,8%)	29,4%
Ind. química/farmacêutica	6 (8,8%)	0 (0,0%)	6 (8,8%)	38,2%
<i>Healthcare</i>	2 (2,9%)	2 (2,9%)	4 (5,9%)	44,1%
Automotivo	3 (4,4%)	0 (0,0%)	3 (4,4%)	48,5%
Eletrônicos	3 (4,4%)	0 (0,0%)	3 (4,4%)	52,9%
Metalúrgica	2 (2,9%)	1 (1,5%)	3 (4,4%)	57,4%
Outros	9 (13,2%)	2 (2,9%)	11 (16,2%)	73,5%
Não especificado	3 (4,4%)	1 (1,5%)	4 (5,9%)	79,4%
Não aplicável	13 (19,1%)	1 (1,5%)	14 (20,6%)	100,0%
Origem dos dados				
Casos reais	32 (37,6%)	12 (14,1%)	44 (51,8%)	51,76%
Dados sintéticos	38 (44,7%)	3 (3,5%)	41 (48,2%)	100,0%
Tipo de variáveis				
Contínuos	36 (52,9%)	11 (16,2%)	47 (69,1%)	69,1%
Discretos	7 (10,3%)	1 (1,5%)	8 (11,8%)	80,9%
Mistos	9 (13,2%)	1 (1,5%)	10 (14,7%)	95,6%
Não especificado	2 (2,9%)	1 (1,5%)	3 (4,4%)	100,0%

De acordo com a

Tabela 1, a maioria dos estudos (31,2%) aplicam metamodelagem adaptativa em funções sintéticas (isto é, um conjunto de equações matemáticas que não representam um contexto prático) com o intuito de avaliar e comparar a eficácia dos métodos propostos em funções *benchmarking* da literatura. Destes, 28,0% abordam métodos para problemas de otimização mono-objetivos (Keys; Rees; Greenwood, 2002; Kim; Boukouvala, 2020; Mehdad; Kleijnen, 2015; Pedrielli; Wang; Ng, 2020) e 3,2% em problemas de otimização multiobjetivos (Chang *et al.*, 2022; Han; Ouyang, 2022; Zheng; Li, 2023).

Em 21,5% dos trabalhos, os autores abordaram métodos de metamodelagem adaptativa para otimização do *design* de produtos. Como apresentado na

Quadro 2, destacam-se aplicações para design de produtos eletrônicos, como exemplo cita-se o trabalho de Gorissen, Dhaene e De Turck (2009), que propõem um método de seleção automática de metamodelos em problemas de otimização complexos, tais como *design* de circuitos eletromagnéticos e geometria de um veículo lançador aeroespacial. Wang *et al.* (2020) apresentaram um método para construção de metamodelos multi-fidedígnos para otimização de circuitos eletromagnéticos. Outros exemplos envolvem otimização do *design* de peças automotivas (Ahrari *et al.*, 2021), estrutura de veículos para maximizar a absorção de energia em colisões (Gu; Li; Dong, 2012), laminados compósitos (Milani *et al.*, 2009), geometria de uma estrutura de isolamento vibracional (Qian *et al.*, 2021) e otimização do projeto de Stent para aplicações em cirurgias cardiovasculares (Li *et al.*, 2017).

Quadro 2: Artigos por área versus tipo de problema

Área/Problema	Funções sintéticas	Design de produto	Parâmetros de processo	Alocação de recursos	Controle de tráfego	Controle de estoque
Gestão hídrica			(Hasanzadeh; Saadatpour; Afshar, 2020; Luo <i>et al.</i> , 2020; Regis, 2016; Saadatpour <i>et al.</i> , 2021; Zhang <i>et al.</i> , 2022a, 2022b)	(Candelieri <i>et al.</i> , 2022)		
Transporte/logística	(Tay; Osorio, 2022; Zheng; Li, 2023)				(Feygin <i>et al.</i> , 2020; Tay; Osorio, 2022; Teixeira <i>et al.</i> , 2022; Yin; Liu; Zheng, 2022; Zheng <i>et al.</i> , 2019; Zheng; LI, 2023; Zhong <i>et al.</i> , 2021)	(Quan <i>et al.</i> , 2015)
Química/ farmacêutica	(Desautels; Krause; Burdick, 2014; Wang; Ierapetritou, 2017, 2018; Xie; Yi; Zheng, 2021)	(Desautels; Krause; Burdick, 2014)	(Boukouvala; Ierapetritou, 2013; Wang; Ierapetritou, 2017)	(Xiong <i>et al.</i> , 2021)		(Xie; Yi; Zheng, 2021)
Healthcare	(Briffoteaux <i>et al.</i> , 2020)	(Li <i>et al.</i> , 2017)			(Coelho; Pinto, 2018; Kasaie; Kelton, 2013)	
Automotivo	(Ahrari <i>et al.</i> , 2021; Gu; Li; Dong, 2012; Ivanova; Kuhnt, 2014)	(Ahrari <i>et al.</i> , 2021; Gu; Li; Dong, 2012)	(Ivanova; Kuhnt, 2014)			
Eletrônicos	(Gorissen; Dhaene; De Turck, 2009; Ur Rehman; Langelaar, 2016)	(Gorissen; Dhaene; De Turck, 2009; Ur Rehman; Langelaar, 2016; Wang <i>et al.</i> , 2020)				
Metalúrgica		(Milani <i>et al.</i> , 2009; Qian <i>et al.</i> , 2021)2(Bamdad; Cholette; Bell, 2020; Bharaj <i>et al.</i> , 2015; Chang <i>et al.</i> , 2022; Gengembre <i>et al.</i> , 2012; Zhou; Turng, 2007)Gengembre <i>et al.</i> , 2012; Zhou; Turng, 2007)	(Del Castillo; Santiago, 2011; Zhang <i>et al.</i> , 2009; Zhou; Turng, 2007)		(Giahi; Mackenzie; Hu, 2022; Rosen; Harmonosky, 2005)	

A terceira classe de problemas mais abordados refere-se à otimização de parâmetros de processos (14,0%). Este tipo de problema é foco de diversos estudos relacionados à gestão hídrica, em especial para controle/determinação de taxa de bombeamento para extração e

remediação de águas subterrâneas (Luo *et al.*, 2020; Zhang *et al.*, 2022a, 2022b). Embora este tipo de problema tenha envolvido predominantemente apenas um objetivo, é possível destacar o uso da metamodelagem adaptativa em problemas de otimização multiobjetivo de parâmetros de processos, como no caso de Hasanzadeh, Saadatpour e Afshar (2020), que associaram lógica *fuzzy* e RNAs para lidar com os objetivos conflitantes em um problema alocação de carga de resíduos multi-poluentes no corpo de água de Behesht-Abad, Irã.

Outro exemplo pode ser encontrado em Saadatpour *et al.* (2021), que buscaram melhorar a eficiência e a sustentabilidade geral da geração de energia hidrelétrica, ao mesmo tempo em que garante a preservação dos recursos hídricos e minimiza os impactos ambientais. A proposta dos autores utilizou RNAs associados a *Particle Swarm Optimization* e *Principal Components Analysis*. Exemplos deste tipo de problema também podem ser encontrados em aplicações visando a otimização de processos contínuos de fabricação farmacêutica (Boukouvala; Ierapetritou, 2013; Wang; Ierapetritou, 2017), e de processos de conformação de chapas metálicas no setor automotivo (Ivanova; Kuhnt, 2014).

O uso da metamodelagem adaptativa para alocação de recursos representou 7,5% dos trabalhos analisados. (Kasaie; Kelton, 2013) discutiram o uso de Metodologia de Superfície de Resposta e SBA para alocação eficiente de recursos de controle de epidemias, buscando melhorar o planejamento e a resposta diante de surtos de doenças contagiosas. Também com aplicações no setor de *healthcare*, Coelho e Pinto (2018) abordaram o uso de *Kriging* e EI para metamodelagem adaptativa de modelos de SED para otimizar a alocação de ambulâncias em um sistema de emergência médica, visando minimizar o tempo de resposta dos serviços.

Em gestão hídrica, Candelieri *et al.* (2022) apresentaram dois algoritmos para otimização multiobjetivo em problemas de programação de bombeamento e alocação de sensores para detecção precoce de intrusão de contaminantes em aquíferos. Outros exemplos envolvem otimização de processos de destilação (Xiong *et al.*, 2021), sistema de energias renováveis (Giahi; Mackenzie; Hu, 2022) e produção de semicondutores (Rosen; Harmonosky, 2005)

Outros problemas abordados pelos trabalhos foram a gestão de tráfego, por exemplo, controle de tempo de semáforos (Tay; Osorio, 2022; Zheng *et al.*, 2019; Zheng; Li, 2023), design de redes e adaptação de vias (Teixeira *et al.*, 2022; Yin; Liu; Zheng, 2022) e definição de estratégia de precificação de pedágios e transporte público (Zhong *et al.*, 2021). Além destes, é possível citar aplicações de dimensionamento e controle de estoques (Quan *et al.*, 2015; Xie; Yi; Zheng, 2021), programação/agendamento (Candelieri; Perego; Archetti, 2018; Zhao;

Minsker, 2021) e calibração de modelos (Briffoteaux *et al.*, 2020; Kugalur Palanisamy *et al.*, 2022; Liu *et al.*, 2022).

O segundo tópico apresentado na

Tabela 1 e destacado na Figura 13 (a) é a área de aplicação no qual os artigos aplicaram seus respectivos métodos de MAOVs propostos. Como já destacado anteriormente, aplicações em gestão hídrica foram as mais frequentes, abrangendo cerca de 17,6% dos trabalhos analisados, sendo que 11,8% abordaram problemas de otimização mono-objetivos e 5,9% apresentaram métodos para solução de problemas multiobjetivos. O segundo setor mais estudado foi o de transporte/logística (11,8%), com problemas envolvendo controle de semáforos, planejamento de tráfego e vias urbanas, precificação e gestão de estoque.

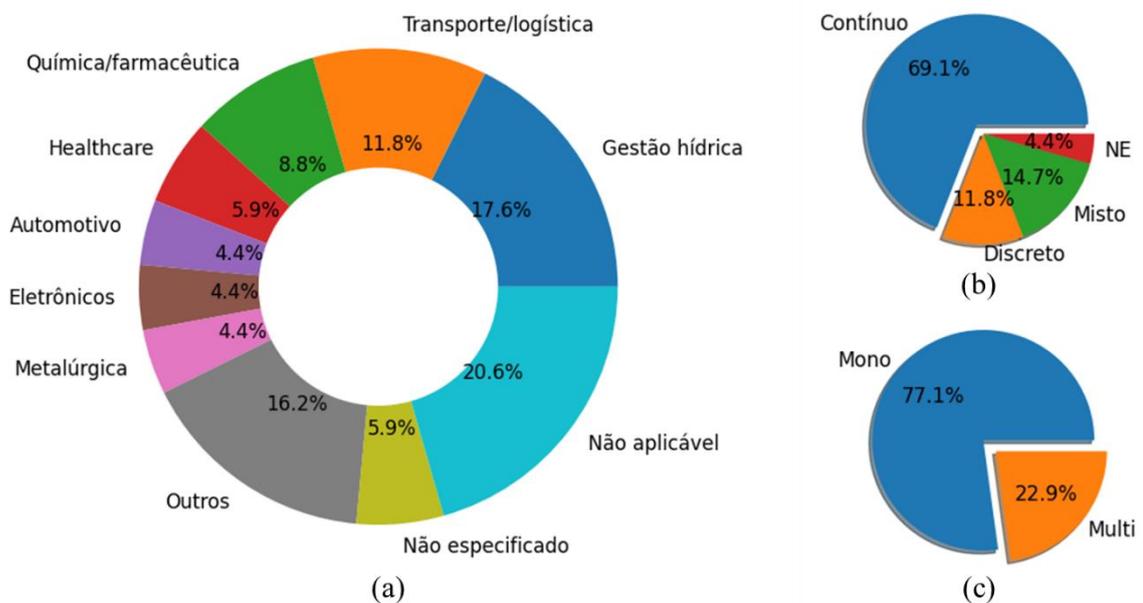


Figura 13: (a) Áreas de aplicação, (b) tipos de variáveis e (c) objetivos

Aplicações no setor químico/farmacêutico também estiveram entre os mais observados, representado 8,8% dos trabalhos, os quais predominantemente trataram de problemas mono-objetivos. Além destes, aplicações no setor de *healthcare*, automotivo, eletrônicos e metalurgia também obtiveram destaque entre os artigos. Por fim, 20,6% dos trabalhos discutiram problemas de cunho geral, sem aplicação em um setor específico ou puramente teórico, como funções matemáticas sintéticas, e 5,9% abordaram casos reais, mas não especificaram o setor de aplicação.

O terceiro tópico abordado na

Tabela 1 diz respeito a origem dos dados. Em 51,2% dos artigos foram apresentados casos reais da indústria ou serviços, como é o caso apresentado em Bamdad, Cholette e Bell (2020), que buscaram otimizar o consumo de energia em construções utilizando uma estratégia

adaptativa com base em um *Committing (Ensemble)* de RNAs, e Zhou e Turng (2007), que utilizaram metamodelagem adaptativa por *Kriging* para determinar as condições ideais do processo de moldagem por injeção de lentes ópticas sem intervenção do usuário.

Em contrapartida, 48,8% dos trabalhos apresentaram estudos de caso teóricos, os quais os dados e informações para construção do modelo proviam da literatura ou representados por funções sintéticas *benchmarking*. Ressalta-se que este tipo de estudo de caso teórico, embora não apresente um contexto prático claro, é particularmente útil pois estes casos possuem condições de contorno e forma bem definidas e ótimos globais muitas vezes conhecidos, facilitando a avaliação e comparação dos métodos propostos em um ambiente controlado. Neste sentido, 25,0% apresentaram cases teóricos e práticos, com o intuito de avaliar suas propostas em funções sintéticas *benchmarking* e posteriormente discutir sua aplicação em cases práticos (Ahrari *et al.*, 2021; Ur Rehman; Langelaar, 2016; Yaohui, 2017).

O quarto tópico da

Tabela 1, e destacado na Figura 13 (b), aponta o tipo de variável de decisão envolvida nos problemas abordados pelos autores. Estudos envolvendo apenas variáveis de decisão contínuas foram predominantes, representando 96,1% dos trabalhos. Nestes, a dimensão dos problemas variou entre 1 e 162 variáveis de decisão, com média de 7,8 e mediana de 3 variáveis, sendo mais comum problemas bidimensionais (moda). Estudos envolvendo apenas variáveis de decisão discretas representaram 11,8%, apresentando problemas com dimensão de até 20 variáveis, sendo uma média de 5,1, mediana de 5 e moda de 6 variáveis de decisão. Por fim, os problemas envolvendo otimização inteira mista estiveram presentes em 14,7% dos artigos. Neste caso, os problemas apresentavam uma combinação de variáveis contínuas, discretas, categóricas e/ou binárias e dimensão média de 11,4 e mediana de 7 variáveis.

Por fim, como apresentado na Figura 13 (c), grande parte dos estudos abordaram problemas mono-objetivos (77,1%), enquanto 23,9% tratam de otimização multiobjetivo. Neste último, vale apenas destacar que estudos multiobjetivos apresentam desafios para a metamodelagem, como a necessidade de treinar um metamodelo para cada objetivo, ou utilizar técnicas que permitam a previsão de múltiplas respostas simultaneamente, como é o caso de RNAs. Outro desafio, e oportunidade, refere-se ao estabelecimento de uma função de aquisição eficiente e capaz de selecionar novos pontos a serem simulados considerando múltiplos objetivos, bem como o método de aglutinação empregado. Para mais informações sobre estes pontos sugere-se (Han; Ouyang, 2022).

Por fim, salienta-se que 10,3% dos trabalhos consideraram incerteza na tomada de decisões, trabalhando no conceito conhecido como otimização robusta. Em 20,6% dos trabalhos

os autores abordaram problemas com restrições, sejam elas diretamente ligadas às variáveis de decisão ou relacionadas a outras variáveis de resposta advindas do modelo. Para tratar deste último tipo de restrição é necessário, assim como na otimização multiobjetivo, treinar metamodelos adicionais para cada restrição, utilizar metamodelos aptos a gerarem múltiplas respostas, ou criar uma estratégia para agregar a violação das restrições como penalização na função objetivo.

Com base nestas informações, em resposta a primeira questão de pesquisa, observa-se que, além de demonstrações dos métodos propostos em funções matemáticas sintéticas, muitos estudos focaram em aplicações envolvendo *design* de produto e otimização de parâmetros de processos, principalmente no contexto de gestão hídrica. Estes problemas muitas vezes envolvem modelos complexos e caros de se executar, o que desperta a necessidade de elevá-los a outro nível de abstração: a metamodelagem. Além disto, destaca-se que grande parte dos trabalhos focaram em problemas de otimização mono-objetivos com variáveis de decisão contínuas. Vale a pena destacar que aplicações de otimização envolvendo variáveis discretas ou mistas e múltiplos abjetivos foram encontrados apenas em dois trabalhos (Candelieri *et al.*, 2022; Kasaie; Kelton, 2013), o que pode significar uma lacuna ainda pouco explorada.

4.6.2. Métodos Adotados

O objetivo desta Seção é apresentar as abordagens mais comuns na literatura sobre MAOvS, como método de simulação, metamodelos, projetos experimentais, funções de aquisição e métodos de busca (otimização), técnicas de validação, *softwares* e as linguagens de programação mais comuns. Os resultados são resumidos nas Tabela 2.

Tabela 2: Métodos adotados

Tipos de simulação	Mono-objetivo	Multiobjetivo	Total	Acum.
Simulação numérica	33 (47,1%)	12 (17,1%)	45 (64,3%)	64,3%
SED	6 (8,6%)	1 (1,4%)	7 (10%)	74,3%
SBA	1 (1,4%)	1 (1,4%)	2 (2,9%)	77,1%
Monte Carlo	1 (1,4%)	0 (0,0%)	1 (1,4%)	78,6%
Não especificado	14 (20%)	1 (1,4%)	15 (21,4%)	100,0%
Metamodelos				
<i>Kriging</i>	45 (54,2%)	7 (8,4%)	52 (62,7%)	62,7%
RNA	5 (6,0%)	2 (2,4%)	7 (8,4%)	71,1%
Regressão Polinomial	4 (4,8%)	3 (3,6%)	7 (8,4%)	79,5%
RBF	3 (3,6%)	1 (1,2%)	4 (4,8%)	84,3%
SVM	2 (2,4%)	1 (1,2%)	3 (3,6%)	87,9%
<i>Splines</i>	2 (2,4%)	0 (0,0%)	2 (2,4%)	90,4%
Outros	7 (8,4%)	1 (1,2%)	8 (9,6%)	100,0%
Designs Experimentais				
LHD	37 (52,1%)	5 (7%)	42 (59,2%)	59,2%
Design Aleatório	8 (11,3%)	3 (4,2%)	11 (15,5%)	74,6%
Fatorial Completo	2 (2,8%)	1 (1,4%)	3 (4,2%)	78,9%
<i>Grid Search</i>	1 (1,4%)	1 (1,4%)	2 (2,8%)	81,7%
População de AE	1 (1,4%)	1 (1,4%)	2 (2,8%)	84,5%
Outros	2 (2,8%)	3 (4,2%)	5 (7,0%)	91,5%
Não especificado	5 (7,0%)	1 (1,4%)	6 (8,5%)	100,0%
Funções de aquisição				
<i>Expected Improvement</i>	25 (30,5%)	5 (6,1%)	30 (36,6%)	36,6%
<i>Predicted Best</i>	13 (15,9%)	4 (4,9%)	17 (20,7%)	57,3%
<i>Probability of Improvement</i>	2 (2,4%)	2 (2,4%)	4 (4,9%)	62,2%
<i>Knowledge-gradient</i>	3 (3,7%)	1 (1,2%)	4 (4,9%)	67,1%
<i>Upper/Lower Confidence Bound</i>	3 (3,7%)	0 (0,0%)	3 (3,7%)	70,7%
Outros	17 (20,7%)	3 (3,7%)	20 (24,4%)	95,1%
Não especificado	2 (2,4%)	2 (2,4%)	4 (4,9%)	100,0%
Algoritmos de otimização da FA				
<i>Genetic Algorithm</i>	9 (12,7%)	0 (0,0%)	9 (12,7%)	12,7%
<i>Particle Swarm</i>	3 (4,2%)	4 (5,6%)	7 (9,9%)	22,5%
Baseados em Gradiente	4 (5,6%)	2 (2,8%)	6 (8,5%)	31,0%
<i>Sequential Quadratic Programming</i>	3 (4,2%)	1 (1,4%)	4 (5,6%)	36,6%
BFGS	3 (4,2%)	0 (0,0%)	3 (4,2%)	40,8%
Outros Algoritmos	3 (4,2%)	0 (0,0%)	3 (4,2%)	45,1%
Evolutivos				
NSGAI (III)	1 (1,4%)	2 (2,8%)	3 (4,2%)	49,3%
<i>Ranking and Selection</i>	2 (2,8%)	1 (1,4%)	3 (4,2%)	53,5%
Outros	8 (11,3%)	1 (1,4%)	9 (12,7%)	66,2%
Não especificado	20 (28,2%)	4 (5,6%)	24 (33,8%)	100,0%

4.6.2.1. Método de Simulação

Na Tabela 2 foi apresentado o primeiro aspecto estudado, que diz respeito ao método de simulação adotado pelos pesquisadores. Em 21,4% dos estudos analisados, não foi possível

identificar qual método de simulação foi utilizado. Uma sugestão para pesquisas futuras é destacar essa informação, pois os pesquisadores podem identificar na literatura o método de simulação mais adequado para seus problemas e quais outras ferramentas podem estar associadas a ele.

Entre os artigos analisados, cerca de 62,9% optaram por empregar métodos numéricos. Esses métodos são baseados em equações diferenciais e incluem técnicas como o Método de Elementos Finitos, Método de Diferenças Finitas, Fluidodinâmica Computacional, modelo de Gipps, método de Muskingum-Cunge, Simulação Eletromagnética, entre outros. Um exemplo de aplicação dos métodos numéricos é o uso do método de diferenças finitas para modelar o comportamento de sistemas hídricos submetidos a bombeamento para extração de água e tratamento de contaminantes (Yosefipoor *et al.*, 2022; Zhang *et al.*, 2022a, 2022b).

A construção de modelos pelo Método de Elementos Finitos esteve presente em muitos artigos, os quais foram utilizados para OvS em problemas de *design* de produtos e otimização de parâmetros de processo em diversas áreas, como metalurgia (Milani *et al.*, 2009), automotivo (Gu; Li; Dong, 2012), instrumentos musicais (Bharaj *et al.*, 2015), injeção plástica (Qian *et al.*, 2021), healthcare (Li *et al.*, 2017), entre outros.

Também relacionado a simulação numérica, os métodos de Fluidodinâmica Computacional, também conhecidos como volumes finitos ou simplesmente CFD (acrônimo em inglês para *Computational Fluid Dynamics*), são frequentemente empregados em problemas de *design* de produtos (Ahrari *et al.*, 2021; Gorissen; Dhaene; De Turck, 2009; He *et al.*, 2020; Yaohui, 2017). Neste tipo de simulação, devido ao tempo computacional demandado para execução dos experimentos, as técnicas de metamodelagem adaptativa proveem um caminho para otimizar as variáveis de *design* dos produtos em um tempo hábil.

Outra abordagem numérica comum é a simulação de tráfego. Nesse contexto, a metamodelagem adaptativa é empregado para minimizar o tempo médio de viagem, ajustando os tempos de ciclo dos sinais de trânsito e estabelecendo estratégias para projeto e adequação de vias urbanas utilizando modelos macro e microscópicos (Teixeira *et al.*, 2022; Yin; Liu; Zheng, 2022; Zheng *et al.*, 2019).

O segundo método de simulação mais utilizado foi a SED. A SED é uma técnica de modelagem e análise que permite estudar o comportamento de sistemas complexos ao longo do tempo, considerando eventos distintos e não contínuos. Nesse tipo de simulação, o sistema é representado por entidades individuais, e as mudanças ocorrem apenas nos momentos específicos em que eventos ocorrem, resultando em uma evolução discreta do sistema (Law, 2013). As aplicações de SED remetem a estudo de sistemas sujeitos a interdependência e

estocasticidade, tais como para alocação de recursos em processos fabris (Rosen; Harmonosky, 2005) e serviços de saúde (Coelho; Pinto, 2018) e para controle de estoque (Quan *et al.*, 2015)

Por outro lado, a Simulação Baseada em Agentes (SBA) é uma abordagem que enfoca a interação e comportamento de entidades autônomas, conhecidas como agentes, dentro de um ambiente simulado. Cada agente possui suas próprias regras, objetivos e capacidades, e a simulação consiste em observar como suas ações e interações afetam o sistema como um todo. Essa técnica é especialmente útil para estudar fenômenos complexos e emergentes que surgem das interações individuais dos agentes (Siebers *et al.*, 2010).

Este é o caso apresentado por Feygin *et al.* (2020), que considera em seu *framework* a interação entre os agentes usuários, operadores e autoridades, no contexto do sistema de transporte integrado, e suas interações são consideradas para desenvolver estratégias de otimização que visam melhorar a eficiência, a sustentabilidade e a qualidade da mobilidade. Na área de saúde, Kasaie e Kelton (2013) utilizaram metamodelagem adaptativa e SBA para otimização da alocação de recursos no controle de epidemia de influenza, de modo a minimizar os impactos e conter efetivamente o surto.

Além destes, também foi encontrado um estudo envolvendo simulação de Monte Carlo. Giahi, Mackenzie e Hu (2022) desenvolveram estratégias para aprimorar a operação e o dimensionamento de sistemas híbridos de energia limpa, considerando, via simulação de Monte Carlo, a variabilidade das fontes renováveis e as necessidades energéticas específicas.

Portanto, em resposta a segunda questão de pesquisa, os métodos de simulação que estão relacionados aos estudos de MAOVs são a simulação numérica, em especial o Método de Elementos Finitos, CFD e simulação de tráfego micro/macrocópica, seguida da SED, SBA e simulação de Monte Carlo.

4.6.2.2. Metamodelos e *Design* Experimentais

O segundo tópico analisado na Tabela 2 é o tipo de metamodelo utilizado. Nos estudos de metamodelagem adaptativa analisados, a maioria dos autores (62,5%) empregaram a metamodelagem por *Kriging* (e suas variantes). O *Kriging*, também conhecido como *Gaussian Process*, é uma abordagem Bayesiana não paramétrica, que utiliza variáveis seguindo uma distribuição gaussiana (Kleijnen, 2014). Seu conceito fundamental é atribuir uma probabilidade a um número infinito de funções.

O sucesso e popularidade do *Kriging* em estudos de MA pode estar relacionado a diversos fatores. Dentre estes fatores destaca-se o fato de o *Kriging* ser um metamodelo

estocástico, fornecendo como resposta tanto o valor previsto ($\hat{f}(\mathbf{x})$) quanto a incerteza da previsão ($\sigma(\mathbf{x})$) para determinado ponto \mathbf{x} , o que permite o cálculo de funções de aquisição que balanceiem busca global e local. Além disto, o *Kriging* consegue aproximar funções complexas, com alta não linearidade. Outro fator, é que o *Kriging* foi o metamodelo utilizado no algoritmo EGO, proposto inicialmente por Jones, Schonlau e Welch (1998), e muito utilizado no contexto de OvS e *machine learning* (Kleijnen, 2009; Mehdad; Kleijnen, 2015; Rojas-Gonzalez; Van Nieuwenhuysse, 2020). Assim como destacado na Quadro 3, o *Kriging* tem sido muito utilizado para aproximar modelos de simulação numérica, mas também apresenta aplicações em modelos de SED e simulação de Monte Carlo.

Quadro 3: Artigos por metamodelo adotado versus método de simulação

Metamodelo/ Simulação	Simulação numérica	SED	SBA	Monte Carlo
<i>Kriging</i>	(Ahrari et al., 2021; Bharaj et al., 2015; Boukouvala; Ierapetritou, 2013; Candelieri et al., 2022; Candelieri; Perego; Archetti, 2018; Gengembre et al., 2012; Gorissen; Dhaene; De Turck, 2009; Gu; Li; Dong, 2012; He et al., 2020; Ivanova; Kuhnt, 2014; Kugalur Palanisamy et al., 2022; Li et al., 2017; Liu et al., 2022; Luo et al., 2020, 2022; Parnianifard et al., 2020b, 2020c; Qian et al., 2021; Tay; Osorio, 2022; Teixeira et al., 2022; Ur Rehman; Langelaar, 2016; Wang et al., 2020; Wang; Ierapetritou, 2017, 2018; Xie; Yi; Zheng, 2021; Xiong et al., 2021; Yaohui, 2017; Yin; Liu; Zheng, 2022; Zhang et al., 2022a, 2022b; Zheng et al., 2019; Zhong et al., 2021; Zhou; Turng, 2007)	(Coelho; Pinto, 2018; Han; Ouyang, 2022; Quan et al., 2015; Ungredda; Pearce; Branke, 2022; Xie; Frazier; Chick, 2016)		(Giahi; Mackenzie; Hu, 2022)
RNA	(Bamdad; Cholette; Bell, 2020; Gorissen; Dhaene; De Turck, 2009; Hasanzadeh; Saadatpour; Afshar, 2020; Saadatpour et al., 2021; Zhao; Minsker, 2021)			
Regressão Polinomial	(Castillo et al., 2010; Gu; Li; Dong, 2012; Milani et al., 2009; Zhang et al., 2009; Zheng et al., 2019)	(Rosen; Harmonosky, 2005)	(Kasaie; Kelton, 2013)	
RBF	(Gorissen; Dhaene; De Turck, 2009; Gu; Li; Dong, 2012; Zheng; Li, 2023)	(Fan; Hu, 2018)		
SVM	(Gorissen; Dhaene; De Turck, 2009; Yosefipoor et al., 2022; Zhao; Minsker, 2021)			
<i>Splines</i>	(Gorissen; Dhaene; De Turck, 2009)			
Outros	(Candelieri; Perego; Archetti, 2018; Rodriguez; Ludkovski, 2020; Zhao; Minsker, 2021)		(Feygin et al., 2020)	

O segundo tipo de metamodelo mais frequente foram as RNAs, estando presente em 8,4% dos trabalhos. As RNAs são um conjunto de modelos computacionais inspirados na estrutura e funcionamento do sistema nervoso humano. O funcionamento das RNAs se assemelha ao processamento de informações realizado pelos neurônios do cérebro, com uma estrutura composta por camadas interconectadas, no qual cada neurônio, ou unidade, processa e transmite sinais ponderados para as camadas subsequentes (Zhao; Minsker, 2021).

As RNAs tem sido empregadas em estudos mono e multiobjetivos, porém focados em modelos de simulação numérica. Como exemplo, Saadatpour et al. (2021) utilizaram RNA na

metamodelagem adaptativa de modelos hídricos para projetar as localizações ideais do sistema de retirada seletiva e derivar regras de operação para maximizar a geração de energia hidrelétrica, minimizando ao mesmo tempo os danos ambientais e assegurando a qualidade da água no reservatório.

O uso de Regressão Polinomial esteve presente em 8,4% dos trabalhos. Segundo Myers, Montgomery e Anderson-Cook (2016), o conceito da aproximação polinomial de uma determinada resposta é baseado na teoria de expansão da série de Taylor em torno de um conjunto de dados. Neste contexto, destaca-se a Metodologia de Superfície de Resposta, que prevê a construção iterativa de metamodelos de Regressão Polinomial para aproximar funções *black-box*. Aplicações de Regressão Polinomial são encontradas em estudos de simulação numérica (Castillo *et al.*, 2010; Gu; Li; Dong, 2012; Milani *et al.*, 2009; Zhang *et al.*, 2009; Zheng *et al.*, 2019), SED (Rosen; Harmonosky, 2005), e SBA (Kasaie; Kelton, 2013).

Outros tipos de metamodelos também foram encontrados na literatura, tais como RBF (Fan; Hu, 2018; Gorissen; Dhaene; De Turck, 2009; Gu; Li; Dong, 2012; Zheng; Li, 2023), *Support Vector Machine* (SVM) (Gorissen; Dhaene; De Turck, 2009; Yosefipoor *et al.*, 2022; Zhao; Minsker, 2021), *Splines* (Gorissen; Dhaene; De Turck, 2009; Keys; Rees; Greenwood, 2002), entre outros.

Por fim, destaca-se o uso de técnicas de *Ensembles*, que combinam diversos metamodelos para criar um metamodelo com maior capacidade de aprendizagem. Alguns trabalhos utilizaram *Ensembles* em suas propostas, como Bamdad, Cholette e Bell (2020), que aplicaram um metamodelo *Ensemble (Committee)* para otimizar o *design* de construções e reduzir o consumo de energia. Candelieri, Perego e Archetti (2018) propuseram uma metodologia de otimização das operações de bombas em sistemas de distribuição de água usando Random Forest, um modelo *Bagging*.

Ademais, Gorissen, Dhaene e De Turck (2009) combinaram diferentes metamodelos, por média simples, para prever problemas de *design* de produtos. Gu, Li e Dong (2012) utilizaram a intersecção das melhores soluções previstas por diferentes metamodelos para guiar a simulação. Zhao e Minsker (2021) propuseram o uso de *Majority Voting* (combinando RNA, SVM e Regressão Logística) para classificação de soluções viáveis em modelos hídricos.

O terceiro tópico analisado na Tabela 2 é quanto a técnica de *Design Experimental* utilizado para gerar a base de treinamento inicial para a metamodelagem adaptativa. Dos artigos analisados, a maioria (59,2%) optaram pelo uso do LHD, seguido pelo *Design Aleatório* (15,5%), Fatorial Completo (4,2%), *Grid Search* (2,8%) e população inicial de algoritmos evolutivos (2,8%).

O LHD é um método popular de amostragem estratificada em estudos de metamodelagem. Ele divide as variáveis em n intervalos disjuntos com igual probabilidade, selecionando aleatoriamente um ponto de cada célula de um hiper-retângulo definido por uma matriz $n \times d$, no qual d é o número de variáveis e n é o número de amostras desejado. Mais detalhes sobre esta técnica podem ser encontrados em Parnianifard *et al.* (2018).

Com base nestas informações é possível responder a terceira e quarta questões de pesquisa deste trabalho. O principal tipo de metamodelo utilizado em estudos de MAOvS foi o *Kriging*, seguido das RNAs, Regressão Polinomial, RBF, SVM e *Splines*. A respeito do *Design Experimental* utilizado para delineamento da base de treinamento inicial, destaca-se o uso da técnica de LHD, que esteve presente na maioria dos trabalhos analisados, mostrando-se robusta em diferentes contextos e objetivos.

4.6.2.3. Funções de Aquisição e Algoritmos de Otimização

O quarto tópico analisado na Tabela 2 e destacado na Figura 14 (a) é em respeito a função de aquisição adotada pelos artigos. A FA é o mecanismo responsável por guiar a seleção de novos pontos a serem simulados, sendo um dos principais fatores de impacto nos *frameworks* de MA.

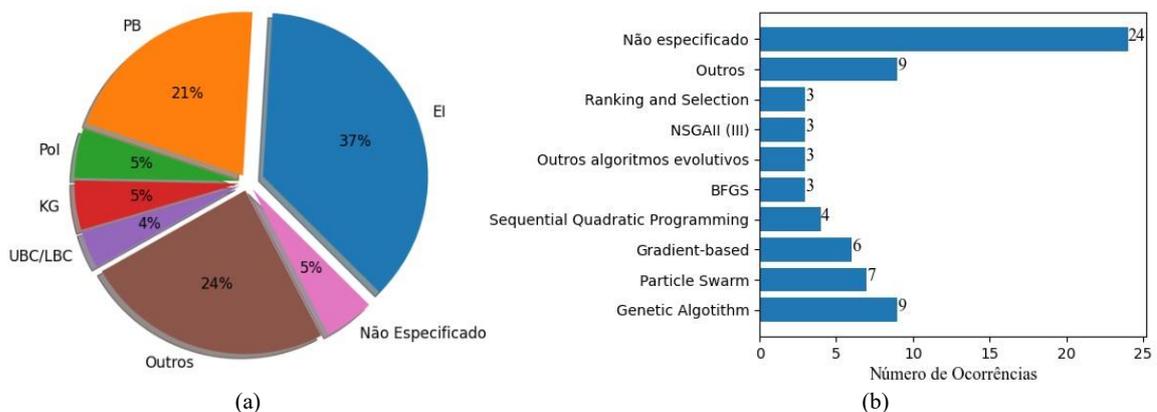


Figura 14: Funções de Aquisição (a) e Método de otimização (b) mais usados.

A principal FA utilizada foi a *Expected Improvement* e suas variações. A formulação tradicional da EI envolve o balanceamento entre regiões promissoras de conterem o ótimo e regiões com alta incerteza de previsão, assim como demonstrado na Eq. (7) (Boukouvala; Ierapetritou, 2013; Ivanova; Kuhnt, 2014; Wang; Ierapetritou, 2017).

$$E[I(\mathbf{x})] = (f_{min} - \hat{f}(\mathbf{x}))\Phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma(\mathbf{x})}\right) \quad (7)$$

O menor valor observado no conjunto de dados de treinamento é denotado como f_{min} , a previsão do metamodelo (*Kriging*) para um determinado ponto \mathbf{x} é representado por $\hat{f}(\mathbf{x})$, e $\sigma(\mathbf{x})$ é o desvio padrão da previsão. Φ e ϕ referem-se às funções de distribuição cumulativa normal e densidade de probabilidade. Avaliando as derivadas da função EI para $\hat{f}(\mathbf{x})$ e $\sigma(\mathbf{x})$ revela que EI é maior quando $\hat{f}(\mathbf{x})$ é menor e $\sigma(\mathbf{x})$ é maior. Assim, maximizar a função EI busca identificar regiões com *trade-offs* ótimos entre busca local (menor $\hat{f}(\mathbf{x})$) e busca global (maior $\sigma(\mathbf{x})$).

Muitos trabalhos também utilizaram variações do EI. Exemplos incluem o *Augmented* EI, também conhecido como *Stochastic* EI que adiciona o fator de estocasticidade da simulação à formulação tradicional do EI (Candelieri; Perego; Archetti, 2018; Han; Ouyang, 2022). Em problemas de otimização com restrições, também é possível encontrar FAs que adicionam um fator multiplicativo a EI para considerar a probabilidade de viabilidade da solução em sua formulação, como é o caso apresentado por Wang e Ierapetritou (2018). Quando se trata de otimização multiobjetivo, o *Expected Hypervolume Improvement* (EHV) calcula a melhoria esperada no hipervolume da Fronteira de Pareto que uma nova solução candidata pode alcançar, quando adicionada ao conjunto atual de soluções (Candelieri *et al.*, 2022; Han; Ouyang, 2022).

Uma parte considerável dos trabalhos (22,7%) utilizou a *Predicted Best* (PB) para guiar seu processo de busca. Nesta abordagem, o próximo ponto a ser amostrado corresponde aquele que maximiza ou minimiza o valor de função objetivo previsto pelo metamodelo. A PB concentra esforços em regiões promissoras de conterem o ótimo, não considerando a incerteza de previsão do metamodelo. A PB é ideal quando o metamodelo apresenta uma boa aproximação em diferentes regiões do espaço de busca, ou seja, um baixo erro associado. Embora seja a mais simples das FA, ela apresenta excelentes resultados em diversas áreas (Fan; Hu, 2018; Zhang *et al.*, 2009; Zhou; Turng, 2007). A PB pode ser formulado seguindo a Eq. (8).

$$PB(\mathbf{x}) = \hat{f}(\mathbf{x}) \quad (8)$$

Em 4,9% dos trabalhos, a FA utilizada foi a *Probability of Improvement* (PoI) e suas variações (He *et al.*, 2020; Lee; Zhou; Chen, 2023; Yin; Liu; Zheng, 2022). A ideia básica da PoI é calcular a probabilidade de que uma nova solução candidata melhore o desempenho em relação à melhor solução atualmente conhecida. Para fazer isso, ela utiliza a distribuição de probabilidade estimada para o valor da função objetivo no espaço de busca. Assim como demonstrado na Eq. (9), é possível observar que a EI é uma expansão da formulação da PoI.

$$PoI(\mathbf{x}) = \Phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma(\mathbf{x})}\right) \quad (9)$$

FAs baseadas em gradiente também foram utilizadas nos artigos analisados, como *Knowledge-gradient* (KNG) (Xie; Frazier; Chick, 2016), *Hierarchical KNG* (Mes; Powell; Frazier, 2011) e *Robust KNG* (Hasanzadeh; Saadatpour; Afshar, 2020). A KNG tradicional pode ser representada pela Eq. (10) (Chen; Liu; Tang, 2021).

$$KNG^n(\mathbf{x}) = \mathbb{E}\left[\max_{\mathbf{x}'} \mu^{n+1}(\mathbf{x}') - \max_{\mathbf{x}''} \mu^n(\mathbf{x}'') \mid \mathcal{F}^n, \mathbf{x}^{n+1} = \mathbf{x}\right] \quad (10)$$

Sendo que \mathcal{F}^n representa os dados coletados até o momento $\{(x^1, f(x^1)), \dots, (x^n, f(x^n))\}$, $\mu^n(\mathbf{x})$ corresponde à média posterior após n dados coletados e adotando-se a premissa que a próxima amostra a ser avaliada \mathbf{x}^{n+1} será \mathbf{x} .

Outra FA também utilizada é a *Upper/Lower Confidence Bound* (UCB/LCB), que é baseada na construção de intervalos de confiança em torno das previsões feitas por um modelo probabilístico da função objetivo. Esses intervalos permitem selecionar pontos com alta incerteza, facilitando a exploração de regiões desconhecidas do espaço de busca (Desautels; Krause; Burdick, 2014; Yin; Liu; Zheng, 2022). A formulação do UCB e LCB é utilizada em casos de maximização e minimização, respectivamente, e pode ser descrita segundo as equações Eq. (11) e Eq. (12) (Candelieri; Perego; Archetti, 2018).

$$UBC(\mathbf{x}) = \hat{f}(\mathbf{x}) + k\sigma(\mathbf{x}) \quad (11)$$

$$LBC(\mathbf{x}) = \hat{f}(\mathbf{x}) - k\sigma(\mathbf{x}) \quad (12)$$

Nas Eqs. (11) e (12), $\hat{f}(\mathbf{x})$ e $\sigma(\mathbf{x})$ representam o valor médio previsto e o desvio padrão da previsão para \mathbf{x} , e $k \geq 0$ é o parâmetro para gerenciar o *trade-off* de busca global/local ($k = 0$ significa busca local pura).

Por fim, também foram observadas outras FAs, tais como as baseadas em entropia (Rodriguez; Ludkovski, 2020; Wang; Yuan; Ng, 2020), baseada em quartis (Han; Ouyang, 2022; Rodriguez; Ludkovski, 2020), *Jackknife Leave-one-out* (Parnianifard *et al.*, 2020c), *Value of Information* (Ungredda; Pearce; Branke, 2022), entre outros.

Vale ressaltar que uma FA é a métrica de performance pela qual uma solução candidata é avaliada, sendo aquela com melhor performance a escolhida como próximo ponto a ser avaliado no modelo de simulação. A busca pela solução candidata que maximiza (ou minimiza) a FA é guiada por um algoritmo de otimização. Nos artigos analisados, 12,7% utilizaram o Algoritmo Genético para otimização de suas FAs (Mehdad; Kleijnen, 2015; Qian *et al.*, 2021; Zhao; Minsker, 2021), assim como apresentado no quinto tópico da Tabela 2 e destacado na

Figura 14 (b). Mais informações sobre o AG podem ser encontradas na Seção 2.4 deste trabalho.

O segundo algoritmo mais usado é o *Particle Swarm Optimization* (9,9%) (Gengembre *et al.*, 2012; Hasanzadeh; Saadatpour; Afshar, 2020; Parnianifard *et al.*, 2020c), seguido por algoritmos baseados em gradiente (8,5%), tais como *Steepest Ascent* (Kasaie; Kelton, 2013; Scott; Frazier; Powell, 2011) e *Generalized reduced-gradient* (Kim; Boukouvala, 2020), *Sequential Quadratic Programming* (5,6%) (Li *et al.*, 2017; Tay; Osorio, 2022; Wang; Ierapetritou, 2017), *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) (Candelieri; Perego; Archetti, 2018; Xie; Yi; Zheng, 2021), entre outros.

Respondendo à questão de pesquisa cinco (QP5), a FA mais utilizada foi a EI e suas variações, mostrando bons resultados em balancear busca global e local ao longo do processo de MA. A EI é utilizada para atribuir um valor de performance às soluções (ou pontos) do espaço de busca do problema. Complementarmente, o uso do AG foi o método de otimização mais utilizado para encontrar a solução que otimiza a FA escolhida.

4.6.2.4. Etapas do *Framework* de MAOvS Utilizado/Proposto

Esta seção discute as etapas seguidas no *framework* de MA utilizados/propostos pelos trabalhos. Muitos autores destacam como contribuição central de seus artigos a proposição de *frameworks* de metamodelagem adaptativa para otimização de problemas exclusivos de suas áreas de atuação ou de cunho geral (Qian *et al.*, 2021; Xiong *et al.*, 2021; Zhang *et al.*, 2009; Zhou; Turng, 2007). Outros autores ainda buscaram contribuições com adaptações pontuais em etapas específicas de *frameworks* existentes (Gengembre *et al.*, 2012; Mes; Powell; Frazier, 2011; Ur Rehman; Langelaar, 2016). Vale ressaltar que em 7,4% dos trabalhos não foi possível identificar as etapas/*frameworks* seguidas pelos autores, logo estes foram excluídos das análises apresentadas nesta seção.

Como destacado na Figura 15 (a), os trabalhos analisados utilizaram entre três e oito etapas em seus *frameworks*, com mediana centrada em cinco etapas. Como é possível observar na Figura 15 (b), foram encontradas dez etapas comuns em pelo menos dois trabalhos, ordenadas de acordo com a sequência que aparecem nos *frameworks*. Dentre estas destaca-se as chamadas etapas fundamentais, que estiveram presentes na maioria dos trabalhos, são elas: Amostragem Inicial, Treinamento do Metamodelo; Otimização da Função de Aquisição, Avaliação na Simulação e Critério de Parada.

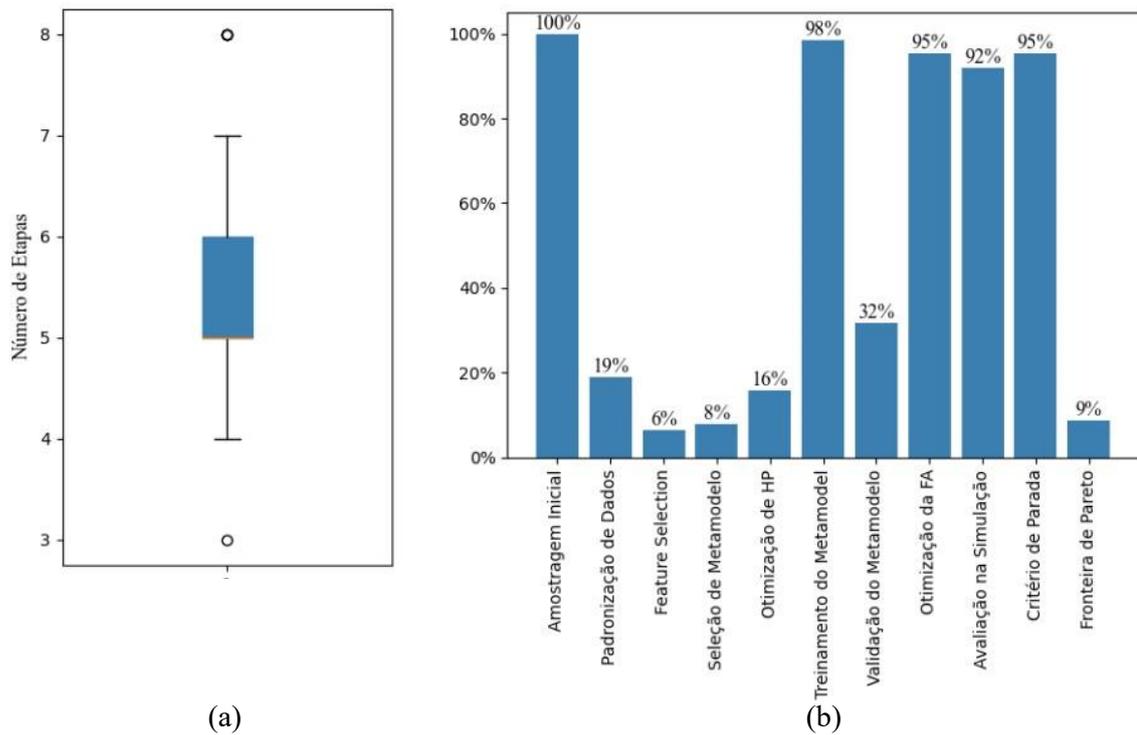


Figura 15: Número de etapas utilizadas pelos trabalhos (a) e percentual de trabalhos que utilizaram cada etapa em seus *Frameworks* Utilizados/Propostos (b).

A primeira etapa, Amostragem Inicial, esteve presente em 100% dos trabalhos. Ela é responsável pela geração da base de treinamento inicial do metamodelo, e como ressaltado na Seção 4.6.2.2, o método mais utilizado para sua geração foi o LHD. Quanto o tamanho da amostra inicial, o mais frequente foi a utilização da regra de $10d$, onde d representa a dimensão do problema.

A segunda etapa, Padronização de Dados foi encontrada em 19,0% dos artigos, correspondendo processo de transformar os valores das variáveis na base de treinamento, tornando-os comparáveis e facilitando a análise estatística. A maioria destes trabalhos utilizaram a padronização *Min-Max*, que escala os valores de uma variável para um intervalo específico, geralmente entre 0 e 1, preservando a relação de proporção entre os valores originais (Del Castillo; Santiago, 2011; Zheng; Li, 2023). Outras formas de padronização também foram utilizadas, como a normalização, que transforma os dados para que tenham média zero e desvio padrão igual a um (Teixeira *et al.*, 2022), e *One-hot-encoder*, o qual converte variáveis categóricas em uma representação vetorial binária (Milani *et al.*, 2009).

Com presença em apenas 6,3% dos trabalhos, a etapa de *Feature Selection* é uma importante técnica no contexto de ML. *Feature Selection* é o processo de escolher um subconjunto relevante e informativo das variáveis dos dados para melhorar o desempenho do

modelo de ML. Esta técnica é geralmente aplicada quando o número de *Features* (ou variáveis de decisão no caso de OvS) é relativamente grande, sendo necessário a redução de dimensionalidade do problema. Nesta etapa diversas técnicas podem ser utilizadas, como a avaliação do *p-value* em regressão de primeira ordem (Kasaie; Kelton, 2013), Análise dos Componentes Principais (Hasanzadeh; Saadatpour; Afshar, 2020), e método de Morris (Liu *et al.*, 2022).

A quarta etapa, Seleção de Metamodelo esteve presente em 7,9% dos artigos, que após a geração da amostra inicial ou de forma recorrente a cada iteração do algoritmo de MA, treina dois ou mais tipos de metamodelo e escolhe o que melhor se ajusta aos dados coletados até o momento. No *framework* proposto por Kim e Boukouvala (2020), a cada iteração do algoritmo é selecionado o melhor metamodelo (RNA ou *Kriging*) via *k-fold cross validation*. Em outro exemplo, Gorissen, Dhaene e De Turck (2009) utiliza AG para definir a melhor combinação de metamodelos a serem utilizados no modelo de *Ensemble*, conjuntamente com seus hiperparâmetros.

A seleção apropriada dos hiperparâmetros tem um papel fundamental no treinamento dos algoritmos de ML, reduzindo seu erro e podendo até aumentar a velocidade de treinamento (Amaral *et al.*, 2022b). Neste sentido, a etapa de Otimização de HPs esteve presente em 15,9% dos trabalhos. Esta etapa corresponde a utilização de técnicas de seleção/otimização para encontrar a combinação de HPs que minimizem o erro do metamodelo, podendo ser feita apenas uma vez ou de forma recorrente ao longo das iterações. Esta seleção pode ser realizada com algoritmos de otimização, como o AG (Gorissen; Dhaene; De Turck, 2009; Zhou; Turng, 2007) e *Particle Swarm Optimization* (Xiong *et al.*, 2021), ou técnicas de busca exaustiva, como o *Grid Search* (Bamdad; Cholette; Bell, 2020; Briffoteaux *et al.*, 2020; Kugalur Palanisamy *et al.*, 2022).

A sexta etapa, Treinamento do Metamodelo, esteve presente em 98% dos trabalhos, sendo considerada uma etapa fundamental para *frameworks* de MA. Como destacado na Seção 4.6.2.2, o metamodelo mais utilizado pelos autores foi o *Kriging*, seguido de RNA e Regressão Polinomial. Nesta etapa, a cada iteração do algoritmo, o metamodelo é treinado com a base de dados (experimentos) simulada até o momento, e então é utilizado para fornecer previsões para soluções ainda não avaliados pelo modelo de simulação.

Ao longo das iterações, com o aumento da base de dados, a capacidade que o metamodelo tem de realizar previsões assertivas tende a aumentar. Todavia, esta capacidade depende de vários fatores, assim como o tipo de metamodelo, os seus HPs, a qualidade e tamanho da base de dados, entre outros. Neste sentido, uma boa prática em ML é quantificar a

capacidade deste metamodelo por meio cálculo de seu erro/ajuste. Esta avaliação da qualidade do metamodelo corresponde a etapa de Validação do Metamodelo, presente em 31,7% dos artigos, a qual visa avaliar o quão bem o metamodelo representa o modelo de simulação original, com o intuito de informar sua qualidade ou guiar decisões ao longo das iterações.

Dentre os artigos que utilizaram esta etapa, 50,0% aplicaram as técnicas de *Leave-one-out cross validation* e *k-fold cross validation*. A técnica de *k-fold cross validation* divide igualmente e aleatoriamente os dados de treinamento em k partes. Durante o processo, $k-1$ partes são utilizadas para treinar o algoritmo e a parte restante é reservada para o teste. Esse procedimento é repetido k vezes, cada vez selecionando uma parte diferente, denotada como k_i' , para o teste, com i variando de 1 a k . Em seguida, o erro é calculado para cada arranjo nas k iterações de validação. No caso do *Leave-one-out cross validation* o número de partes k é igual ao número de amostras na base de treinamento. A performance do algoritmo é medida com base na média dos erros obtidos nas k iterações de validação, o que proporciona uma métrica mais robusta do que se fosse medida em apenas uma observação isolada (Bergmeir; Benítez, 2012; Pedrielli; Wang; Ng, 2020).

Assim como apontado no primeiro tópico da Tabela 3, as principais métricas de performance utilizadas pelos trabalhos analisados foram o *Mean Square Error* (MSE)/*Root MSE* (RMSE), presente em 13,8% dos artigos, 9,2% utilizaram o coeficiente de correlação (R), coeficiente de determinação (R^2) ou R^2 ajustado (R^2_{adj}), e 5,7% dos trabalhos utilizaram o *Mean Relative Error* ou *Root Mean Relative Error*. Mais detalhes sobre estas métricas sugere-se Chicco, Warrens e Jurman (2021).

Tabela 3: Métricas e critérios de parada.

Métrica de Performance	Mono-objetivo	Multiobjetivo	Total	Acum.
MSE/RMSE	9 (10,3%)	3 (3,4%)	12 (13,8%)	13,8%
R , R^2 e R^2_{adj}	6 (6,9%)	2 (2,3%)	8 (9,2%)	23,0%
<i>Mean Relative Error/ Root Relative Square Error</i>	4 (4,6%)	1 (1,1%)	5 (5,7%)	28,7%
Outros	16 (18,4%)	7 (8%)	23 (26,4%)	55,2%
Não usa	18 (20,7%)	7 (8%)	25 (28,7%)	83,9%
Não especificado	11 (12,6%)	3 (3,4%)	14 (16,1%)	100,0%
Critério de Parada				
<i>Budget</i> de simulações	35 (42,2%)	10 (12%)	45 (54,2%)	54,2%
Erro do Metamodelo < <i>threshold</i>	9 (10,8%)	1 (1,2%)	10 (12,0%)	66,3%
n iterações sem melhoria no valor da FO	8 (9,6%)	1 (1,2%)	9 (10,8%)	77,1%
Melhor valor da FA < <i>threshold</i>	3 (3,6%)	2 (2,4%)	5 (6%)	83,1%
Outros	4 (4,8%)	2 (2,4%)	6 (7,2%)	90,4%
Não especificado	6 (7,2%)	2 (2,4%)	8 (9,6%)	100,0%

Presente em 95,2% dos trabalhos, a etapa de Otimização da Função de Aquisição também é considerada uma etapa fundamental para *frameworks* de MA. Nesta etapa é(são) selecionado(s) o(s) próximo(s) ponto(s) a ser(em) simulado(s) e adicionado(s) à base de treinamento. Embora a seleção de apenas um único ponto a cada iteração do algoritmo, foram encontrados trabalhos que selecionavam, a cada iteração, múltiplos pontos a serem simulados (Bamdad; Cholette; Bell, 2020; Desautels; Krause; Burdick, 2014; Kim; Boukouvala, 2020; Zhang *et al.*, 2022b), principalmente com o objetivo de utilizarem os benefícios do processamento paralelo, que foi considerado por 14,7% dos trabalhos.

Como discutido na Seção 4.6.2.3, o principal método de otimização utilizado pelos autores foi o AG, representando 12,7% dos trabalhos. O sucesso desta técnica reflete sua flexibilidade em diversas aplicações, conseguindo lidar com variáveis contínuas, binárias, discretas e categóricas, alta não linearidade e descontinuidade nas funções objetivos (Chaudhry; Luo, 2005).

Uma vez selecionado(s) o(s) próximo(s) ponto(s) a ser(em) adicionado(s) à base de treino, a próxima etapa é avaliá-lo(s) no modelo de simulação (etapa de Avaliação na Simulação). Nesta etapa fundamental, a escolha do método de simulação a ser empregada está associada ao problema abordado e habilidades do modelador. Porém, ressalta-se que os métodos de MA foram muito aplicados em simulação numérica, principalmente em modelos de Elementos Finitos, CFD e simulação de tráfego, como destacado na Seção 4.6.2.1. É importante ressaltar que, ao considerar simulações estocásticas, é recomendado simular replicações para um mesmo ponto, sendo a resposta final a média destas replicações. A determinação do número ideal de replicações leva em consideração fatores como a variabilidade do modelo, o erro aceitável, as limitações de *budget*, e pode ser fixa ou adaptativa ao longo das iterações da otimização (Pedrielli; Wang; Ng, 2020; Quan *et al.*, 2015).

Os *frameworks* de MA retratam um algoritmo recursivo ao qual a cada iteração um conjunto de etapas se repetem até que um critério de parada seja atingido. Em 95,2% dos artigos os autores deixaram claro o critério utilizado para finalizarem o processo de metamodelagem e otimização. Como discriminado na Tabela 3, a maioria dos trabalhos definiram o *budget* de simulações, isto é, número máximo de execuções do modelo de simulação original, como critério de parada de seus *frameworks*. Já 12,0% dos trabalhos utilizaram o resultado da etapa de Validação do Metamodelo como critério de parada, ou seja, é definido um nível de erro *threshold* ao qual, caso atingido pelo metamodelo, o processo de otimização é encerrado.

Em 10,8% dos trabalhos foi definido que a otimização seria encerrada após n iterações sem melhoria na FO. Outros 6,0% definiram um *threshold* para o valor da FA, que se não

atingido durante a etapa de Otimização da FA encerra o algoritmo. Além disto, vale a pena destacar que 22,1% dos artigos utilizaram mais de um critério de parada em seus *frameworks*, como foi o caso apresentado em Gorissen, Dhaene e De Turck (2009), Mehdad e Kleijnen (2015) e Parnianifard *et al.* (2020c).

A etapa que corresponde à construção da Fronteira de Pareto ao longo das iterações do algoritmo é aplicável apenas a *frameworks* voltados para otimização multiobjetivo. Embora esta etapa esteja presente em apenas 8,8% dos trabalhos, ela foi utilizada em 37,5% dos trabalhos de otimização multiobjetivo. Mais informações sobre esta etapa podem ser encontradas em Han e Ouyang (2022) e Yosefipoor *et al.* (2022).

Com base nas discussões desta seção é possível responder a sexta questão de pesquisa (QP6). Nos trabalhos foram encontradas 10 etapas presentes em pelo menos dois artigos, sendo elas classificadas em etapas fundamentais (percentual $\geq 90\%$), etapas recomendadas ($10\% \leq$ percentual $< 90\%$) e opcionais (percentual $\leq 10\%$). Com base neste critério tem-se:

- Etapas fundamentais: Amostragem Inicial, Treinamento do Metamodelo, Otimização da FA, Avaliação na Simulação e Critério de Parada;
- Etapas recomendadas: Padronização dos Dados (Fundamental a depender do tipo de metamodelo), Otimização de HP e Validação do Metamodelo;
- Etapas opcionais: *Feature Selection*, Seleção de Metamodelos, Fronteira de Pareto (em caso de problemas multiobjetivo).

4.7. Discussões e Futuras Direções

Os principais objetivos desta RSL foram (a) identificar e explorar os principais aspectos relacionados a MAOVs, (b) analisar e sumarizar os artigos científicos a fim de responder as questões de pesquisa, e (c) delimitar o estado da arte, lacunas, tendências e oportunidades para estudos futuros.

Vale ressaltar que o objetivo deste trabalho não é exaurir toda literatura de MAOVs, mas prover uma análise sistemática de uma amostra significativa dos trabalhos acerca do tema. Neste contexto, a aplicação da metodologia de RSL mostrou-se eficaz em guiar uma análise sistemática e imparcial da literatura, as quais permitiram responder consistentemente as sete QPs levantadas neste estudo.

Considerando as QPs, conclui-se que muitos estudos aplicaram métodos propostos em funções matemáticas sintéticas com forma de validarem/compararem suas propostas. Além disto, destacam-se aplicações para design de produto e otimização de processos, especialmente

em gestão hídrica e transporte/logística, atuando principalmente em problemas de otimização mono-objetivo e com variáveis contínuas.

Os problemas abordados são, inicialmente, representados computacionalmente por modelos, que posteriormente serão utilizados no processo de otimização. Estes modelos podem ser desenvolvidos por meio de diversas técnicas de modelagem e simulação, aos quais se destacam a simulação numérica, SED, SBA e simulação de Monte Carlo. Estes possuem como característica elevada complexidade e alto custo computacional na execução das simulações, impulsionando a necessidade de metamodelagem para maior abstração do problema. Neste quesito, os principais tipos de metamodelos utilizados nos artigos analisados foram o *Kriging*, RNA, Regressão Polinomial, SVM e *Splines*. Para o treinamento destes metamodelos, a maioria dos trabalhos optaram por delinear a base de dados experimentais por meio da técnica conhecida como LHD.

Outros aspectos analisados por esta revisão foram a FA e o método de otimização utilizados para determinar o próximo ponto a ser avaliado na simulação. Por ser considerada um dos principais fatores envolvidos na metamodelagem adaptativa, muitas FAs foram propostas pelos artigos. Entretanto, muitos autores utilizaram o EI ou suas variações. Esta FA tem como objetivo balancear busca local e global e está diretamente associada ao uso do *Kriging*, o qual é utilizado para prever o valor de $\hat{f}(\mathbf{x})$ e $\sigma(\mathbf{x})$ para as possíveis soluções do problema. Parte considerável dos trabalhos também utilizaram o PB, ou seja, consideram apenas a busca local por meio da análise do $\hat{f}(\mathbf{x})$ predito pelo metamodelo.

Uma possível lacuna a ser explorada é o uso de métodos *Ensemble* para construção dos metamodelos. Embora tenham sido encontrados trabalhos envolvendo métodos *Ensembles*, não foram encontrados estudos envolvendo *Ensembles* homogêneos, ou seja, que combinam múltiplas instâncias (modelos) do mesmo tipo de algoritmos de ML para melhorar a precisão e desempenho geral. Como exemplo deste tipo de método, pode-se citar o *Bagging*, o qual treina e agrega n modelos por meio da técnica de *Boostrapping*, sendo que o valor da previsão final deste metamodelo é dado pela média das previsões individuais de cada modelo treinado. Uma vantagem do *Bagging* é a possibilidade de calcular o desvio padrão da previsão destes modelos, o qual habilita o uso de diversas FAs, como o EI, UBC/LBC, PoI, entre outros.

Além disto, esta revisão apresentou uma visão abrangente dos *frameworks* de MA propostos na literatura. A análise revelou estágios-chave compartilhados em diferentes trabalhos, categorizados como fundamentais, recomendados e opcionais com base em sua prevalência. Os estágios fundamentais incluem Amostragem Inicial, Treinamento do Metamodelo, Otimização da Função de Aquisição, Avaliação na Simulação e Critério de

Parada. Estágios recomendados englobam Padronização dos Dados, Otimização de Hiperparâmetros e Validação do Metamodelo. Estágios opcionais consistem *Feature Selection*, Seleção de Metamodelos, e Fronteira de Pareto (em caso de problemas multiobjetivo).

Baseado nos artigos analisados nesta RSL é possível identificar tendências e lacunas presentes na literatura, as quais refletem oportunidades para futuras pesquisas envolvendo o contexto de MAOvS. Sobre os métodos adotados, o progresso na inteligência computacional tem contribuído para a melhoria dos algoritmos de aprendizado, o que consequentemente pode ser explorado na criação de metamodelos mais precisos e flexíveis.

5. FRAMEWORK DE METAMODELAGEM PROPOSTO

5.1. Indicação, Requisitos e Benefícios do *Framework* Proposto

Frameworks são estruturas pré-definidas que oferecem um conjunto de ferramentas, componentes e padrões de desenvolvimento para facilitar a criação de aplicações ou a execução de tarefas complexas. A utilização de um *framework* apresenta diversas vantagens, como a reusabilidade de componentes e ferramentas em diferentes projetos, economizando tempo e esforço. Além disso, *frameworks* impõem um conjunto de boas práticas e padrões que garantem a consistência e a qualidade do código desenvolvido. A padronização também facilita a manutenção do código, permitindo que diferentes desenvolvedores compreendam e modifiquem a aplicação com maior facilidade. *Frameworks* frequentemente incluem otimizações e soluções pré-construídas para problemas comuns, o que pode melhorar a performance da aplicação. Ademais, ao fornecer ferramentas e componentes prontos para uso, um *framework* permite que os desenvolvedores se concentrem nas funcionalidades específicas de sua aplicação, aumentando a produtividade (Coelho *et al.*, 2011).

Neste sentido, o *framework* AMSO foi desenvolvido para otimizar problemas de SED utilizando técnicas avançadas de metamodelagem adaptativa, Bagging, GBT, otimização de hiperparâmetros e meta-heurísticas. O AMSO é indicado para problemas complexos de SED, sendo ideal para modelos que demandam elevado tempo computacional para execução e necessitam de soluções eficientes dentro de um orçamento restrito de simulações. O *framework* é especialmente útil em ambientes industriais, podendo ser aplicado na otimização de alocação de recursos, controle de estoques, programação de produção e outros processos críticos. Além disso, o AMSO é uma ferramenta valiosa para pesquisadores que trabalham com otimização baseada em simulação e buscam novas abordagens e técnicas para resolver problemas complexos.

Para a utilização do AMSO, é necessário atender a alguns pré-requisitos. Primeiramente, o usuário deve dispor do modelo de simulação construído, verificado e validado. Além do modelo de simulação, é importante a definição do modelo de otimização, com o delineamento das variáveis de decisão e seus limites, a função objetivo e restrições do problema. O usuário também deve ponderar sobre o seu *budget* de simulação, ou seja, quantos experimentos este está disposto a realizar durante a otimização.

Além disto, é importante ressaltar que nesta pesquisa o *framework* foi implementado em *Python*, todavia o usuário pode optar por implementá-lo em outra linguagem de programação

de sua preferência. O *software* de simulação adotado deve permitir a comunicação com o ambiente de programação do *framework* (que neste caso foi o *Python*), possibilitando a alteração dos valores das variáveis de decisão, a leitura dos resultados da simulação e o envio de comandos de controle, como executar e parar a simulação. FlexSim® é um exemplo de *software* de simulação compatível com essas necessidades.

Além disso, é imprescindível que os usuários possuam conhecimento básico de programação, bem como familiaridade com técnicas de metamodelagem e meta-heurísticas. Para garantir a eficiência das execuções, recomenda-se o uso de um computador com capacidade de processamento adequada, incluindo um processador *multi-core* e memória suficiente. O *framework* AMSO proporciona uma solução inovadora e eficiente para problemas de otimização complexos em SED, contribuindo significativamente para avanços tanto na pesquisa quanto na aplicação prática em diversos setores industriais.

5.2. Apresentação do *Framework*

Esta seção descreve o *framework* proposto nesta tese: *Adaptive Metamodeling-based Simulation Optimization*. O desenvolvimento do AMSO teve como base os resultados da RSL (Capítulo 4), e buscou atender a todas as etapas classificadas na Seção 4.6.2.4 como fundamentais e recomendadas (i.e., Amostragem Inicial, Avaliação na Simulação, Padronização dos Dados, OHP, Treinamento do Metamodelo, Validação do Metamodelo, Otimização da FA e Critério de Parada) convergindo em uma abordagem fundamentada nas melhores práticas da literatura de MA.

O AMSO tem como base cinco técnicas principais: *Latin Hypercube Design*, otimização de hiperparâmetros via Algoritmo Genético, *Bagging-based Gradient Boosted Trees* e Algoritmo Genético para otimização da função de aquisição (*Balanced Expected Improvement*). A Figura 16 apresenta as etapas do *framework* proposto.

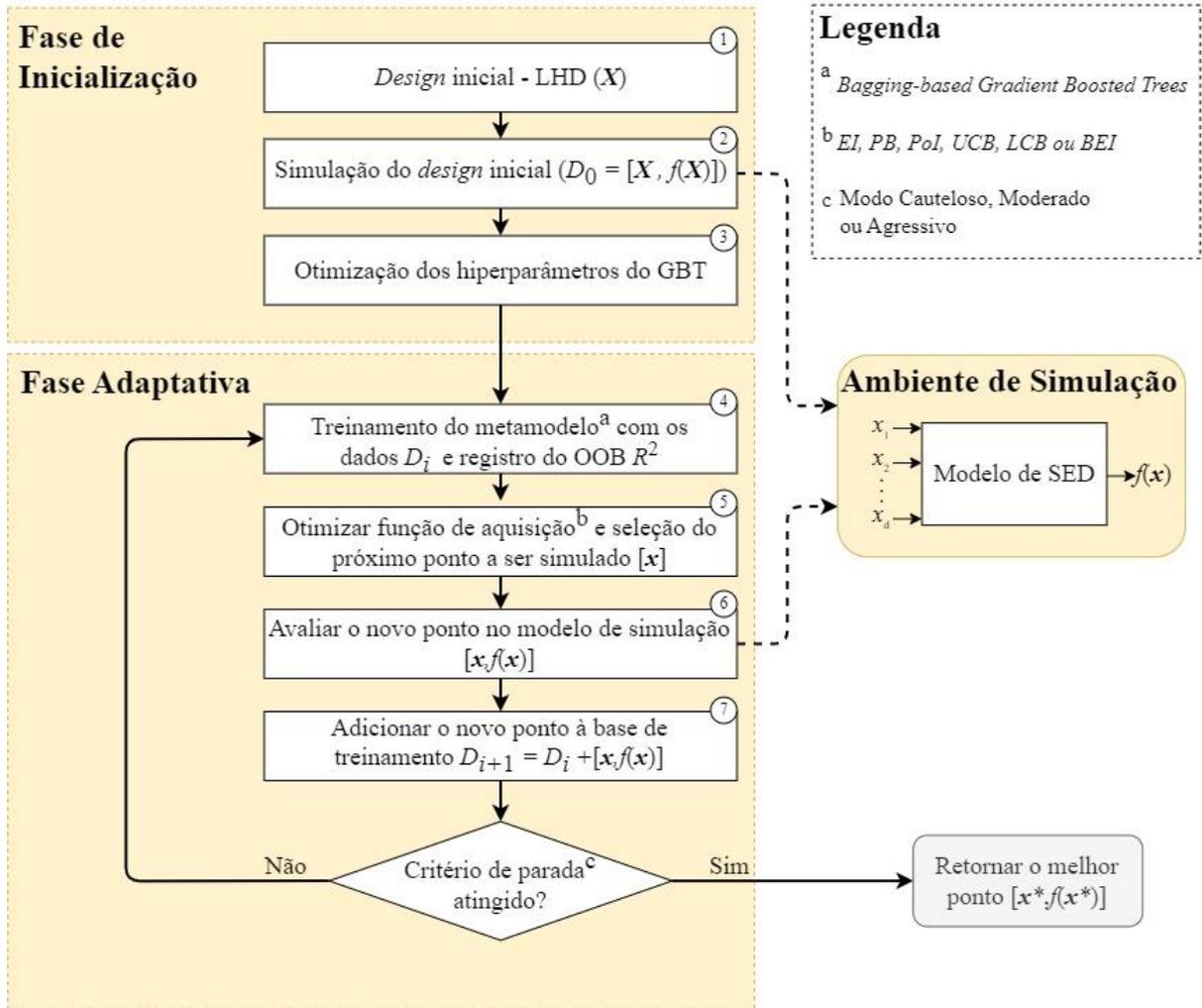


Figura 16: *Framework Adaptive Metamodeling-based Simulation Optimization*

Este *framework* contempla duas macro fases, a fase de inicialização e fase adaptativa.

5.2.1. Fase de Inicialização

5.2.1.1. Etapa 1: *Design Inicial*

A Etapa 1 da fase de inicialização compreende a geração do arranjo inicial (X_0) para treinamento do metamodelo. Para isto, o arranjo é delineado pela técnica de DoE conhecida como *Latin Hypercube Design*. O LHD é o método de amostragem estratificada mais popular em estudos de metamodelagem (veja Seção 4.6.2.2). O tamanho amostral sugerido para o arranjo LHD inicial é de $2d$ a $10d$, sendo d a dimensão do problema (Bharaj *et al.*, 2015; Wang; Ierapetritou, 2017). Este é um parâmetro a cargo do usuário, que deve avaliar a disponibilidade de recursos (número de experimentos) para alocar entre a fase de inicialização e adaptativa.

5.2.1.2. Etapa 2: Simulação do *Design* Inicial

Na Etapa 2, cada solução \mathbf{x} (com $\mathbf{x} \in \mathbf{X}_0$) do arranjo experimental é avaliada no modelo de simulação (com r replicações). Conforme destacado na Seção 2.1, o uso de replicações, embora demande mais tempo computacional, é essencial em estudos de SED e outros tipos de simulações sujeitas a estocasticidade. No *framework* proposto, a simulação de cada replicação r é realizada utilizando o conceito de paralelismo, aproveitando os benefícios de máquinas *multi-core* para reduzir o tempo total do processo de otimização. Desta forma, a Etapa 2 é responsável por gerar o vetor de respostas $f(\mathbf{X}_0)$ para o arranjo inicial, formando a base de treinamento inicial do metamodelo, $\mathbf{D}_0 = [\mathbf{X}_0, f(\mathbf{X}_0)]$.

5.2.1.3. Etapa 3: Otimização dos Hiperparâmetros

A Etapa 3 otimiza os hiperparâmetros do algoritmo GBT por meio do AG e *k-fold cross-validation*, visando definir um modelo *base-learner* mais adequado à base experimental do problema. Esta tese optou pelo uso do AG devido a sua relevância na literatura da área, assim como ressaltado na Seção 4.6.2.3. Os hiperparâmetros do GBT adotados e sua amplitude foram: número de estimadores (1 – 100); taxa de aprendizagem (0,05 – 0,30); e amostras mínimas por folha (0,12 – 0,50) (Amaral et al., 2022b). Como não é objetivo desta tese explorar o AG, seus parâmetros foram configurados de acordo com o padrão da biblioteca *sklearn_genetic* do *Python* (conforme delimitação da pesquisa apresentada na Seção 1.5).

Vale ainda destacar que a OHP é realizada por meio de um processamento paralelo, reduzindo os impactos negativos desta etapa no tempo total. Ressalta-se que esta etapa de OHP é realizada apenas na inicialização do *framework*, utilizando a base experimental inicial normalizada $\mathbf{D}_0 = [\mathbf{Z}_0, f(\mathbf{X}_0)]$, na qual \mathbf{Z}_0 representa os valores normalizados de \mathbf{X}_0 (média 0 e desvio padrão 1).

Por fim, é importante ressaltar o uso do GBT como *base-learner*, pois na literatura existem diversos métodos de ML que podem ser utilizados para construção de metamodelos, assim como discutido na Seção 4.6.2.2. A opção pelo uso do GBT foi baseada em estudos prévios realizados e publicados pelo autor (Amaral et al., 2022d, 2022a), os quais compararam os principais métodos de ML para metamodelagem em problemas típicos de SED e concluíram sobre a superioridade do GBT para este tipo de aplicação. Vale destacar que na literatura de ML existem outros algoritmos com grande capacidade de aprendizado, como é o caso do *Deep Learning*, todavia requerem uma grande base de dados para seu treinamento apropriado.

5.2.2. Fase Adaptativa

Esta fase é responsável pelo aspecto adaptativo do *framework*. Ele é adaptativo pois a cada iteração, com base no critério de busca definido, novos pontos são adicionados a base de treinamento e o metamodelo é retreinado, adquirindo maior precisão nas regiões mais promissoras do espaço de soluções, e desta forma, adaptando-se ao problema.

5.2.2.1. Etapa 4: Treinamento do Metamodelo

Na Etapa 4 inicia-se a fase adaptativa do método e compreende o treinamento do metamodelo. Nesta etapa, o algoritmo *Bagging* com GBT otimizado (*base-learner*) é treinado recursivamente com a base de treino \mathbf{D}_i de tamanho ξ , no qual é acrescido de um novo experimento a cada iteração i do algoritmo, $\xi_{i+1} = \xi_i + 1$. Assim como na Etapa 3, a base experimental utilizada para o treinamento do metamodelo é normalizada para média 0 e desvio padrão 1. Após o treinamento, o metamodelo passa por uma validação, na qual é calculado o R^2 (pelo método de OOB) do metamodelo B-GBT, valor que será utilizado na próxima etapa. Mais detalhes sobre o algoritmo *Bagging* e GBT podem ser encontrados nas Seções 2.3.1 e 2.3.2.

5.2.2.2. Etapa 5: Otimizar Função de Aquisição

A Etapa 5 é relacionada ao sorteio do critério de busca entre *Exploration* e *Exploitation*. *Exploration-Exploitation* é um *trade-off* fundamental que surge na tomada de decisões quando se enfrenta situações incertas ou desconhecidas. Refere-se à escolha entre explorar novas opções ou aproveitar as opções conhecidas. *Exploration* envolve a busca por novas opções, a coleta de informações e a experimentação de diferentes alternativas, ou seja, regiões nas quais o erro de previsão do metamodelo é alto. Essa abordagem é útil quando o ambiente é imprevisível e há a necessidade de aprender sobre novas regiões que podem conter o ótimo, contribuindo para melhoria do ajuste do metamodelo. *Exploitation*, por outro lado, envolve maximizar o conhecimento atual escolhendo a melhor opção em regiões conhecidas do espaço de soluções, ou seja, melhor valor de $\hat{f}(\mathbf{x})$.

Nesta etapa o objetivo é a definição do valor da variável binária λ , que assume o valor 0 para *Exploration* e 1 para *Exploitation*. Considerando ρ um número aleatório distribuído uniformemente, tal que $\rho \sim \text{Uniform}(l, u)$, sendo l e u o limite inferior e superior para o sorteio. Os parâmetros l e u podem ser definidos pelo usuário e representam, respectivamente, o R^2 mínimo desejável para iniciar uma busca de *Exploitation* e o valor máximo para permitir a *Exploration*. Neste trabalho o valor de l foi definido como 0,85, e u como 1,00, considerando

que um metamodelo com $R^2 = 0,85$ é considerado um valor de ajuste minimamente aceitável (Østergård; Jensen; Maagaard, 2018) e $R^2 = 1,00$ representa um metamodelo 100% aderente à base experimental.

O valor de λ é selecionado com base no valor sorteado para ρ e o R^2 calculado na etapa anterior, de modo que quanto maior o valor R^2 maior será a probabilidade de *Exploitation* ($\lambda = 1$), segundo a Eq. (13).

$$\lambda = \begin{cases} 1, & \text{Se } \rho < R^2 \\ 0, & \text{Caso contrário} \end{cases} \quad (13)$$

A Etapa 6 compreende a otimização da função de aquisição, cujo objetivo é selecionar o melhor ponto a ser simulado e adicionado à base de treino na próxima iteração do algoritmo. A FA proposta neste trabalho, denominada *Balanced Expected Improvement* (BEI), representa uma associação entre as duas FAs mais utilizadas nos artigos analisados pela RSL: EI e PB. A função BEI representa a *trade-off* entre o termo de busca global (*Exploration*) da função EI, $\sigma\phi\left(\frac{f_{min}-\hat{f}(x)}{\sigma}\right)$, com a busca local proporcionada pela função PB, $\hat{f}(x)$. A função BEI proposta nesta tese é definida pela Eq. (14).

$$BEI(x) = (1 - \lambda)\sigma\phi\left(\frac{f_{min} - \hat{f}(x)}{\sigma}\right) + \lambda\hat{f}(x) \quad (14)$$

Na Eq. (14), ϕ representa a função densidade de probabilidade, f_{min} é o menor valor observado na base de treino, $\hat{f}(x)$ e σ são a previsão e o desvio padrão da previsão do metamodelo para o ponto x , definidos pelas Equações (3) e (4) (Seção 2.3.1), respectivamente. O primeiro termo da Eq. (14) retrata o componente de *Exploration*, e o segundo termo representa o componente de *Exploitation*. Percebe-se que o valor de λ encontrado na etapa anterior é o critério que define se a função de aquisição optará por *Exploration* ou *Exploitation*.

Vale ressaltar, que além da função BEI, este *framework* possibilita a utilização de outras cinco FAs, a saber: EI, PB, PoI, UCB e LCB, que foram as funções mais utilizadas na literatura, segundo a RSL apresentada no Capítulo 4. Mais detalhes sobre estas funções e suas formulações podem ser encontradas na Seção 4.6.2.3. O objetivo desta Etapa é encontrar o ponto x que maximiza o valor de função de aquisição selecionada, e para atingir este objetivo, o AG é utilizado como mecanismo de busca. Como parâmetros do AG, utilizou-se o padrão da biblioteca *geneticalgorithm2*.

5.2.2.3. Etapa 6: Avaliar Solução no Modelo de Simulação

Na Etapa 6, a melhor solução encontrada pelo AG é avaliada no modelo de simulação, retornando o valor de $f(x)$ para este ponto. Conforme sinalizado na Seção 1.5, podem ser

utilizados diversos *softwares* de simulação para construção e simulação do modelo, desde que este provenha comunicação direta com o ambiente *Python* e possa ser controlado externamente por este (modificação dos valores das variáveis de decisão, leitura dos resultados da simulação, comandos de execução e finalização da simulação).

5.2.2.4. Etapa 7: Incremento da base de treinamento

Na Etapa 7, o novo ponto, definido na Etapa 5 e simulado na Etapa 6, é adicionado a base de treino do metamodelo, $D_{i+1} = D_i + [\mathbf{x}, f(\mathbf{x})]$, e as Etapas 4 – 8 são repetidas até que o critério de parada seja atingido.

5.2.2.5. Critério de Parada

Nesta tese, foram definidos três critérios de parada possíveis, a serem definidos pelo usuário, a saber:

- Modo Cauteloso: O critério de parada é satisfeito apenas quando o número máximo de iterações (ou simulações a serem realizadas na fase iterativa), E_{max} , é atingido. Este número máximo de iterações é um parâmetro definido pelo usuário e reflete o *budget* de tempo e simulações envolvidas em cada projeto. Vale ressaltar que este critério foi adotado por ser um critério tangível para o gestor, diretamente ligado ao propósito de aplicação do AMSO (quando a quantidade de simulações possíveis de serem executadas é um limitante) e por ser o critério de parada mais usado na literatura (vide Seção 4.6.2.4).
- Modo Moderado: O critério de parada é satisfeito se o número de iterações sem melhoria na função objetivo, τ , for igual ou maior que 25% do número máximo de iterações definido pelo usuário, ou seja, $\tau \geq 0,25 \times E_{max}$.
- Modo Agressivo: O critério de parada é satisfeito se o número de iterações sem melhoria na função objetivo, τ , for igual ou maior que 10% do número máximo de iterações definido pelo usuário, ou seja, $\tau \geq 0,1 \times E_{max}$.

Quando o critério de parada é atingido, o algoritmo é encerrado e a solução \mathbf{x}^* com melhor valor de $f(\mathbf{x}^*)$ é retornada como solução do problema.

5.3. Implementação e Validação do *Framework*

O *framework* AMSO foi implementado na linguagem de programação *Python* (versão 3.0) em uma estrutura que permite controle direto do ambiente de SED e executa as simulações em paralelo. O *software* de simulação adotado neste trabalho foi o *FlexSim*[®] (versão 22.2.3), no qual os modelos de SED foram desenvolvidos.

A eficácia do *framework* proposto será avaliada por meio de sua aplicação para otimização de dois objetos de estudo reais, representando problemas de alocação de recursos na indústria. O primeiro estudo de caso apresenta um modelo que tem como objetivo a alocação semanal de recursos produtivos em uma manufatura têxtil. O segundo objeto de estudo apresenta um problema de expansão de capacidades em uma mina integrada de minério de ferro, compreendendo processos desde a extração e estocagem, até o processamento do minério (usina de classificação) e expedição.

A fim de validar o *framework* proposto, o método EGO, implementado em *Python* pelo *Surrogate Modeling Toolbox* (Bouhlef *et al.*, 2019), foi selecionado como *benchmarking* por ser um dos *frameworks* de otimização baseados em metamodelagem mais utilizados da literatura. Além disso, AMSO e EGO estão na mesma classe de algoritmos OvS (baseados em metamodelagem adaptativa), permitindo uma comparação direta. Além disso, segundo Sousa Junior *et al.* (2019a), algoritmos evolutivos são os mais utilizados em problemas de otimização de simulação a eventos discretos. Portanto, o AG, implementado em *Python* pelo pacote *Geneticalgorithm2* (Pascal; Solgi, 2023), também foi utilizado como *benchmarking*.

6. APLICAÇÃO DO *FRAMEWORK* PROPOSTO

Este Capítulo avalia a eficácia do *framework* AMSO ao aplicá-lo para otimização de três problemas reais de alocação de recursos na indústria.

6.1. Objeto de Estudo 01: Manufatura Têxtil

O Objeto de Estudo 01 (OE1) representa uma minifábrica de uma manufatura do setor de *fast-fashion*, cuja principal característica é a rápida mudança na demanda e mix de produtos, tempo de ciclo curto, alta volatilidade, baixa previsibilidade e alto nível de compra impulsiva e competição de preço.

O modelo de SED foi desenvolvido para representar os fluxos produtivos, restrições operacionais e de matéria prima, variabilidades dos processos, quebra de equipamentos, entre outras peculiaridades do sistema. O fluxo de produção é segmentado em oito postos de trabalho (Processo B ao Processo I), além das áreas de Recepção (A) e Expedição (J). A operação é em grande parte manual, embora parte do transporte interno de produtos seja automatizado por meio de um veículo guiado por automação. É relevante destacar que, com exceção do Processo B, que envolve uma tarefa específica, a linha conta com operadores multifuncionais. Nesse contexto, os operadores podem ser divididos em dois grupos: o Grupo I, composto por até dez operadores, responsáveis pelos processos C, D, E e G; e o Grupo II, também formado por até dez operadores, encarregados dos demais processos (F, H, I e J).

A empresa em questão trabalha com previsões de demanda semanal, ao qual na sexta-feira o gestor recebe a demanda da próxima semana e deve decidir sobre a alocação dos recursos produtivos (maquinário) e equipe para atender tal demanda e potencializar os lucros da minifábrica. O gestor tem a sua disposição a possibilidade de locar equipamentos e funcionários de outras minifábricas da empresa, porém incorrendo em um custo operacional adicional ao seu processo. Mais detalhes sobre este case podem ser consultados em Dos Santos *et al.* (2021). A visualização 3D do modelo é apresentada na Figura 17.



Figura 17: Modelo computacional (OE1)

O intuito do modelo é prover auxílio a esta tomada de decisão, que por meio da OvS, busca encontrar a alocação de recursos que otimiza o lucro da minifábrica. Assim como apresentado na Tabela 4, as variáveis de decisão, $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8]^T$, são o tamanho da equipe dedicada ao grupo de tarefas tipo 1 e tipo 2 (x_1 e x_2), e a quantidade de equipamentos de diferentes tipos alocados ao processo (x_3 a x_8).

Tabela 4: Variáveis de decisão e limites (OE1)

Variáveis	Descrição	Limite Inf. (\mathbf{l})	Limite Sup. (\mathbf{u})	Custo (\mathbf{c})
x_1	Operadores GrupoA	2	10	R\$ 800,00
x_2	Operadores GrupoB	2	10	R\$ 800,00
x_3	Máquinas Reta	1	3	R\$ 500,00
x_4	Máquina Overlook	2	10	R\$ 500,00
x_5	Máquina Debrum	2	10	R\$ 500,00
x_6	Estação de Arremate	2	10	R\$ 300,00
x_7	Estação de Bainha	2	10	R\$ 350,00
x_8	Estação de Passadoria	2	10	R\$ 300,00

Considerando os limites inferiores, $\mathbf{l} = [2 \ 2 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2]^T$, e superiores, $\mathbf{u} = [10 \ 10 \ 3 \ 10 \ 10 \ 10 \ 10 \ 10]^T$, das variáveis de decisão sujeitas ao custo unitário (R\$) $\mathbf{c} = [800 \ 800 \ 500 \ 500 \ 500 \ 300 \ 350 \ 300]^T$, a função lucro é dado pela Eq. (15).

$$L(\mathbf{x}) = \psi v - \sum_{i=1}^8 x_i c_i \quad (15)$$

Na Eq. (15), $L(\mathbf{x})$ representa o lucro, ψ é a produção semanal alcançada pelo modelo e v é a margem de contribuição média dos produtos. A função objetivo compreende a maximização da expectativa para a função lucro, $\max f(\mathbf{x}) = \mathbb{E}_\omega\{\tilde{\phi}[L(\mathbf{x}, \omega)]\}$, que pode ser obtida pela média de $L(\mathbf{x})$ das replicações do modelo de simulação ($\tilde{\phi}$), sujeitas a sementes aleatórias ω distintas. Para este modelo foram utilizadas 15 replicações (quantidade mínima de replicações considerando a variabilidade do modelo). O modelo matemático que representa o OE1 é dado pela maximização da função Eq.(16), sujeito as restrições (Eq.(17) e Eq.(18)) referentes ao espaço de busca.

$$\max f(\mathbf{x}) = \mathbb{E}_\omega\{\tilde{\phi}[L(\mathbf{x}, \omega)]\} \quad (16)$$

Sujeito a:

$$l_i \leq x_i \leq u_i \quad (17)$$

$$x_i \in \mathbb{Z}, \quad \forall i \in \{1, \dots, 8\} \quad (18)$$

Vale ressaltar que o valor da FO é obtido apartir da produção encontrada (e conseqüentemente o valor de $L(\mathbf{x})$) pelo modelo de SED (representado por $\tilde{\phi}$), que compreende diversas restrições operacionais, estocasticidade, interdependências e complexidades não esboçadas no modelo matemático.

A fim de avaliar sua efetividade, o *framework* AMSO proposto neste trabalho foi aplicado e comparado ao EGO (*Surrogate Modeling Toolbox*) para otimizar este problema de OvS, dado um *budget* limitado de simulações. Para fins de comparação, em ambos os *frameworks* (AMSO e EGO), o critério de parada foi definido como o número máximo de simulações, sendo 50% destinado a geração da base de treinamento inicial e o restante alocado na fase adaptativa dos *frameworks*. No que tange a FA, a função BEI foi utilizada para o AMSO e a função EI foi escolhida para o EGO, já que este é o parâmetro *default* deste método.

Para esta avaliação foram realizados dois testes, o primeiro com um total de 80 experimentos e o segundo com 160 experimentos. Além disto, dado o caráter estocástico destes métodos, o processo de otimização foi repetido 10 vezes para cada método, de modo a avaliar as respostas e a robustez alcançada pelos algoritmos. As Seções 6.1.1 e 6.1.2 apresentam os resultados pelos algoritmos em cada *budget*.

Além desta comparação entre o AMSO e EGO, a Seção 6.1.3 apresenta a aplicação do método AG como *benchmark* e sua comparação com o AMSO para o problema de OvS apresentado. Na sequência, a Seção 6.1.4. apresenta um comparativo entre as funções de aquisição implementadas no *framework*, a fim de avaliar sua sensibilidade a este importante parâmetro e avaliar se a função BEI proposta é competitiva àquelas estabelecidas na literatura. Por fim, a Seção 6.1.5 demonstra os resultados da aplicação do método com os três diferentes critérios de parada disponíveis. Para fins demonstrativos e de melhor entendimento do leitor, o Anexo A deste documento apresenta a aplicação passo a passo do *framework* AMSO para otimização do OE1.

6.1.1. Resultados OE1: AMSO vs. EGO com 80 experimentos

Esta seção apresenta os resultados otimização do problema OE1 considerando um *budget* de simulação de 80 experimentos. A Figura 18 (a) apresenta a média e barra de erro (considerando as 10 repetições) para o gráfico de convergência obtido com o EGO. Ao final dos 80 experimentos, o EGO obteve um resultado médio de R\$ 47.827,07 e desvio padrão de R\$ 936,42, atingindo resultados que variaram entre R\$ 46.900,4 e R\$ 49.512,0 de lucro na pior e melhor repetição do algoritmo, respectivamente. A melhor solução encontrada foi $\mathbf{x}^* = [4\ 6\ 3\ 2\ 2\ 6\ 6\ 4]$, com um lucro de R\$ 49.512,0.

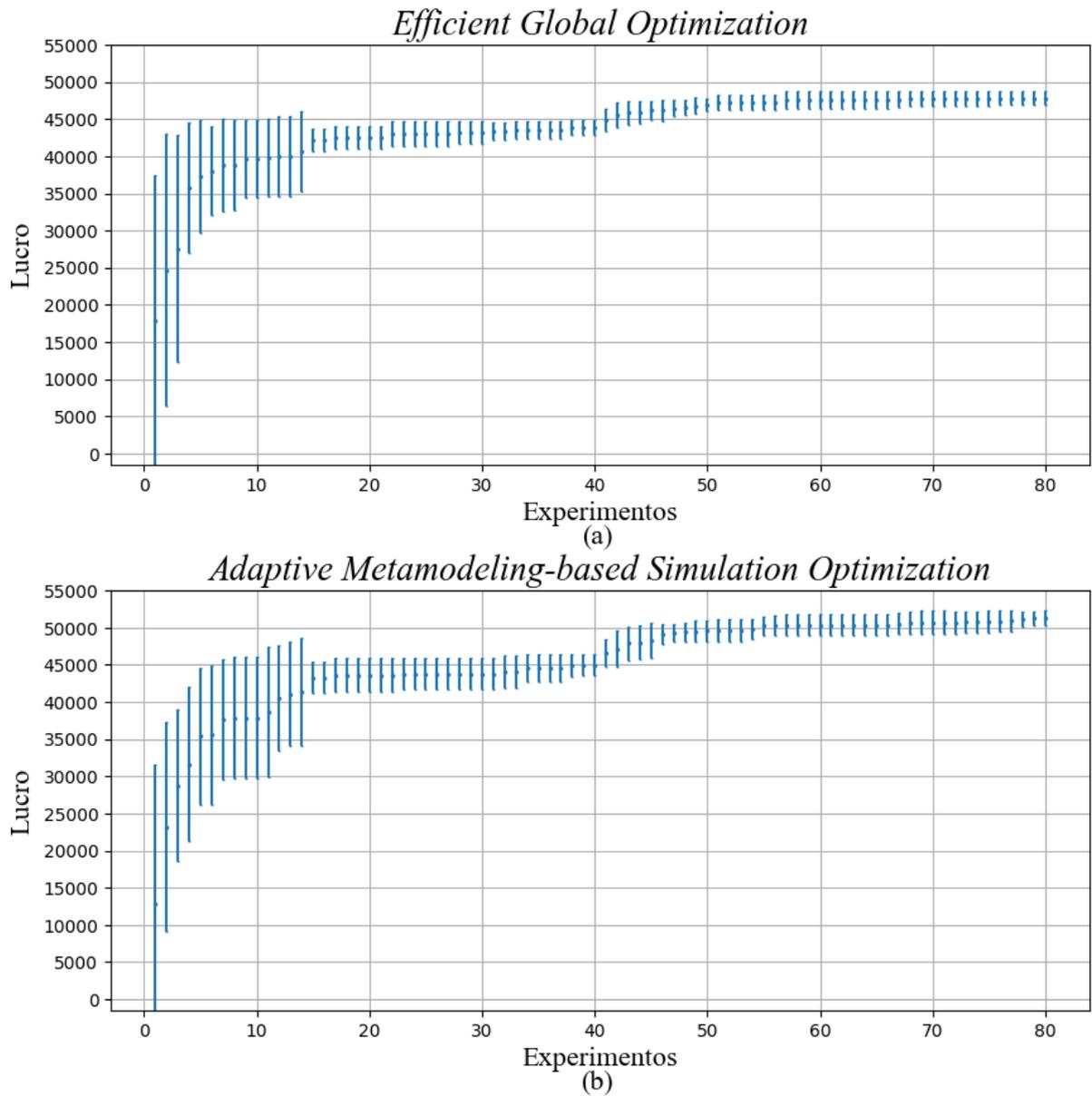


Figura 18: Gráfico de convergência para EGO (a) e o AMSO (b) com 80 experimentos (OE1)

A Figura 18 (b) apresenta os resultados obtidos com *framework* proposto também considerando o *budget* de 80 experimentos. Em média o método proposto alcançou um lucro de R\$ 51.590,00 e desvio padrão de R\$ 891,44, com resultados variando entre R\$ 50.148,80 e R\$ 52.750,40, assim como apresentado na Figura 19 (a). A melhor solução encontrada pelo AMSO foi $\mathbf{x}^* = [4 \ 6 \ 3 \ 2 \ 2 \ 2 \ 2]$.

Comparando os resultados pelo teste estatístico para medianas de Mann-Whitney, constata-se que o *framework* proposto apresenta resultados significativamente superiores ao EGO ($p\text{-value} < 0,001$), com diferença de R\$ 3.763,00 na mediana e R\$ 4.311,00 na média. Com relação ao tempo demandado para execução da otimização, não houve diferença

significativa entre ambos os métodos, sendo que o EGO demandou em média 1.554 segundos e o método proposto demandou 1.535 segundos, assim como destacado na Figura 19 (b).

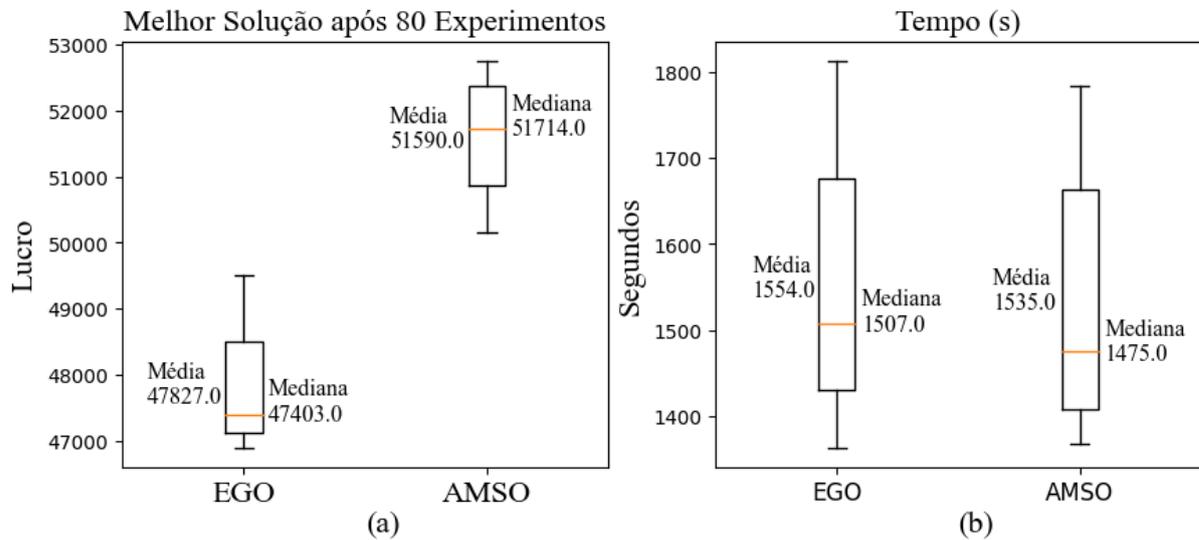


Figura 19: *Boxplot* para valor do lucro atingido após 80 experimentos (a) e tempo gasto na otimização (b) (OE1)

6.1.2. Resultados OE1: AMSO vs. EGO com 160 experimentos

Esta Seção discute os resultados obtidos com a otimização pelo EGO e o AMSO considerando *budget* de simulação de 160 experimentos (com 15 réplicas cada). A Figura 20 apresenta o gráfico de convergência para ambos os *frameworks* (com 10 repetições).

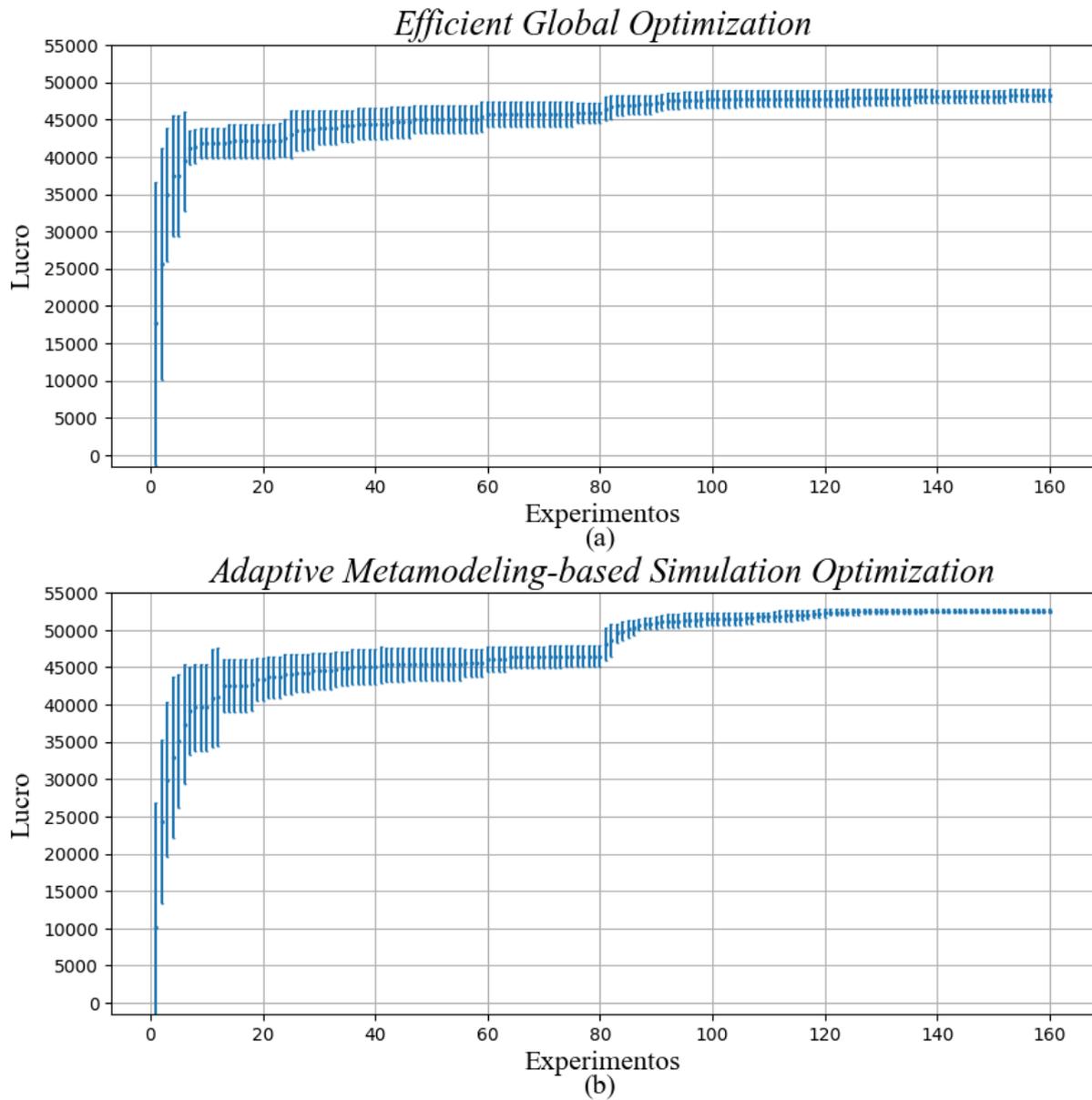


Figura 20: Gráfico de convergência para EGO (a) e o AMSO (b) com 160 experimentos (OE1)

Considerando o novo *budget* de experimentos como critério de parada, o EGO atingiu, em média, um lucro de R\$ 48.262,04, variando entre 47.148,80 e 49.017,60 reais (Figura 20 (a) e Figura 21 (a)), demandando um tempo médio de 3.351 segundos (Figura 21 (b)). A melhor solução encontrada pelo EGO, dentro das 10 repetições, foi $\mathbf{x}^* = [6 \ 5 \ 3 \ 3 \ 2 \ 6 \ 4 \ 4]$. Por outro lado, o *framework* proposto neste trabalho obteve um lucro médio de R\$ 52.558,13, variando entre 51.955,20 e 52.750,40 reais (Figura 20 (b) e Figura 21 (a)), demandando um tempo médio de 3.374 segundos. A melhor solução encontrada pelo AMSO foi $\mathbf{x}^* = [4 \ 6 \ 3 \ 2 \ 2 \ 2 \ 2 \ 2]$.

Nota-se que o *framework* AMSO atingiu um resultado significativamente superior ao EGO, comprovado pelo teste de Mann-Whitney ($p\text{-value} < 0,001$), com uma diferença de R\$

3.903,64 para mediana e R\$ 4.296,09 para média. Observa-se ainda que, durante as 10 repetições do algoritmo, o AMSO apresentou uma menor dispersão dos resultados atingidos, demonstrando maior robustez do *framework* proposto. Destaca-se ainda que a pior solução encontrada pelo AMSO (R\$ 51.955,20) supera o melhor valor encontrado pelo EGO (R\$ 49.017,60), ressaltando sua melhor performance.

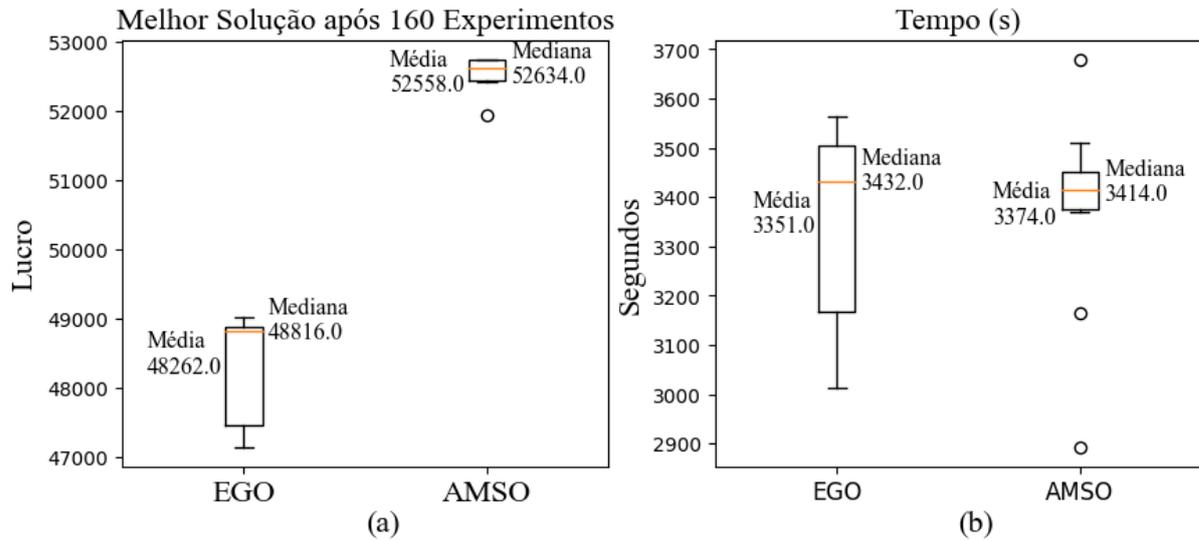


Figura 21: *Boxplot* para valor do lucro atingido após 160 experimentos (a) e tempo gasto na otimização (b) (OE1)

6.1.3. Resultados OE1: AMSO vs. AG

O segundo *benchmark* empregado foi o AG, implementado a partir do pacote *Python geneticalgorithm2* (Pascal; Solgi, 2023) do *PyPI*. Considerando que o AG não trabalha com um número pequeno de avaliações de funções (experimentos), este teste não o limitou ao mesmo *budget* aplicado ao AMSO. O AG foi definido com um número considerável de gerações para analisar se e quando este método supera os resultados obtidos pelo AMSO. Os parâmetros do AG foram definidos como 20 indivíduos para o tamanho da população e o número máximo de iterações igual a 100. Os demais parâmetros foram configurados conforme o padrão do pacote *Python geneticalgorithm2*. A Figura 22 apresenta a média e o erro do gráfico de convergência do AG (10 repetições).

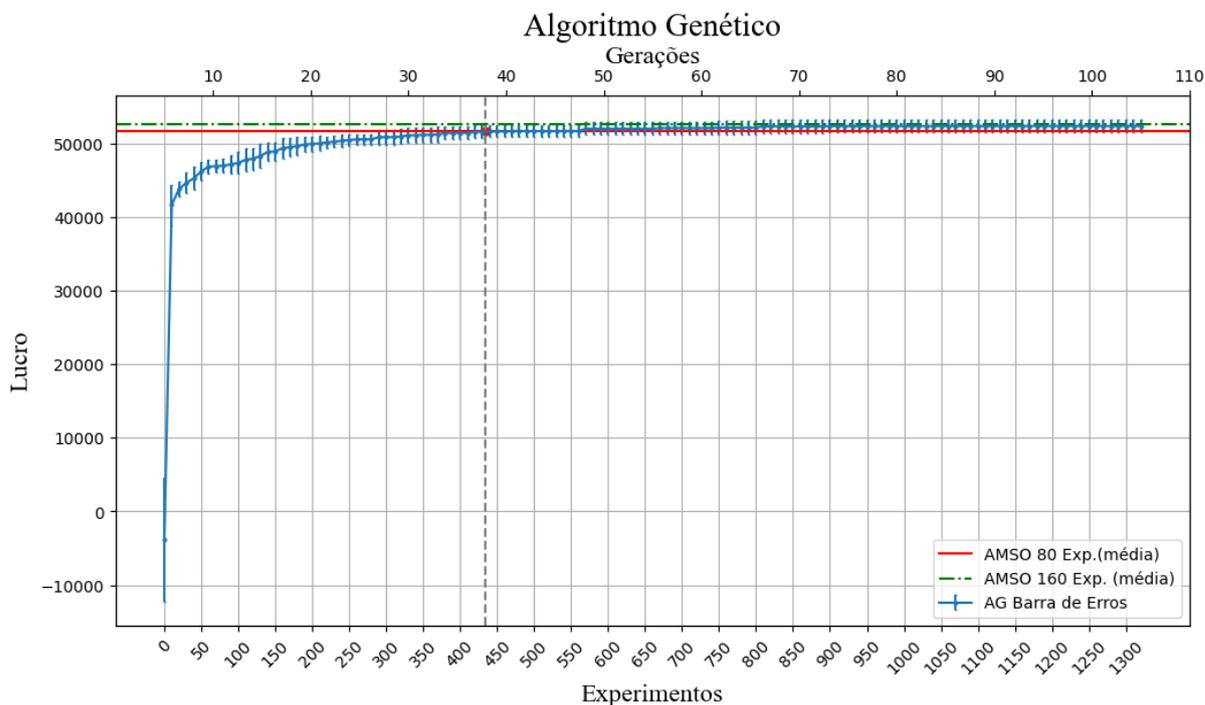


Figura 22: Gráfico de convergência do AG para o OE1

Conforme destacado na Figura 22, o AG demandou, em média, 434 experimentos para superar os resultados do AMSO com 80 experimentos. Além disso, o AG atingiu um lucro médio de R\$ 52.377,62 após 100 gerações (mais de 1.300 experimentos), necessitando em média de 18.653,89 segundos. Pelo teste de Mann-Whitney esse resultado é estatisticamente igual ao AMSO com apenas 160 experimentos (R\$ 52.558,13) com um *p-value* de 0,290. É possível inferir que o AG é um método robusto e com bons resultados, mas requer muitos experimentos.

Ressalta-se que os resultados obtidos com o AMSO se equipararam aos obtidos com o AG, todavia demandando consideravelmente menos experimentos (e tempo), o que mostra a eficiência do *framework* proposto para esta aplicação. Vale ressaltar que não é objetivo analisar a superioridade do AMSO sobre o AG, uma vez que ambos pertencem a classes de algoritmos de otimização distintas e possuem aplicações e características particulares.

6.1.4. Comparativo das Funções de Aquisição para o OE1

Segundo apontado pela RSL, um dos mais importantes parâmetros a serem definidos é a função de aquisição que guiará o processo de seleção dos novos pontos a serem simulados. No *framework* proposto foi implementado seis diferentes funções de aquisição, incluindo a função BEI, EI, PB, PoI, UCB e LCB. Nesta Seção é apresentado um estudo de sensibilidade para este parâmetro nos resultados do AMSO para o OE1 com 80 experimentos (repetidos 10

vezes) a fim de verificar sua robustez quando a função de aquisição utilizada e se a função proposta BEI é competitiva perante as propostas pela literatura. Como é possível observar nos resultados obtidos com cada função de aquisição apresentados na Figura 23, embora a função PB tenha alcançado resultados médios superiores, não é possível afirmar que houve diferença significativa entre as funções de aquisição (p-valor de 0,161 para o teste ANOVA). Além disso, não houve diferença significativa no tempo computacional demandado por cada FA. Logo, conclui-se que, para o OE1, o AMSO foi robusto a este parâmetro e a função BEI foi competitiva perante as demais funções da literatura.

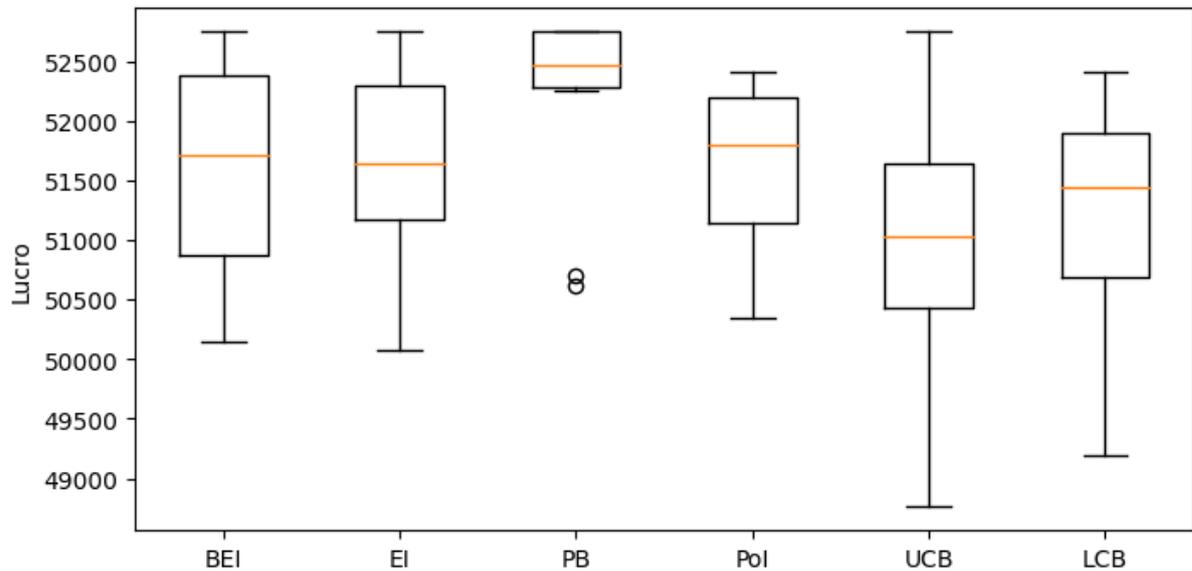


Figura 23: Comparativo entre funções de aquisição para o OE1

6.1.5. Comparação dos Critérios de Parada para o OE1

Esta seção demonstra a aplicação do AMSO considerando os três critérios de parada (aqui denominados modo de operação) disponíveis no *framework*, ou seja, o modo “Agressivo”, “Moderado” e “Cauteloso”. Para esta demonstração foi utilizado a mesma base de treinamento inicial ($\xi = 40$) para os três modos, e definido o número máximo de iterações igual a 40. Como resultado, o modo “Agressivo” parou após 9 iterações e retornou na melhor solução um lucro de R\$ 47.654,80. Já o modo “Moderado” retornou um lucro de R\$ 50.921,00 após 16 iterações. O modo “Cauteloso”, conforme esperado, terminou após 40 iterações retornando um lucro de R\$ 51.894,40. A Figura 24 apresenta o gráfico de convergência para os três modos.

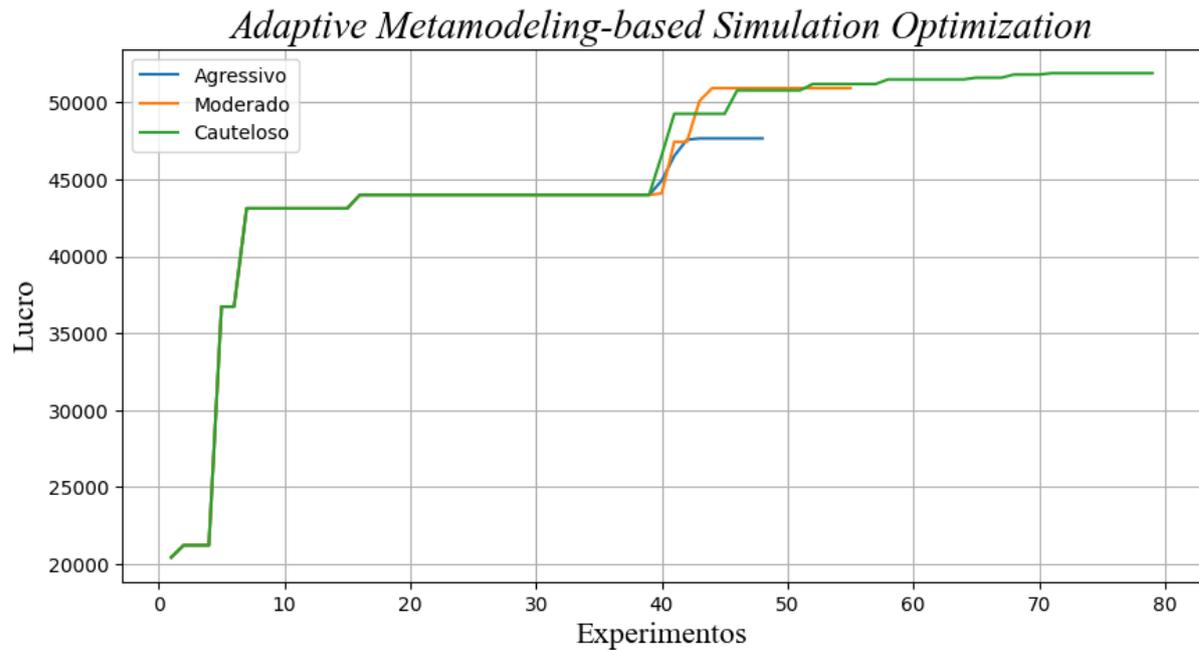


Figura 24: Gráfico de convergência do OE1 para os três critérios de parada

6.2. Objeto de Estudo 02: Mineração

O segundo objeto de estudo (OE2) corresponde a um problema de expansão de capacidades de uma planta de processamento de minério de ferro. O objetivo é decidir o que e quanto expandir das capacidades dos processos para melhorar a produção com o menor custo possível, ou seja, melhorar o lucro adicional da planta com o projeto de expansão. As variáveis de decisão correspondem ao número de caminhões *off-road* (x_1), às capacidades (em 10^3 ton) dos três sistemas de peneiramento granulométrico (x_2, x_3, x_4) e às capacidades (em 10^1 ton) dos sistemas de britagem primário, secundário e terciário (x_5, x_6, x_7). Os limites inferior e superior são, respectivamente, $\mathbf{l} = [5 \ 10 \ 170 \ 10 \ 20 \ 30 \ 30]^T$ e $\mathbf{u} = [15 \ 20 \ 200 \ 20 \ 30 \ 50 \ 50]^T$. Os custos associados anuais (US\$) de cada unidade de expansão de recursos são $\mathbf{c} = [24,97 \ 3,00 \ 3,00 \ 3,00 \ 7,80 \ 0,96 \ 0,96]^T \times 10^5$. A

Tabela 5 apresenta as variáveis de decisão, limites e custos associados.

Tabela 5: Variáveis de decisão e limites (OE2)

Variáveis	Descrição	Limite Inf. (\mathbf{l})	Limite Sup. (\mathbf{u})	Custo (\mathbf{c}) (10^5)
x_1	Nº de caminhões <i>Off Road</i>	5	15	\$ 24,97
x_2	Cap. do Peneiramento Primário (10^3 t)	10	20	\$ 3,00
x_3	Cap. do Peneiramento Secundário (10^3 t)	170	200	\$ 3,00
x_4	Cap. do Peneiramento Terciário (10^3 t)	10	20	\$ 3,00
x_5	Cap. da Britagem Primária (10^1 t)	20	30	\$ 7,80
x_6	Cap. da Britagem Secundária (10^1 t)	30	50	\$ 0,96
x_7	Cap. da Britagem Terciária (10^1 t)	30	50	\$ 0,96

O limite inferior para as variáveis de decisão, \mathbf{l} , representa o recurso atual disponível em operação, com custo base (τ_{base}) de \$ 90.844.943,00 e nível de produção atual (ψ_{base}) de 12.500.000 ton/ano. A função lucro ($L(\mathbf{x})$) é dada pela receita com a produção adicional menos o custo adicional com os novos recursos, conforme formulado pela Eq. (19).

$$L(\mathbf{x}) = v(\psi - \psi_{base}) - \left[\left(\sum_{i=1}^7 x_i c_i \right) - \tau_{base} \right] \quad (19)$$

Na Eq. (19), $L(\mathbf{x})$ é o lucro com a expansão da capacidade, v é a margem de contribuição com a venda do minério de ferro (fixada em US\$ 21,00/ton), e ψ é a produção anual atingida com a solução \mathbf{x} . O objetivo é maximizar a expectativa da função lucro, calculada pela média de 15 replicações do modelo de simulação ($\tilde{\phi}$) com diferentes sementes aleatórias ω , ou seja, $\max f(\mathbf{x}) = \mathbb{E}_{\omega} \{ \tilde{\phi}[L(\mathbf{x}, \omega)] \}$. O modelo matemático que representa o OE2 é dado pela maximização da função objetivo Eq.(20), sujeito as restrições (Eq.(21) e Eq.(22)) referentes ao espaço de busca.

$$\max f(\mathbf{x}) = \mathbb{E}_{\omega} \{ \tilde{\phi}[L(\mathbf{x}, \omega)] \} \quad (20)$$

Sujeito a:

$$\mathbf{l}_i \leq x_i \leq \mathbf{u}_i \quad (21)$$

$$x_i \in \mathbb{Z}, \quad \forall i \in \{1, \dots, 7\} \quad (22)$$

É importante destacar que o valor da FO é derivado do lucro (média das replicações) encontrado pelo modelo de SED, representado por $\tilde{\phi}$. Este modelo abrange diversas restrições operacionais, estocasticidade, interdependências e complexidades que não são delineadas no

modelo matemático. A interface 3D do modelo de SED desenvolvido no *software* de simulação *FlexSim*[®] versão 22.2.3) é apresentado na Figura 25.

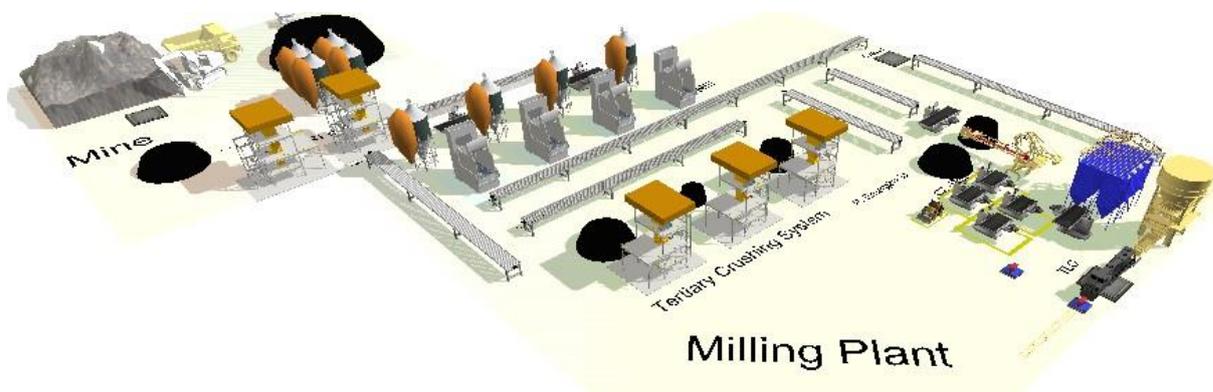


Figura 25: Modelo computacional (OE2)

Considerando que este é um modelo real desenvolvido em um projeto de consultoria, o *budget* de simulação foi definido em conjunto com a equipe do projeto, que considerou que o *framework* deveria retornar um resultado em, no máximo, um turno de trabalho (aproximadamente 9 horas). Dada a complexidade do modelo e tempo necessário para execução de cada simulação (significamente maior que o demandado pelo OE1), para este objeto de estudo foi definido um *budget* de 70 experimentos (10d), o que demanda entre 8 e 9 horas para sua execução.

Assim como apresentado no OE1, a otimização via EGO e AMSO foi repetida 10 vezes, com 70 experimentos cada. Para ambos, foi alocado 50% do *budget* para a geração da base experimental inicial (35 experimentos) e o restante para a fase adaptativa. A Seção 6.2.1 apresenta os resultados obtidos com o AMSO e EGO, a Seção 6.2.2 discute os resultados alcançados pelo segundo método *benchmark*, o AG, e sua comparação com o AMSO. Por fim, assim como apresentado no OE1, a Seção 6.2.3 apresenta e discute os resultados obtidos pelo método AMSO com cada uma das funções de aquisição implementadas.

6.2.1. Resultados OE2: AMSO vs. EGO com 70 experimentos

Esta seção apresenta os resultados de otimização do problema OE2 considerando um *budget* de simulação de 70 experimentos. A Figura 26 apresenta os gráficos de convergência para o EGO e AMSO.

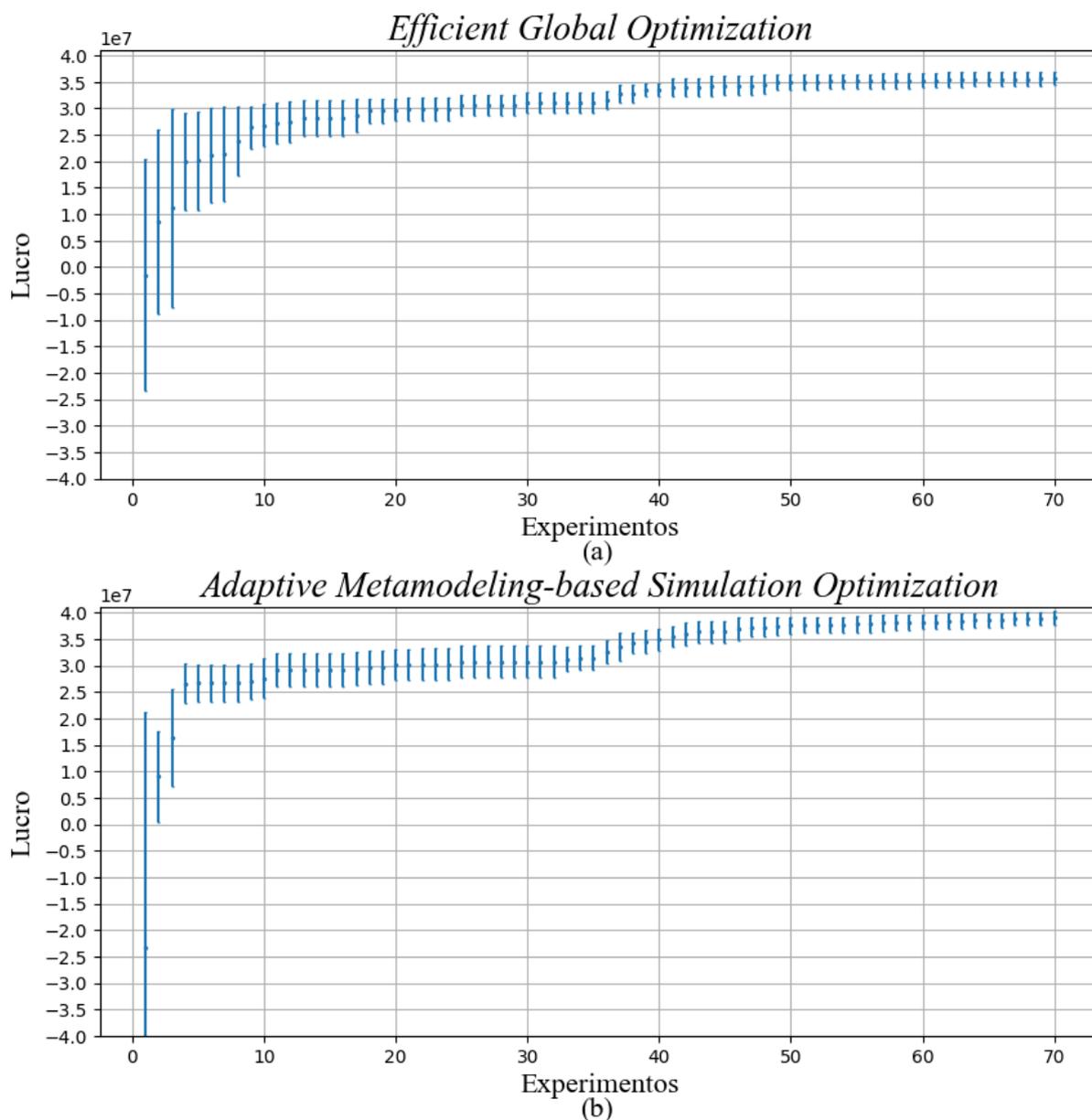


Figura 26: Gráfico de convergência para EGO (a) e o AMSO (b) com 70 experimentos (OE2)

Com um *budget* de simulação de 70 experimentos, a Figura 26 (a) mostra a média e barra de erro do gráfico de convergência obtido pelo EGO. O resultado médio alcançado pelo EGO foi de US\$ 35.655.654,00, com desvio padrão de US\$ 1.161.637,28. Dentro das repetições, a melhor solução encontrada pelo EGO sugere o uso de $\mathbf{x}^* = [11 \ 14 \ 171 \ 12 \ 29 \ 39 \ 33]$ com $f(\mathbf{x}^*) = 37.594.665,00$. Por outro lado, conforme apresentado nas Figura 26 (b), o *framework* proposto obteve um lucro médio de US\$ 39.104.952,80 com um desvio padrão de US\$ 1.254.997,93. A melhor solução encontrada nas repetições foi $\mathbf{x}^* = [11 \ 10 \ 170 \ 11 \ 29 \ 30 \ 32]$ com $f(\mathbf{x}^*) = 40.941.615,00$.

Conforme apresentado na Figura 27 (a), ao comparar os dois *frameworks* pelo teste de Mann-Whitney, observou-se que o AMSO superou o EGO com significância estatística ($p\text{-value} < 0,001$), apresentando diferença na mediana e média de US\$ 3.479.895,00 e US\$ 3.449.298,80, respectivamente. Além disso, conforme observado na Figura 27 (b), não foi observada diferença significativa entre o tempo requerido pelo EGO e AMSO.

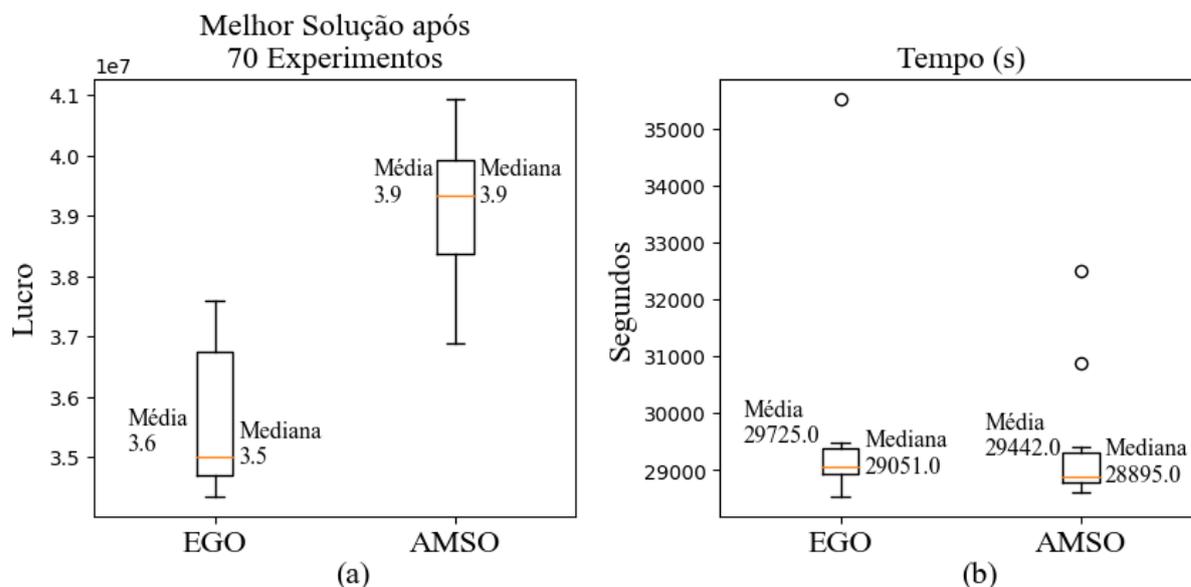


Figura 27: *Boxplot* para valor do lucro atingido após 70 experimentos (a) e tempo gasto na otimização (b) (OE2)

6.2.2. Resultados OE2: AMSO vs. AG

Semelhantemente ao primeiro objeto de estudo, o AG foi empregado como *benchmarking* adicional para avaliar o problema OE2. O AG foi implementado usando o pacote *geneticalgorithm2* (Pascal; Solgi, 2023) em *Python*, utilizando os parâmetros padrão, exceto para um tamanho de população de 20 e o número de gerações definido como 50. Notavelmente, devido à elevada complexidade deste caso e o tempo demandado por simulação, o AG foi executado apenas uma vez. Este ponto contrasta com o primeiro objeto de estudo, no qual múltiplas repetições do algoritmo e mais gerações eram viáveis. Vale ressaltar que o AG não foi repetido 10 vezes devido a restrição de tempo, pois todo este processo demoraria mais de cinco semanas para ser concluído. A Figura 28 mostra o gráfico de convergência do AG (linha azul) e o lucro médio alcançado pela AMSO com 70 experimentos (linha pontilhada vermelha).

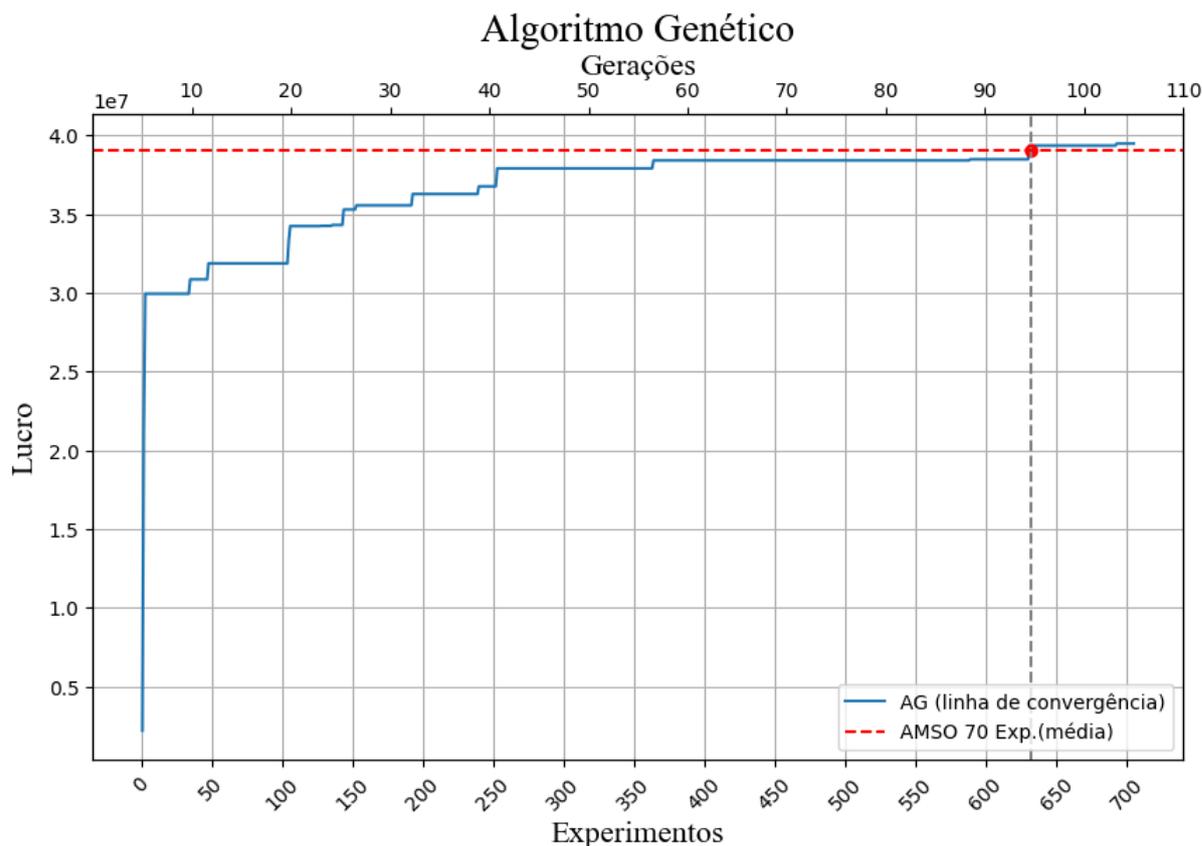


Figura 28: Gráfico de convergência do AG para o OE2

O AG necessitou de 314.464,58 segundos (87,35 horas) para terminar as 50 gerações, retornando $\mathbf{x}^* = [11 \ 12 \ 170 \ 11 \ 29 \ 32 \ 30]$ como melhor solução, com $f(\mathbf{x}^*) = 39.495.735,00$. Em relação ao teste de hipótese *One-Sample Wilcoxon* (Lesik, 2018), esta solução poderia ser considerada estatisticamente igual às encontradas pela AMSO (*p-value* de 0,541); entretanto, utilizando dez vezes mais experimentos que o *framework* proposto. Além disso, ressaltamos que o AG gasta 632 experimentos para superar o lucro médio encontrado pela AMSO com apenas 70 experimentos (US\$ 39.104.952,80).

6.2.3. Comparativo das Funções de Aquisição para o OE2

Nesta seção, é apresentado um estudo de sensibilidade deste parâmetro nos resultados do AMSO para o OE2, utilizando 70 experimentos repetidos 10 vezes. Como apresentado na Figura 29, não é possível observar grandes discrepâncias nos dados. Entretanto, conforme comprovado por um teste ANOVA (*p-value* igual a 0.014), os resultados alcançados pela função BEI foi significativamente maior que aqueles obtidos com as funções UCB e LCB, e não foram significativamente distintos dos obtidos pelo EI, PB e PoI. Além disso, não houve diferença

significativa no tempo computacional demandado por cada FA. Portanto, conclui-se que, para o OE2, o AMSO também apresentou robustez quanto a este parâmetro e a função BEI foi competitiva em relação às demais funções da literatura.

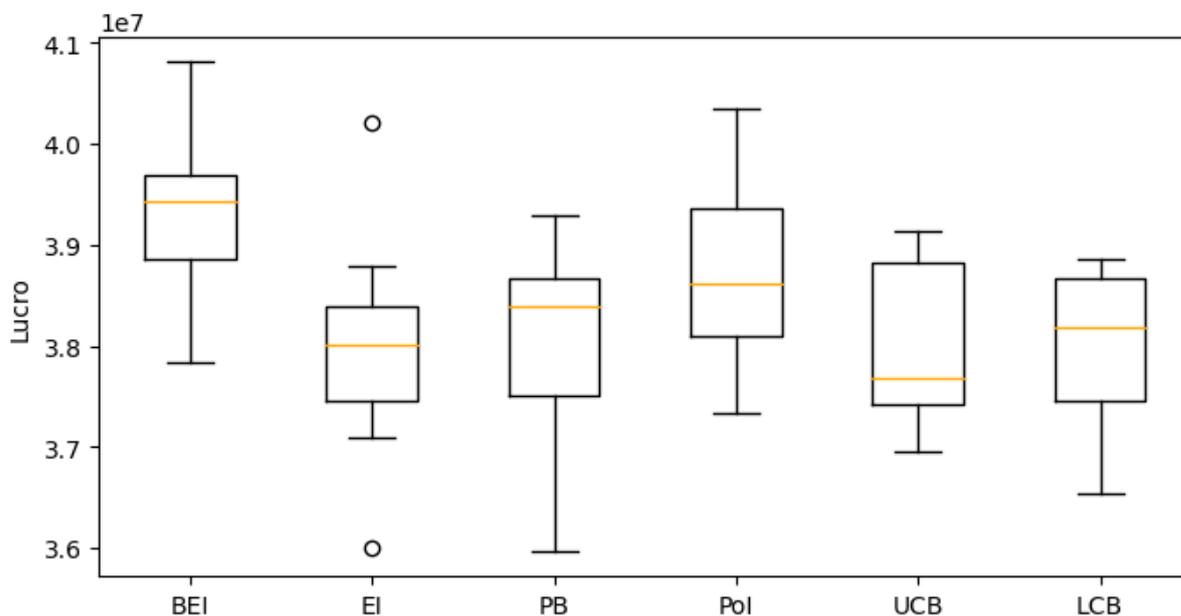


Figura 29: Comparativo entre funções de aquisição para o OE2

6.3. Objeto de Estudo 03: Hospital Temporário

O Objeto de Estudo 03 (OE3) trata-se de um hospital temporário projetado para atender pacientes durante a pandemia de COVID-19. A principal característica deste sistema é a alta variabilidade na demanda de pacientes, que varia em função da gravidade dos sintomas e da taxa de contaminação. O planejamento de recursos hospitalares é desafiador, especialmente em um contexto emergencial onde os sistemas de saúde convencionais são insuficientes para lidar com o aumento da demanda.

Esse modelo foi desenvolvido para permitir a análise dos fluxos de pacientes e a otimização de recursos, considerando variáveis controláveis, como o número de leitos disponíveis e funcionários, e variáveis não controláveis, como a chegada de novos pacientes e a proporção de infectados com COVID-19.

O hospital foi estruturado em várias áreas: triagem, exames, medicação, e uma Unidade de Curta Permanência, além de um pronto-socorro para estabilização de casos críticos. Os pacientes são classificados por gravidade, o que influenciava o fluxo dentro do hospital. A otimização buscou quantificar os recursos necessários para o atendimento aos requisitos de *Length of Stay* (LOS) médio da unidade. O LOS é uma das métricas de performance mais

importantes de um ambiente hospitalar, ele corresponde ao tempo decorrido entre o momento de registro do paciente e momento que o paciente deixa o hospital. Logo, quanto menor for o LOS, maior será a satisfação do paciente, e conseqüentemente, a capacidade do sistema de saúde. As variáveis de decisão do problema correspondem a equipe do hospital, isto é, número de médicos (x_1), técnicos de enfermagem (x_2), profissionais de limpeza e higienização (x_3), enfermeiros de triagem (x_4), enfermeiros da unidade de curta permanência (x_5), além dos recursos físicos disponíveis, isto é, estações de triagem (x_6), leitos de emergência (x_7), salas de exame (x_8), estações de medicação (x_9) e leitos da unidade de curta permanência (x_{10}). Os limites inferior e superior são, respectivamente, $\mathbf{l} = [1\ 5\ 1\ 1\ 1\ 1\ 1\ 3\ 10\ 10]^T$ e $\mathbf{u} = [5\ 10\ 5\ 3\ 3\ 3\ 5\ 10\ 20\ 20]^T$. A Tabela 6 apresenta as variáveis de decisão e limites associados.

Tabela 6: Variáveis de decisão e limites (OE3)

Variáveis	Descrição	Limite Inf. (\mathbf{l})	Limite Sup. (\mathbf{u})
x_1	Nº de médicos	1	5
x_2	Nº de técnicos de enfermagem	5	10
x_3	Nº de profissionais de limpeza	1	5
x_4	Nº de enfermeiros de triagem	1	3
x_5	Nº de enfermeiros da unidade de curta permanência	1	3
x_6	Nº de estações de triagem	1	3
x_7	Nº de leitos de emergência	1	5
x_8	Nº de salas de exame	3	10
x_9	Nº de estações de medicação	10	20
x_{10}	Nº de leitos nas unidades de curta permanência	10	20

A função objetivo ($F(\mathbf{x})$) é dada pelo módulo da diferença entre o LOS médio dos pacientes atendidos no hospital e o LOS alvo esperado para esta unidade (representando o nível de atendimento desejado), ou seja, caso o LOS seja maior que o alvo, significa que o hospital não está atendendo seus pacientes no nível de serviço desejável e que talvez os recursos estejam subdimensionados. Por outro lado, caso o LOS esteja menor que o alvo, significa que o hospital está alocando mais recursos que o necessário, incorrendo em custos e deslocando equipes de outras unidades do sistema de saúde municipal desnecessariamente. Portanto, o objetivo deste problema é a minimização desta diferença, dada pela Eq. (23)(19).

$$F(\mathbf{x}) = |LOS_{médio} - LOS_{alvo}| \quad (23)$$

Na Eq. (23)(19), $F(\mathbf{x})$ representa o modulo da diferença entre o LOS médio encontrado para os pacientes atendidos durante a simulação ($LOS_{médio}$) e o LOS requisitado pelo sistema de saúde (LOS_{alvo}), de 146 minutos. O objetivo é minimizar a expectativa da função objetivo, calculada pela média de 10 replicações do modelo de simulação ($\tilde{\phi}$) com diferentes sementes aleatórias ω , ou seja, $\min f(\mathbf{x}) = \mathbb{E}_{\omega}\{\tilde{\phi}[F(\mathbf{x}, \omega)]\}$. O modelo matemático que representa o OE3 é dado pela minimização da função objetivo Eq. (24), sujeito as restrições (Eq. (25) e Eq. (26)) referentes ao espaço de busca.

$$\min f(\mathbf{x}) = \mathbb{E}_{\omega}\{\tilde{\phi}[F(\mathbf{x}, \omega)]\} \quad (24)$$

Sujeito a:

$$l_i \leq x_i \leq u_i \quad (25)$$

$$x_i \in \mathbb{Z}, \quad \forall i \in \{1, \dots, 10\} \quad (26)$$

É importante destacar que o valor da FO é derivado da média das replicações do modelo de SED, representado por $\tilde{\phi}$. Este modelo abrange diversas restrições operacionais, estocasticidade, interdependências e complexidades que não são delineadas no modelo matemático. Mais detalhes sobre este caso podem ser consultados em Campos *et al.* (2023b). A visualização 3D do modelo é apresentada na Figura 30.

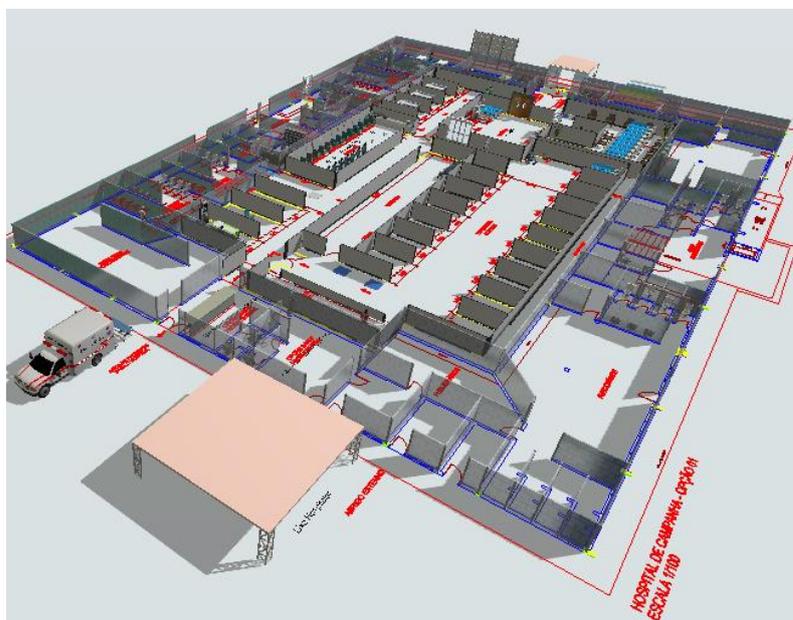


Figura 30: Modelo computacional (OE3)

Devido à maior complexidade e tempo de execução das simulações foi definido um limite de 100 experimentos (10d). Assim como nos casos anteriores, a otimização com EGO e AMSO foi repetida 10 vezes, cada uma com 100 experimentos, sendo 50% destinados à base inicial e o restante à fase adaptativa. A Seção 6.3.1 apresenta os resultados de AMSO e EGO,

a Seção 6.3.2 compara o AG com o AMSO, e a Seção 6.3.3 discute os resultados do AMSO com diferentes funções de aquisição.

6.3.1. Resultados OE3: AMSO vs. EGO com 100 experimentos

Esta seção apresenta os resultados otimização do problema OE3 considerando um *budget* de simulação de 100 experimentos. A Figura 31 apresenta os gráficos de convergência para o EGO e AMSO.

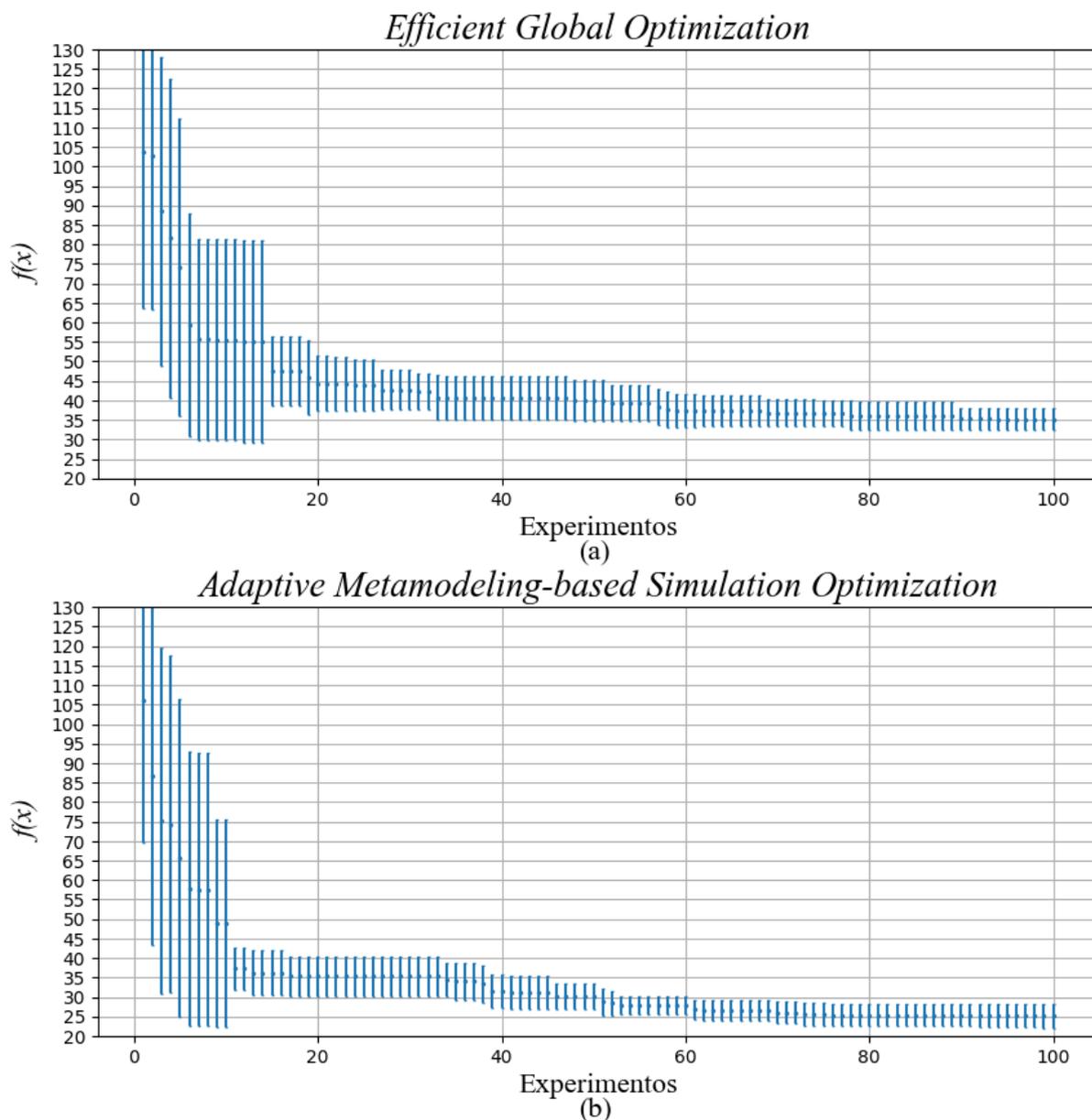


Figura 31: Gráfico de convergência para EGO (a) e o AMSO (b) com 100 experimentos (OE3)

Com um *budget* de simulação de 100 experimentos, a Figura 31 (a) mostra a média e barra de erro do gráfico de convergência obtido pelo EGO. O resultado médio alcançado pelo EGO foi de 35,09, com desvio padrão de 2,77. Dentro das repetições, a melhor solução

encontrada pelo EGO sugere o uso de $\mathbf{x}^* = [1\ 7\ 2\ 3\ 2\ 1\ 2\ 5\ 11\ 16]$ com $f(\mathbf{x}^*) = 30,22$. Por outro lado, conforme apresentado nas Figura 31 (b), o *framework* proposto obteve um valor de FO médio de 25,13 com um desvio padrão de 3,04. A melhor solução encontrada nas repetições foi $\mathbf{x}^* = [1\ 6\ 5\ 1\ 3\ 3\ 5\ 4\ 12\ 13]$ com $f(\mathbf{x}^*) = 22,39$.

Conforme apresentado na Figura 32 (a), ao comparar os dois *frameworks* pelo teste de Mann-Whitney, observou-se que o AMSO superou o EGO com significância estatística ($p\text{-value} = 0,000$), apresentando diferença na mediana e média de 9,27 e 9,95, respectivamente. Além disso, conforme observado na Figura 32 (b), não foi observada diferença significativa entre o tempo requerido pelo EGO e AMSO.

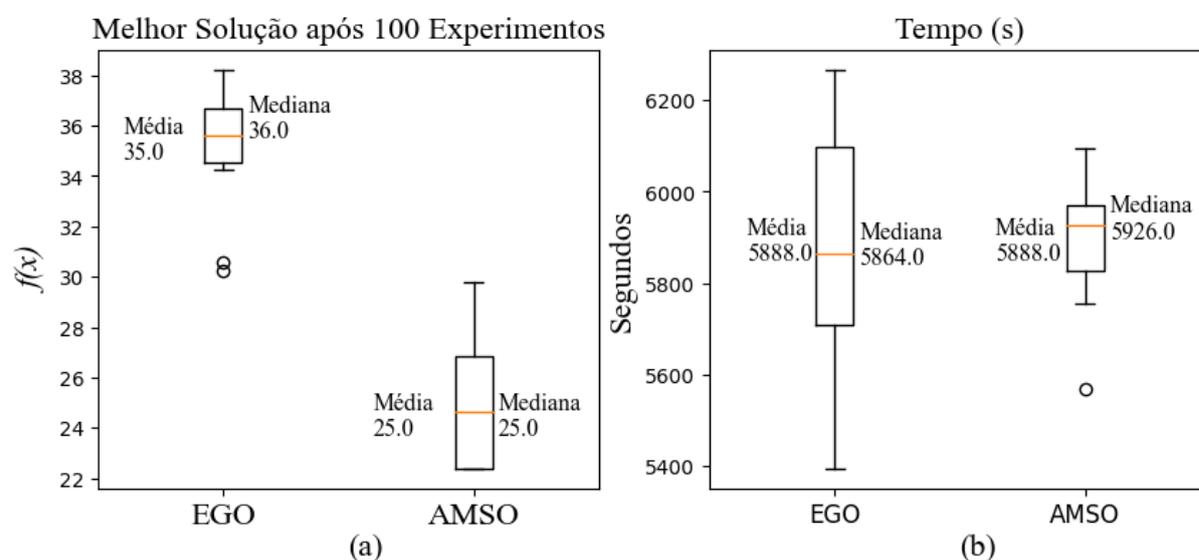


Figura 32: *Boxplot* para valor de FO atingido após 100 experimentos (a) e tempo gasto na otimização (b) (OE3)

6.3.2. Resultados OE3: AMSO vs. AG

Assim como nos demais casos analisados este teste não o limitou o AG ao mesmo *budget* aplicado ao AMSO. O AG foi definido com um número considerável de gerações para analisar se e quando este método supera os resultados obtidos pelo AMSO. Os parâmetros do AG foram definidos como 20 indivíduos para o tamanho da população e o número máximo de iterações igual a 100. Os demais parâmetros foram configurados conforme o padrão do pacote *Python geneticalgorithm2*. A Figura 33 apresenta a média e o erro do gráfico de convergência do AG (10 repetições). Vale ressaltar que, em todos as 10 repetições, o AG convergiu antes da centésima, variando entre 25 e 41 gerações.

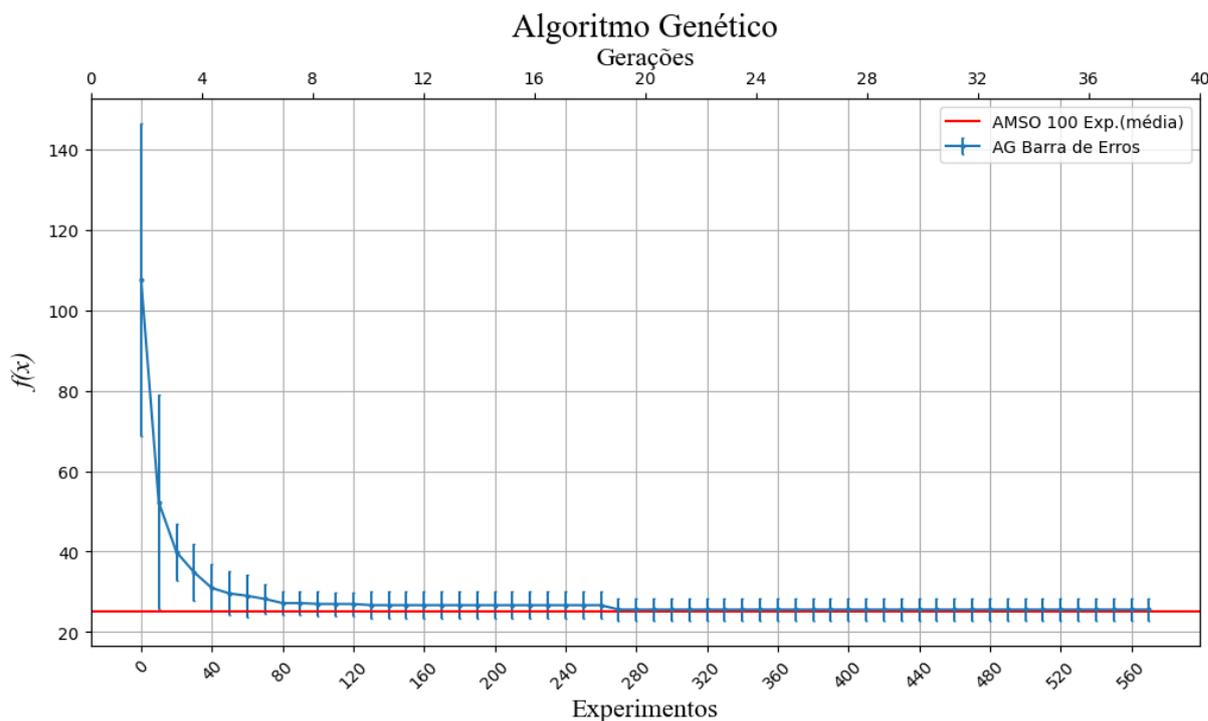


Figura 33: Gráfico de convergência do AG para o OE3

Conforme destacado na Figura 33, o AG demandou 264 experimentos para estabilizar em um resultado médio de 25.61, valor que permaneceu até a última geração, após 580 experimentos. Pelo teste de Mann-Whitney esse resultado, embora inferior, é estatisticamente igual ao AMSO com apenas 100 experimentos (25.13) com um *p-value* de 0,821. Para convergência, o AG demorou, em média, 16.916 segundos, quase três vezes mais do que o requerido pelo AMSO (5.888 segundos). Similar ao AMSO, a melhor solução encontrada pelo AG (dentre as 10 repetições) alcançou um $f(x^*)$ de 22,39, com $x^* = [1 \ 10 \ 3 \ 1 \ 1 \ 3 \ 4 \ 4 \ 20 \ 12]$.

Vale destacar que o foco não é estabelecer a superioridade do AMSO sobre o AG, uma vez que cada um pertence a uma classe distinta de algoritmos de otimização, cada qual com suas próprias características e aplicações específicas. Entretanto, os resultados indicam que o AMSO atinge desempenhos comparáveis aos do AG com um número significativamente menor de experimentos e em menos tempo, evidenciando a eficiência do *framework* proposto para esta aplicação.

6.3.3. Comparativo das Funções de Aquisição para o OE3

Nesta seção, é apresentado um estudo de sensibilidade deste parâmetro nos resultados do AMSO para o OE3, utilizando 100 experimentos repetidos 10 vezes para cada função de aquisição. Como apresentado na Figura 34, e conforme teste ANOVA (*p*-valor igual a 0,675), não é possível observar diferenças significativas nos dados. Similarmente, também não foi observado diferença significativa no tempo computacional. Portanto, conclui-se que, para o

OE3, o AMSO também apresentou robustez quanto a este parâmetro e a função BEI foi competitiva em relação às demais funções da literatura.

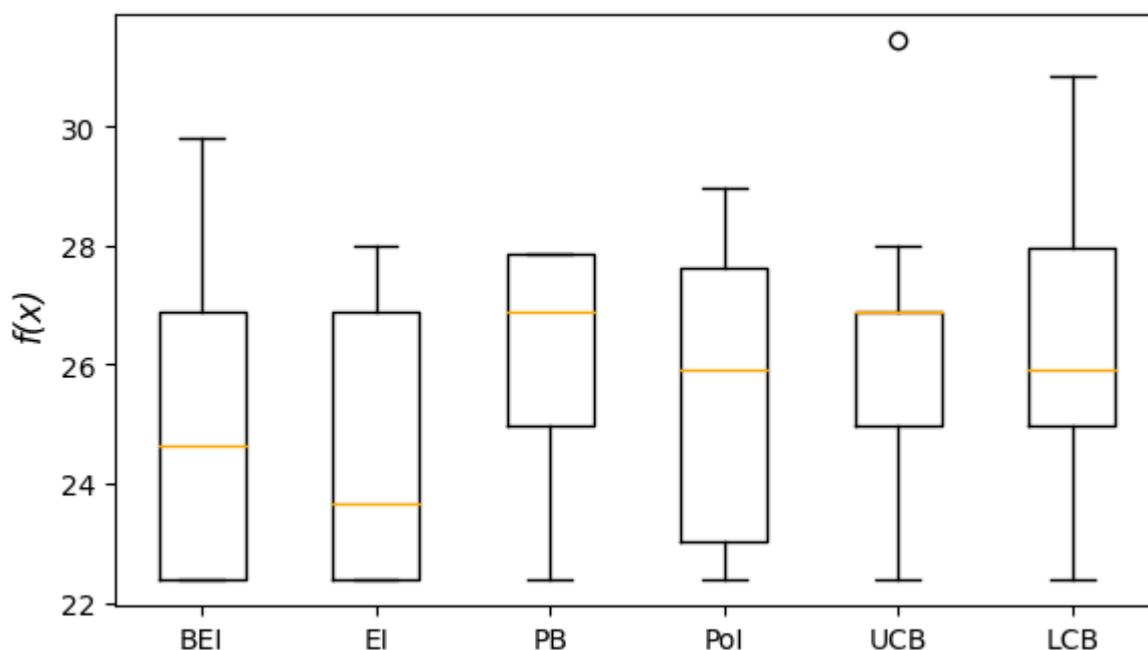


Figura 34: Comparativo entre funções de aquisição para o OE3

7. CONCLUSÕES

7.1. Verificação dos Objetivos Específicos

Este Capítulo objetiva retomar e verificar o atendimento aos objetivos específicos propostos no Seção 1.2.

1. Conduzir uma revisão sistemática de literatura (RSL) para analisar estado da arte em metamodelagem adaptativa em OvS, identificando as oportunidades e lacunas da literatura a fim de embasar a proposição do framework AMSO.

Este objetivo foi desenvolvido ao longo do Capítulo 4, levantando e discutindo o estado da arte sobre metamodelagem adaptativa em problemas de OvS. Esta RSL permitiu identificar lacunas e oportunidades, fundamentando o elemento inovador desta tese. Além disto, a RSL possibilitou a identificação e análise de boas práticas da literatura quanto às etapas e técnicas empregadas nos estudos de MA, provendo insumos importantes para o desenvolvimento do *framework* proposto neste trabalho.

2. Implementar o *framework* AMSO em um ambiente de programação (*Python*) em uma estrutura integrada ao *software* de simulação adotado (*FlexSim*[®]).

O desenvolvimento desse objetivo teve início com a análise do referencial teórico delineado no Capítulo 2 e da RSL no Capítulo 4. Esses elementos proporcionaram o alicerce teórico para a concepção do *framework*. Por fim, o Capítulo 5 apresenta as premissas adotadas, as técnicas utilizadas e a arquitetura do *framework* AMSO. Mais informações sobre o *framework* estão descritas no Capítulo 5.

Considerando que o *framework* proposto integra diversas técnicas (LHD, B-GBT, AG, paralelismo, entre outras), a sua implementação depende de um ambiente de programação flexível e robusto. Além disto, este ambiente deve prover meios para integração com *software* de SED a ser empregado, controlando-o externamente (configurar e alterar variáveis, executar simulações e coletar resultados). Baseado nestes requisitos, o ambiente *Python* foi empregado para implementação do AMSO, principalmente devido à grande disponibilidade de pacotes de ML, DoE, otimização e manipulação de dados, além de permitir o desenvolvimento de uma comunicação eficiente com o *software* FlexSim® para execução de simulações em paralelo. Mais detalhes foram elaborados no Capítulo 5.

A fim de avaliar a efetividade do AMSO, o *framework* proposto foi aplicado na otimização de três problemas reais da indústria, conforme apresentado no Capítulo 6. O primeiro envolve a aplicação do *framework* em um modelo SED que atua como um Gêmeo Digital para a alocação periódica de recursos em uma indústria têxtil. O segundo trata da otimização da expansão de capacidades de uma planta de mineração/moagem de grande porte. Já o terceiro aborda um problema de alocação de recursos físicos e humanos em uma unidade hospitalar.

Por fim, o Capítulo 6 também apresenta e discute os resultados obtidos com a aplicação do AMSO, EGO e AG aos objetos de estudo, avaliando e comparando a performance do *framework* proposto. Esta avaliação foi realizada em dois testes para o primeiro objeto de estudo com *budgets* distintos e um teste para o segundo e terceiro objeto de estudo, obtendo resultados promissores em ambos.

7.2. Considerações Finais

A integração entre otimização e simulação é uma abordagem poderosa e muito utilizada no aprimoramento de sistemas dinâmicos e estocásticos. No entanto, a execução de simulações, especialmente com modelos complexos, pode ser computacionalmente intensiva, impactando negativamente a agilidade do processo de otimização. Nesse contexto, a metamodelagem se apresenta como uma estratégia promissora para mitigar essa problemática, permitindo uma

otimização mais eficiente e ágil ao elevar o problema a um nível de abstração superior e reduzindo a necessidade de inúmeras execuções de simulações computacionais onerosas. Este estudo explora e discute o papel da metamodelagem para aprimorar a eficiência da otimização em modelos de simulação complexos e computacionalmente intensivos.

Esta tese teve como objetivo principal o desenvolvimento de um *framework* de metamodelagem adaptativa, denominado AMSO, que integra conceitos de DoE, *Bagging*, GBT, otimização de hiperparâmetros e Metaheurística (AG) para otimizar modelos de SED sob um *budget* limitado de experimentos. O *framework* consiste em dez etapas principais, conforme elaboradas no Capítulo 5, e visa treinar metamodelos (B-GBT) utilizando uma estratégia de amostragem adaptativa. Esta estratégia seleciona um novo ponto para simular e adicionar ao banco de dados de treinamento do metamodelo a cada iteração. A seleção do novo ponto é baseada na otimização da função de aquisição BEI, EI, PB, PoI, UCB ou LCB, utilizando algoritmo genético. A função BEI foi desenvolvida com o objetivo de equilibrar o *trade-off* entre busca global e local usando previsões do metamodelo previamente treinado.

Este trabalho apresentou três objetos de estudo reais para exemplificar a aplicabilidade do AMSO. Os métodos EGO e AG também foram aplicados aos objetos de estudo e serviram como *benchmarking*. O primeiro caso compreende um modelo de Gêmeo Digital de uma fábrica de *fast fashion*, otimizando recursos produtivos para atender à demanda semanal prevista. O objetivo é maximizar o lucro da fábrica, o que envolve oito variáveis de decisão discretas. Em dois testes realizados, o *budget* de simulação foi limitado a 80 e 160 experimentos para o AMSO e EGO. Devido às suas características, o AG foi configurado com um *budget* de simulação maior a fim de avaliar quando e se superaria os resultados do AMSO.

Os resultados demonstraram a superioridade do *framework* proposto. Em média, o AMSO encontrou uma solução 7,30% e 8,90% melhor que o EGO com 80 (10d) e 160 (20d) experimentos, respectivamente. Além disso, considerando os valores médios, o AG despense 391 experimentos para superar o AMSO com 80 experimentos, ou seja, necessitando de 388,75% mais simulações. Além disto, o AG terminou após a execução de 100 gerações (aproximadamente 1320 experimentos) com um resultado estatisticamente igual ao AMSO com apenas 160 experimentos, que exigiu 5,53 vezes menos tempo que o AG.

O segundo caso apresentou um problema de expansão de uma planta de mineração/moagem visando maximizar o lucro adicional esperado por meio do aumento da capacidade de sete recursos produtivos. Considerando a complexidade deste modelo, o orçamento para testar e comparar o AMSO e o EGO foi definido em 70 experimentos (10d). Além disso, o AG foi limitado a 50 gerações (com 20 indivíduos cada). Como resultado, o

AMSO obteve um lucro médio de US\$ 39,10 milhões, valor 9,67% melhor que o EGO, com tempo computacional semelhante. Além disso, o AG terminou após 50 gerações (706 experimentos) com um resultado estatisticamente igual ao AMSO, mas exigindo 314.464,58 segundos, 10,68 vezes mais tempo que o AMSO.

O terceiro objeto de estudo representa um problema de alocação de recursos em um hospital temporários planejado durante a pandemia de COVID-19. Este problema possui 10 variáveis de decisão e visa a minimização da diferença entre o LOS médio dos pacientes e o LOS alvo do sistema. O orçamento para testar o AMSO e EGO foi definido em 100 experimentos (10d) e o AG foi limitado ao máximo 100 gerações (aproximadamente 1300 experimentos). Como resultado, o AMSO atingiu um $f(x)$ médio de 25,13, valor 28,38% menor que o alcançado pelo EGO com o tempo computacional semelhante, e estatisticamente igual ao alcançado pelo AG, que demandou 2,87 vezes mais tempo que o AMSO.

A estrutura proposta neste estudo produziu resultados consistentes para três problemas típicos de SED: alta estocasticidade e interdependência de processos. Essas características necessitam de metamodelos de alta capacidade de aprendizado, como os empregados nesta pesquisa. Vale ressaltar que o EGO, um método de otimização baseado na metamodelagem por *Kriging*, tem sido amplamente utilizado na literatura e tem produzido excelentes resultados para uma ampla gama de problemas, assim como o AG. Além disso, AG é um método de otimização poderoso e muito utilizado, e se o modelador tiver tempo ilimitado para realizar a otimização e puder levar de dias a semanas para tomar decisões, métodos metaheurísticos podem ser utilizados. Contudo, o tempo ilimitado não é a realidade na maioria dos projetos.

Com base nos resultados apresentados, destaca-se o potencial da estrutura AMSO para resolver problemas de OvS complexos. Outra vantagem do AMSO é que, ao final da otimização, um de seus resultados é um metamodelo treinado, que o tomador de decisão poderia usar para explorar outros cenários potenciais e realizar a otimização com novas restrições sem exigir execuções de novas simulações dispendiosas, agilizando o processo de tomada de decisão.

Este trabalho utilizou o GBT como *base-learner* do *Bagging*. Assim, uma possível extensão desta pesquisa seria a avaliação e comparação sistemática de outros algoritmos de ML, como *Support Vector Machine*, Redes Neurais Artificiais e *Extreme Gradient-boosting*. Além disso, o fator crítico em abordagens de metamodelagem adaptativa é a definição da função de aquisição a ser empregada para orientar o processo de *infill-search*. O *framework* AMSO utiliza como *default* a função BEI para equilibrar a busca local e global com base no R^2 do metamodelo, mas também possibilita o uso de outras funções de aquisição da literatura, como

a EI, PB, PoI, LBC e UBC. Entretanto, sugere-se que novas pesquisas implementem e avaliem outras funções de aquisição. Recomenda-se que uma futura pesquisa possa avaliar a combinação do *framework* AMSO com outros métodos da literatura, visando aumentar sua eficácia, eficiência e gama de aplicação.

Além disso, sugere-se que futuros estudos avaliem e comparem o AMSO para outras classes de problemas de SED, tais como controle de estoque, roteirização e programação de produção, ou outros tipos de simulação, como CFD, Elementos Finitos e Monte Carlo. Para além da simulação, o AMSO pode ser testado e avaliado em problemas de otimização de modelos matemáticos.

Por fim, sugere-se que futuros trabalhos apliquem o método em um ambiente controlado, com cases teóricos com diferentes características, com o objetivo de avaliar a sensibilidade do AMSO quanto a estas características e concluir sobre sua aplicabilidade perante as peculiaridades dos diversos problemas de OvS.

REFERÊNCIAS BIBLIOGRÁFICAS

AHRARI, Ali *et al.* A proximity-based surrogate-assisted method for simulation-based design optimization of a cylinder head water jacket. **Engineering Optimization**, [s. l.], v. 53, n. 9, p. 1574–1592, 2021.

ALABAS, C.; ALTIPARMAK, F.; DENGIZ, B. A comparison of the performance of artificial intelligence techniques for optimizing the number of kanbans. **Journal of the Operational Research Society**, [s. l.], v. 53, n. 8, p. 907–914, 2002.

AMARAL, João Victor Soares do *et al.* Data envelopment analysis for algorithm efficiency assessment in metamodel-based simulation optimization. **International Journal of Advanced Manufacturing Technology**, [s. l.], v. 121, n. 11–12, p. 7493–7507, 2022a.

AMARAL, João Victor Soares do *et al.* Data envelopment analysis for algorithm efficiency assessment in metamodel-based simulation optimization. **The International Journal of Advanced Manufacturing Technology**, [s. l.], v. 121, n. 11–12, p. 7493–7507, 2022b. Disponível em: <https://link.springer.com/10.1007/s00170-022-09864-z>.

AMARAL, João Victor Soares do *et al.* Metamodel-based simulation optimization: A systematic literature review. **Simulation Modelling Practice and Theory**, [s. l.], v. 114, 2022c.

AMARAL, João Victor Soares do *et al.* Metamodeling-based simulation optimization in manufacturing problems: a comparative study. **The International Journal of Advanced Manufacturing Technology**, [s. l.], p. 20, 2022d. Disponível em: <https://link.springer.com/10.1007/s00170-022-09072-9>.

AMARAL, João Victor Soares do; MONTEVECHI, José Arnaldo Barra; MIRANDA, Rafael de Carvalho. Metamodel-Based Simulation Optimization: A Systematic Literature Review. *In: UNCERTAINTY MODELING: FUNDAMENTAL CONCEPTS AND MODELS*. [S. l.]: Biblioteca Central da Universidade de Brasília, 2022. p. 119–171.

AMARAN, Satyajith *et al.* Simulation optimization: a review of algorithms and applications. **Annals of Operations Research**, [s. l.], v. 240, n. 1, p. 351–380, 2016.

AOUADNI, Sourour; AOUADNI, Ismahene; REBAÏ, Abdelwaheb. A systematic review on supplier selection and order allocation problems. **Journal of Industrial Engineering International**, [s. l.], v. 15, n. 2009, p. 267–289, 2019.

APORNAK, Arash *et al.* Human resources optimization in hospital emergency using the genetic algorithm approach. **International Journal of Healthcare Management**, [s. l.], v. x, p. 1–8, 2020.

BAMDAD, Keivan; CHOLETTE, Michael E.; BELL, John. Building energy optimization using surrogate model and active sampling. **Journal of Building Performance Simulation**, [s. l.], v. 13, n. 6, p. 760–776, 2020.

BANKS, Jerry *et al.* **Discrete-Event System Simulation**. 5th. ed. London: Pearson, 2014.

BARLAS, Panagiotis; HEAVEY, Cathal. Automation of input data to discrete event simulation for manufacturing: A review. **International Journal of Modeling, Simulation, and Scientific Computing**, [s. l.], v. 7, n. 1, 2016.

BASAGLIA, Alberto *et al.* A Discrete-Event Simulation Model of Hospital Patient Flow Following Major Earthquakes. **International Journal of Disaster Risk Reduction**, [s. l.], v. 71, 2022.

BELGIN, Onder. Data Envelopment Analysis Based Metamodeling for Multi Objective Simulation Optimization in a Manufacturing Line. **Sigma Journal of Engineering and Natural Sciences**, [s. l.], v. 37, n. 4, p. 1435–1449, 2019.

BERGMEIR, Christoph; BENÍTEZ, José M. On the use of cross-validation for time series predictor evaluation. **Information Sciences**, [s. l.], v. 191, p. 192–213, 2012.

BERTOLINI, Roberto; FINCH, Stephen J.; NEHM, Ross H. Quantifying variability in predictions of student performance: Examining the impact of bootstrap resampling in data pipelines. **Computers and Education: Artificial Intelligence**, [s. l.], v. 3, 2022.

BERTRAND, J. Will M.; FRANSOO, Jan C.; UDENIO, Maxi. Model-based research. *In: RESEARCH METHODS FOR OPERATIONS AND SUPPLY CHAIN MANAGEMENT*. 3. ed. [S. l.: s. n.], 2023. v. 9.

BHARAJ, Gaurav *et al.* Computational design of metallophone contact sounds. *In:* , 2015. **ACM Transactions on Graphics**. [S. l.]: Association for Computing Machinery, 2015.

BIN MAKHASHEN, Yousef *et al.* Exploring the role of ambidexterity and coopetition in designing resilient fashion supply chains: a multi-evidence-based approach. **Journal of Enterprise Information Management**, [s. l.], 2020.

BOOTH, Andrew; SUTTON, Anthea; PAPAIOANNOU, Diana. **Systematic Approaches to a Successful Literature Review**. Seconded. [S. l.]: SAGE Publications, 2016-. ISSN 1708-6892.v. 34

BOUHLEL, Mohamed Amine *et al.* A Python surrogate modeling framework with derivatives. **Advances in Engineering Software**, [s. l.], v. 135, 2019.

BOUKOUVALA, Fani; IERAPETRITOU, Marianthi G. Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. **Journal of Pharmaceutical Innovation**, [s. l.], v. 8, n. 2, p. 131–145, 2013.

BRIFFOTEAUX, Guillaume *et al.* Parallel surrogate-assisted optimization: Batched Bayesian Neural Network-assisted GA versus q-EGO. **Swarm and Evolutionary Computation**, [s. l.], v. 57, 2020.

CAMPOS, Afonso Teberga *et al.* Integrating computer simulation and the normalized normal constraint method to plan a temporary hospital for COVID-19 patients. **Journal of the Operational Research Society**, [s. l.], v. 74, n. 2, p. 562–573, 2023a.

CAMPOS, Afonso Teberga *et al.* Integrating computer simulation and the normalized normal constraint method to plan a temporary hospital for COVID-19 patients. **Journal of the Operational Research Society**, [s. l.], v. 74, n. 2, p. 562–573, 2023b.

CANDELIERI, Antonio *et al.* Lost in optimization of water distribution systems: better call Bayes. **Water**, [s. l.], 2022. Disponível em: www.preprints.org.

CANDELIERI, A.; PEREGO, R.; ARCHETTI, F. Bayesian optimization of pump operations in water distribution systems. **Journal of Global Optimization**, [s. l.], v. 71, n. 1, p. 213–235, 2018.

CASTILLO, Enrique *et al.* A matrix-T approach to the sequential design of optimization experiments A matrix-T approach to the sequential design of optimization experiments. **IEE Transactions**, [s. l.], v. 8830, 2010.

CHABANET, Sylvain; BRIL EL-HAOUZI, Hind; THOMAS, Philippe. Coupling digital simulation and machine learning metamodel through an active learning approach in Industry 4.0 context. **Computers in Industry**, [s. l.], v. 133, 2021.

CHANG, Tyler H. *et al.* Algorithm 1028: VTMOP: Solver for Blackbox Multiobjective Optimization Problems. **ACM Transactions on Mathematical Software**, [s. l.], v. 48, n. 3, 2022.

CHANG, Kuo Hao; HSIEH, Liam Y. Determination of Wafer Start Mix in Semiconductor Manufacturing during New Technology Ramp-Up: Model, Solution Method, and an Empirical Study. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, [s. l.], v. 46, n. 2, p. 294–302, 2016.

CHAUDHRY *, S. S.; LUO, W. Application of genetic algorithms in production and operations management: a review. **International Journal of Production Research**, [s. l.], v. 43, n. 19, p. 4083–4101, 2005. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00207540500143199>.

CHEN, Ken *et al.* Bagging based ensemble learning approaches for modeling the emission of PCDD/Fs from municipal solid waste incinerators. **Chemosphere**, [s. l.], v. 274, 2021.

CHEN, Wenjie; LIU, Shengcai; TANG, Ke. A new knowledge gradient-based method for constrained bayesian optimization. **arXiv preprint arXiv:2101.08743**, 2021.

CHICCO, Davide; WARRENS, Matthijs J.; JURMAN, Giuseppe. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. **PeerJ Computer Science**, [s. l.], v. 7, p. 1–24, 2021.

CHO, Gyeongcheol; JUNG, Kwanghee; HWANG, Heungsun. Out-of-bag Prediction Error: A Cross Validation Index for Generalized Structured Component Analysis. **Multivariate Behavioral Research**, [s. l.], v. 54, n. 4, p. 505–513, 2019.

CHRISTELIS, Vasileios; MANTOGLOU, Aristotelis. Pumping Optimization of Coastal Aquifers Assisted by Adaptive Metamodelling Methods and Radial Basis Functions. **Water Resources Management**, [s. l.], v. 30, n. 15, p. 5845–5859, 2016.

COELHO, Igor Machado *et al.* OPTFRAME: A Computational Framework for Combinatorial Optimization Problems. *In:* , 2011, Porto. **VII ALIO-EURO**. Porto: [s. n.], 2011. p. 51–54.

COELHO, Guilherme F.; PINTO, Luiz R. Kriging-based simulation optimization: An emergency medical system application. **Journal of the Operational Research Society**, [s. l.], v. 69, n. 12, p. 2006–2020, 2018. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/01605682.2017.1418149>.

CUCKLER, Robert; CHANG, Kuo Hao; HSIEH, Liam Y. Optimal Parallel Machine Allocation Problem in IC Packaging Using IC-PSO: An Empirical Study. **Asia-Pacific Journal of Operational Research**, [s. l.], v. 34, n. 6, p. 1–20, 2017.

DE JONG, Kenneth Alan. **Analysis of the Behavior of a Class of Genetic Adaptive Systems**. 1975. Thesis - University of Michigan, Hampton, Virginia, 1975.

DEL CASTILLO, Enrique; SANTIAGO, Eduardo. A matrix-T approach to the sequential design of optimization experiments. **IIE Transactions (Institute of Industrial Engineers)**, [s. l.], v. 43, n. 1, p. 54–68, 2011.

DENGIZ, Berna; BEKTAS, Tolga; ULTANIR, A. Eren. Simulation optimization based DSS application: A diamond tool production line in industry. **Simulation Modelling Practice and Theory**, [s. l.], v. 14, n. 3, p. 296–312, 2006.

DENYER, David; TRANFIELD, David; VAN AKEN, Joan Ernst. Developing design propositions through research synthesis. **Organization Studies**, [s. l.], v. 29, n. 3, p. 393–413, 2008.

DESAUTELS, Thomas; KRAUSE, Andreas; BURDICK, Joel. Parallelizing exploration-exploitation tradeoffs with Gaussian process bandit optimization. **Journal of Machine Learning Research**, [s. l.], v. 15, p. 1191–1198, 2014.

DOS SANTOS, Carlos Henrique *et al.* Decision support in productive processes through DES and ABS in the Digital Twin era: a systematic literature review. **International Journal of Production Research**, [s. l.], v. 60, n. 8, p. 2662–2681, 2022. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/00207543.2021.1898691>.

DOS SANTOS, Carlos Henrique *et al.* Decision-making in a fast fashion company in the Industry 4.0 era: a Digital Twin proposal to support operational planning. **International Journal of Advanced Manufacturing Technology**, [s. l.], v. 116, n. 5–6, p. 1653–1666, 2021.

DOS SANTOS, Carlos Henrique *et al.* Simulation-based digital twins monitoring: an approach focused on models' accreditation. **The International Journal of Advanced Manufacturing Technology**, [s. l.], v. 124, n. 7–8, p. 2423–2435, 2023. Disponível em: <https://link.springer.com/10.1007/s00170-022-10674-6>.

DUNKE, Fabian; NICKEL, Stefan. Neural networks for the metamodeling of simulation models with online decision making. **Simulation Modelling Practice and Theory**,

[s. l.], v. 99, n. September 2019, p. 102016, 2020. Disponível em: <https://doi.org/10.1016/j.simpat.2019.102016>.

FAN, Wenxuan *et al.* A novel surrogate model for channel geometry optimization of PEM fuel cell based on Bagging-SVM Ensemble Regression. **International Journal of Hydrogen Energy**, [s. l.], v. 47, n. 33, p. 14971–14982, 2022.

FAN, Qi; HU, Jiaqiao. Surrogate-based promising area search for Lipschitz continuous simulation optimization. **INFORMS Journal on Computing**, [s. l.], v. 30, n. 4, p. 677–693, 2018.

FEYGIN, Sidney A. *et al.* BISTRO: Berkeley Integrated System for Transportation Optimization. **ACM Transactions on Intelligent Systems and Technology**, [s. l.], v. 11, n. 4, 2020.

FRIEDMAN, Jerome H. Stochastic gradient boosting. **Computational Statistics and Data Analysis**, [s. l.], v. 38, n. 4, p. 367–378, 2002.

FU, Michael C. Optimization for simulation: Theory vs. Practice. **INFORMS Journal on Computing**, [s. l.], v. 14, n. 3, p. 192–215, 2002. Disponível em: <https://pubsonline.informs.org/doi/10.1287/ijoc.14.3.192.113>.

GANJISAFFAR, Yasser; CARUANA, Rich; LOPES, Cristina Videira. Bagging gradient-boosted trees for high precision, low variance ranking models. **SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval**, [s. l.], n. c, p. 85–94, 2011.

GENGEMBRE, E. *et al.* A Kriging constrained efficient global optimization approach applied to low-energy building design problems. **Inverse Problems in Science and Engineering**, [s. l.], v. 20, n. 7, p. 1101–1114, 2012.

GIAHI, Ramin; MACKENZIE, Cameron A; HU, Chao. Optimizing the flexible design of hybrid renewable energy systems. **The Engineering Economist**, [s. l.], v. 0, n. 0, p. 1–27, 2022. Disponível em: <https://doi.org/10.1080/0013791X.2022.2028047>.

GIRI, Brijesh Kumar *et al.* Genetic programming through bi-objective genetic algorithms with a study of a simulated moving bed process involving multiple objectives. **Applied Soft Computing Journal**, [s. l.], v. 13, n. 5, p. 2613–2623, 2013. Disponível em: <http://dx.doi.org/10.1016/j.asoc.2012.11.025>.

GORISSEN, Dirk; DHAENE, Tom; DE TURCK, Filip. Evolutionary model type selection for global surrogate modeling. **Journal of Machine Learning Research**, [s. l.], v. 10, p. 2039–2078, 2009.

GREASLEY, Andrew; OWEN, Chris. Modelling people's behaviour using discrete-event simulation: a review. **International Journal of Operations & Production Management**, [s. l.], v. 38, n. 5, p. 1228–1244, 2018. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/IJOPM-10-2016-0604/full/html>.

GU, J.; LI, G. Y.; DONG, Z. Hybrid and adaptive meta-model-based global optimization. **Engineering Optimization**, [s. l.], v. 44, n. 1, p. 87–104, 2012.

HACHICHA, W. A simulation metamodelling based neural networks for lot-sizing problem in MTO sector. **International Journal of Simulation Modelling**, [s. l.], v. 10, n. 4, p. 191–203, 2011.

HAN, Mei; OUYANG, Linhan. A novel Bayesian approach for multi-objective stochastic simulation optimization. **Swarm and Evolutionary Computation**, [s. l.], v. 75, 2022.

HASANZADEH, Sepideh Kheirkhah; SAADATPOUR, Motahareh; AFSHAR, Abbas. A fuzzy equilibrium strategy for sustainable water quality management in river-reservoir system. **Journal of Hydrology**, [s. l.], v. 586, 2020.

HASSANNAYEBI, Erfan *et al.* Train schedule optimization in a high-speed railway system using a hybrid simulation and meta-model approach. **Computers and Industrial Engineering**, [s. l.], v. 138, n. July, p. 106110, 2019. Disponível em: <https://doi.org/10.1016/j.cie.2019.106110>.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning**. New York, NY: Springer New York, 2009. (Springer Series in Statistics). v. 2 Disponível em: <http://link.springer.com/10.1007/978-0-387-84858-7>.

HE, Youwei *et al.* Dual Kriging assisted efficient global optimization of expensive problems with evaluation failures. **Aerospace Science and Technology**, [s. l.], v. 105, p. 106006, 2020. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S127096382030688X>.

HOAD, K.; ROBINSON, S.; DAVIES, R. Automated selection of the number of replications for a discrete-event simulation. **Journal of the Operational Research Society**, [s. l.], v. 61, n. 11, p. 1632–1644, 2010.

HOLLAND, John H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. 1^oed. Cambridge, Massachusetts: MIT Press, 1975.

HOU, Zeyu *et al.* Application of Mixed-Integer Nonlinear Optimization Programming Based on Ensemble Surrogate Model for Dense Nonaqueous Phase Liquid Source Identification in Groundwater. **Environmental Engineering Science**, [s. l.], v. 36, n. 6, p. 699–709, 2019.

HÜLLEN, Gordon *et al.* Managing uncertainty in data-driven simulation-based optimization. **Computers and Chemical Engineering**, [s. l.], v. 136, p. 106519, 2020. Disponível em: <https://doi.org/10.1016/j.compchemeng.2019.106519>.

IVANOVA, Momchil; KUHN, Sonja. A parallel optimization algorithm based on FANOVA decomposition. **Quality and Reliability Engineering International**, [s. l.], v. 30, n. 7, p. 961–974, 2014.

JONES, Donald R; SCHONLAU, Matthias; WELCH, William J. Efficient Global Optimization of Expensive Black-Box Functions. **Journal of Global Optimization**, [s. l.], v. 13, p. 455–492, 1998.

JUNG, Christian *et al.* Metamodel-based optimization of hot rolling processes in the metal industry. **International Journal of Advanced Manufacturing Technology**, [s. l.], v. 90, n. 1–4, p. 421–435, 2017.

KASAIE, Parastu; KELTON, W. David. Simulation optimization for allocation of epidemic-control resources. **IIE Transactions on Healthcare Systems Engineering**, [s. l.], v. 3, n. 2, p. 78–93, 2013.

KATOCH, Sourabh; CHAUHAN, Sumit Singh; KUMAR, Vijay. A review on genetic algorithm: past, present, and future. **Multimedia Tools and Applications**, [s. l.], v. 80, n. 5, p. 8091–8126, 2021.

KEYS, Anthony C; REES, Loren Paul; GREENWOOD, Allen G. Performance measures for selection, of metamodels to be used in simulation optimization. **Decision Sciences**, [s. l.], v. 33, n. 1, p. 31–58, 2002.

KIM, Sun Hye; BOUKOUVALA, Fani. Surrogate-based optimization for mixed-integer nonlinear problems. **Computers and Chemical Engineering**, [s. l.], v. 140, 2020.

KITCHENHAM, Barbara *et al.* Systematic literature reviews in software engineering - A systematic literature review. **Information and Software Technology**, [s. l.], v. 51, n. 1, p. 7–15, 2009. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2008.09.009>.

KLEIJNEN, Jack P.C. Kriging metamodeling in simulation: A review. **European Journal of Operational Research**, [s. l.], v. 192, n. 3, p. 707–716, 2009.

KLEIJNEN, Jack P.C. Regression and Kriging metamodels with their experimental designs in simulation: A review. **European Journal of Operational Research**, [s. l.], v. 256, n. 1, p. 1–16, 2017. Disponível em: <http://dx.doi.org/10.1016/j.ejor.2016.06.041>.

KLEIJNEN, Jack P. C. Simulation Optimization Through Regression or Kriging Metamodels. **SSRN Electronic Journal**, [s. l.], 2017.

KLEIJNEN, Jack P C. Simulation-optimization via Kriging and bootstrapping: a survey. **Journal of Simulation**, [s. l.], v. 8, n. 4, p. 241–250, 2014.

KOCHAN, Cigdem Gonul; NOWICKI, David R. Supply chain resilience: a systematic literature review and typological framework. **International Journal of Physical Distribution and Logistics Management**, [s. l.], v. 48, n. 8, p. 842–865, 2018.

KRAUSS, Christopher; DO, Xuan Anh; HUCK, Nicolas. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. **European Journal of Operational Research**, [s. l.], v. 259, n. 2, p. 689–702, 2017. Disponível em: <http://dx.doi.org/10.1016/j.ejor.2016.10.031>.

KUGALUR PALANISAMY, Nithyaraaj *et al.* Identification of the Parameter Values of the Constitutive and Friction Models in Machining Using EGO Algorithm: Application to Ti6Al4V. **Metals**, [s. l.], v. 12, n. 6, 2022.

LAL, Alvin; DATTA, Bithin. Performance Evaluation of Homogeneous and Heterogeneous Ensemble Models for Groundwater Salinity Predictions: a Regional-Scale Comparison Study. **Water, Air, & Soil Pollution**, [s. l.], v. 231, n. 6, p. 320, 2020. Disponível em: <https://link.springer.com/10.1007/s11270-020-04693-w>.

LAW, Averill M. **Simulation Modeling and Analysis**. 5th. ed. New York: McGraw-Hil, 2013-. ISSN 01621459. Disponível em: <https://www.jstor.org/stable/2288169?origin=crossref>.

LEE, Jaesung; ZHOU, Shiyu; CHEN, Junhong. Robust Parameter Design on Dual Stochastic Response Models With Constrained Bayesian Optimization. **IEEE Transactions on Automation Science and Engineering**, [s. l.], p. 1–12, 2023. Disponível em: <https://ieeexplore.ieee.org/document/10064253/>.

LESIK, Sally A. **Applied Statistical Inference with MINITAB®, Second Edition**. [S. l.]: Chapman and Hall/CRC, 2018. Disponível em: <https://www.taylorfrancis.com/books/9780429816659>.

LEVY, Yair; ELLIS, Timothy J. A systems approach to conduct an effective literature review in support of information systems research. **Informing Science**, [s. l.], v. 9, n. May 2014, p. 181–211, 2006.

LI, Yaohui *et al.* A kriging-assisted multi-objective constrained global optimization method for expensive black-box functions. **Mathematics**, [s. l.], v. 9, n. 2, p. 1–22, 2021.

LI, Y. F. *et al.* A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems. **Applied Soft Computing Journal**, [s. l.], v. 10, n. 4, p. 1257–1273, 2010.

LI, Hongxia *et al.* Design optimization of stent and its dilatation balloon using kriging surrogate model. **BioMedical Engineering Online**, [s. l.], v. 16, n. 1, 2017.

LIU, Yong *et al.* Inversion of hydrogeological parameters based on an adaptive dynamic surrogate model. **Hydrogeology Journal**, [s. l.], v. 30, n. 5, p. 1513–1527, 2022.

LIU, Haitao; ONG, Yew Soon; CAI, Jianfei. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. **Structural and Multidisciplinary Optimization**, [s. l.], v. 57, n. 1, p. 393–416, 2018.

LOUK, Maya Hilda Lestari; TAMA, Bayu Adhi. Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. **Expert Systems with Applications**, [s. l.], v. 213, 2023.

LUO, Jiannan *et al.* An adaptive dynamic surrogate model using a constrained trust region algorithm: application to DNAPL-contaminated-groundwater-remediation design. **Hydrogeology Journal**, [s. l.], v. 28, n. 4, p. 1285–1298, 2020.

LUO, Jiannan *et al.* Inversion of groundwater contamination source based on a two-stage adaptive surrogate model-assisted trust region genetic algorithm framework. **Applied Mathematical Modelling**, [s. l.], v. 112, p. 262–281, 2022.

LUO, Jiannan; JI, Yefei; LU, Wenxi. Comparison of Surrogate Models Based on Different Sampling Methods for Groundwater Remediation. **Journal of Water Resources Planning and Management**, [s. l.], v. 145, n. 5, p. 04019015, 2019. Disponível em: <http://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0001062>.

MAIRINQUE, Laryssa de Andrade *et al.* A bootstrap control chart for the availability index. **International Journal of Advanced Manufacturing Technology**, [s. l.], v. 120, n. 7–8, p. 5151–5161, 2022.

MCKAY, M D; BECKMAN, R J; CONOVER, W J. **A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code**. [S. l.: s. n.], 2000.

MEHDAD, Ehsan; KLEIJNEN, Jack P.C. Classic Kriging versus Kriging with bootstrapping or conditional simulation: Classic Kriging's robust confidence intervals and optimization. **Journal of the Operational Research Society**, [s. l.], v. 66, n. 11, p. 1804–1814, 2015.

MES, Martijn R.K.; POWELL, Warren B; FRAZIER, Peter I. Hierarchical Knowledge Gradient for Sequential Sampling. **Journal of Machine Learning Research**, [s. l.], v. 12, p. 2931–2974, 2011.

MIGUEL, PAC *et al.* **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. 3^oed. Rio de Janeiro: Elsevier, 2018.

MILANI, A. S. *et al.* Evaluating three DOE methodologies: Optimization of a composite laminate under fabrication error. **Quality Engineering**, [s. l.], v. 21, n. 1, p. 96–110, 2009.

MIRANDA, Rafael De Carvalho *et al.* A new approach to reducing search space and increasing efficiency in simulation optimization problems via the fuzzy-DEA-BCC. **Mathematical Problems in Engineering**, [s. l.], v. 2014, 2014.

MIRANDA, Rafael de Carvalho *et al.* Increasing the efficiency in integer simulation optimization: Reducing the search space through data envelopment analysis and orthogonal arrays. **European Journal of Operational Research**, [s. l.], v. 262, n. 2, p. 673–681, 2017.

MIRFENDERESGI, Golnazalsadat; MOUSAVI, S. Jamshid. Adaptive meta-modeling-based simulation optimization in basin-scale optimum water allocation: A comparative analysis of meta-models. **Journal of Hydroinformatics**, [s. l.], v. 18, n. 3, p. 446–465, 2016.

MITROFF, Ian I. *et al.* On Managing Science in the Systems Age: Two Schemas for the Study of Science as a Whole Systems Phenomenon. **Interfaces**, [s. l.], v. 4, n. 3, p. 46–58, 1974.

MOGHADDAM, Samira; MAHLOOJI, Hashem. A new metamodel-based method for solving semi-expensive simulation optimization problems. **Communications in Statistics: Simulation and Computation**, [s. l.], v. 46, n. 6, p. 4795–4811, 2017.

MOHAMMAD NEZHAD, Ali; MAHLOOJI, Hashem. An artificial neural network meta-model for constrained simulation optimization. **Journal of the Operational Research Society**, [s. l.], v. 65, n. 8, p. 1232–1244, 2014. Disponível em: <http://dx.doi.org/10.1057/jors.2013.73>.

MONTEVECHI, Jose Arnaldo Barra *et al.* Conceptual modeling in simulation projects by mean adapted IDEF: An application in a Brazilian tech company. *In:* , 2010, Baltimore, MD. **Proceedings of the 2010 Winter Simulation Conference**. Baltimore, MD: IEEE, 2010. p. 1624–1635. Disponível em: <http://ieeexplore.ieee.org/document/5678908/>.

MONTEVECHI, José Arnaldo Barra; MIRANDA, Rafael de Carvalho; DANIEL, Jonathan. Sensitivity Analysis in Discrete Event Simulation Using Design of Experiments. *In:* DISCRETE EVENT SIMULATIONS - DEVELOPMENT AND APPLICATIONS. [S. l.]: InTech, 2012.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied Statistics and Probability for Engineers**. 3rd. ed. New York: John Wiley & Sons, 2003-. ISSN 09641998.v. 158

MORALES-HERNÁNDEZ, Alejandro; VAN NIEUWENHUYSE, Inneke; ROJAS GONZALEZ, Sebastian. A survey on multi-objective hyperparameter optimization algorithms for machine learning. **Artificial Intelligence Review**, [s. l.], 2022.

MYERS, Raymond H.; MONTGOMERY, Douglas C.; ANDERSON-COOK, Christine M. **Response surface methodology: process and product optimization using designed experiments**. 4^oed. Hoboken, New Jersey: John Wiley & Sons, 2016.

NAKATSU, Robbie T. An Evaluation of Four Resampling Methods Used in Machine Learning Classification. **IEEE Intelligent Systems**, [s. l.], v. 36, n. 3, p. 51–57, 2021.

NIE, Peng *et al.* Prediction of home energy consumption based on gradient boosting regression tree. **Energy Reports**, [s. l.], v. 7, p. 1246–1255, 2021.

OLIVEIRA, Josenildo Brito; LIMA, Renato Silva; MONTEVECHI, José Arnaldo Barra. Perspectives and relationships in Supply Chain Simulation: A systematic literature review. **Simulation Modelling Practice and Theory**, [s. l.], v. 62, p. 166–191, 2016.

OSORIO, Carolina; CHONG, Linsen. A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. **Transportation Science**, [s. l.], v. 49, n. 3, p. 623–636, 2015.

ØSTERGÅRD, Torben; JENSEN, Rasmus Lund; MAAGAARD, Steffen Enersen. A comparison of six metamodeling techniques applied to building performance simulations. **Applied Energy**, [s. l.], v. 211, n. October 2017, p. 89–103, 2018.

PANG, Min *et al.* Efficient, parallelized global optimization of groundwater pumping in a regional aquifer with land subsidence constraints. **Journal of Environmental Management**, [s. l.], v. 310, n. February, p. 114753, 2022. Disponível em: <https://doi.org/10.1016/j.jenvman.2022.114753>.

PARNIANIFARD, Amir *et al.* A new approach for low-dimensional constrained engineering design optimization using design and analysis of simulation experiments.

International Journal of Computational Intelligence Systems, [s. l.], v. 13, n. 1, p. 1663–1678, 2020a.

PARNIANIFARD, Amir *et al.* Comparative study of metamodeling and sampling design for expensive and semi-expensive simulation models under uncertainty. **Simulation**, [s. l.], v. 96, n. 1, p. 89–110, 2020b.

PARNIANIFARD, Amir *et al.* Expedited surrogate-based quantification of engineering tolerances using a modified polynomial regression. **Structural and Multidisciplinary Optimization**, [s. l.], v. 66, n. 3, 2023a.

PARNIANIFARD, Amir *et al.* Expedited surrogate-based quantification of engineering tolerances using a modified polynomial regression. **Structural and Multidisciplinary Optimization**, [s. l.], v. 66, n. 3, p. 61, 2023b. Disponível em: <https://link.springer.com/10.1007/s00158-023-03493-0>.

PARNIANIFARD, Amir *et al.* Kriging and Latin hypercube sampling assisted simulation optimization in optimal design of PID controller for speed control of DC motor. **Journal of Computational and Theoretical Nanoscience**, [s. l.], v. 15, n. 5, p. 1471–1479, 2018.

PARNIANIFARD, Amir *et al.* New Adaptive Surrogate-Based Approach Combined Swarm Optimizer Assisted Less Tuning Cost of Dynamic Production-Inventory Control System. **IEEE Access**, [s. l.], 2021.

PARNIANIFARD, Amir *et al.* Robust optimal design of FOPID controller for five bar linkage robot in a Cyber-Physical System: A new simulation-optimization approach. **PLoS ONE**, [s. l.], v. 15, n. 11 November, 2020c.

PARNIANIFARD, Amir; AZFANIZAM, Ahmad Siti. Metamodel-based robust simulation-optimization assisted optimal design of multiloop integer and fractional-order PID controller. **International Journal of Numerical Modelling: Electronic Networks, Devices and Fields**, [s. l.], v. 33, n. 1, p. 1–20, 2020.

PASCAL, Demetry; SOLGI, Ryan. **Geneticalgorithm2**. [s. l.]: PyPI.org, 2023.

PEDRIELLI, Giulia; WANG, Songhao; NG, Szu Hui. An extended Two-Stage Sequential Optimization approach: Properties and performance. **European Journal of Operational Research**, [s. l.], v. 287, n. 3, p. 929–945, 2020. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0377221720303982>.

PÉREZ-RODRÍGUEZ, Javier; FERNÁNDEZ-NAVARRO, Francisco; ASHLEY, Thomas. Estimating ensemble weights for bagging regressors based on the mean–variance portfolio framework. **Expert Systems with Applications**, [s. l.], v. 229, p. 120462, 2023.

QIAN, Jiachang *et al.* Optimization design of metamaterial vibration isolator with honeycomb structure based on multi-fidelity surrogate model. **Structural and Multidisciplinary Optimization**, [s. l.], v. 64, n. 1, p. 423–439, 2021.

QUAN, Ning *et al.* Simulation optimization via kriging: A sequential search using expected improvement with computing budget constraints. **IIE Transactions (Institute of Industrial Engineers)**, [s. l.], v. 45, n. 7, p. 763–780, 2015.

RAMOSAJ, Burim; PAULY, Markus. Consistent estimation of residual variance with random forest Out-Of-Bag errors. **Statistics and Probability Letters**, [s. l.], v. 151, p. 49–57, 2019.

RAPONI, Elena *et al.* Methodology for parameter identification on a thermoplastic composite crash absorber by the Sequential Response Surface Method and Efficient Global Optimization. **Composite Structures**, [s. l.], v. 278, n. September, p. 114646, 2021. Disponível em: <https://doi.org/10.1016/j.compstruct.2021.114646>.

RAZALI, Che Munira Che *et al.* Adaptive infill sampling strategy for metamodeling: Challenge and future research directions. **Bulletin of Electrical Engineering and Informatics**, [s. l.], v. 9, n. 5, p. 2020–2029, 2020.

REGIS, Rommel G. Trust regions in Kriging-based optimization with expected improvement. **Engineering Optimization**, [s. l.], v. 48, n. 6, p. 1037–1059, 2016.

RENGANATHAN, S. Ashwin; MAULIK, Romit; AHUJA, Jai. Enhanced data efficiency using deep neural networks and Gaussian processes for aerodynamic design optimization. **Aerospace Science and Technology**, [s. l.], v. 111, p. 106522, 2021. Disponível em: <https://doi.org/10.1016/j.ast.2021.106522>.

RODIČ, Blaž. Industry 4.0 and the New Simulation Modelling Paradigm. **Organizacija**, [s. l.], v. 50, n. 3, p. 193–207, 2017. Disponível em: <https://www.sciendo.com/article/10.1515/orga-2017-0017>.

RODRIGUEZ, Sergio; LUDKOVSKI, Michael. Generalized probabilistic bisection for stochastic root finding. **ACM Transactions on Modeling and Computer Simulation**, [s. l.], v. 30, n. 1, 2020.

ROJAS-GONZALEZ, Sebastian; VAN NIEUWENHUYSE, Inneke. A survey on kriging-based infill algorithms for multiobjective simulation optimization. **Computers and Operations Research**, [s. l.], v. 116, p. 104869, 2020. Disponível em: <https://doi.org/10.1016/j.cor.2019.104869>.

ROSEN, Scott L.; HARMONOSKY, Catherine M. An improved simulated annealing simulation optimization method for discrete parameter stochastic systems. **Computers and Operations Research**, [s. l.], v. 32, n. 2, p. 343–358, 2005.

SAADATPOUR, Motahareh *et al.* Optimization of selective withdrawal systems in hydropower reservoir considering water quality and quantity aspects. **Expert Systems with Applications**, [s. l.], v. 184, 2021.

SCHAPIRE, Robert E. The Strength of Weak Learnability. **Machine Learning**, [s. l.], v. 5, n. 2, p. 197–227, 1990.

SCHONLAU, Matthias; ZOU, Rosie Yuyan. The random forest algorithm for statistical learning. **Stata Journal**, [s. l.], v. 20, n. 1, p. 3–29, 2020.

SCOTT, Warren; FRAZIER, Peter; POWELL, Warren. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. **SIAM Journal on Optimization**, [s. l.], v. 21, n. 3, p. 996–1026, 2011.

SHARIF, Seyed Amirhosain; HAMMAD, Amin. Developing surrogate ANN for selecting near-optimal building energy renovation methods considering energy consumption, LCC and LCA. **Journal of Building Engineering**, [s. l.], v. 25, n. April, p. 100790, 2019. Disponível em: <https://doi.org/10.1016/j.jobe.2019.100790>.

SIEBERS, P. O. *et al.* Discrete-event simulation is dead, long live agent-based simulation!. **Journal of Simulation**, [s. l.], v. 4, n. 3, p. 204–210, 2010.

SOHAIL, Ayesha. **Genetic Algorithms in the Fields of Artificial Intelligence and Data Sciences**. [S. l.]: Springer Science and Business Media Deutschland GmbH, 2023.

SOUSA JUNIOR, Wilson Trigueiro de *et al.* Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. **Computers & Industrial Engineering**, [s. l.], v. 128, n. December 2018, p. 526–540, 2019. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S036083521830682X>.

SOUSA JUNIOR, Wilson Trigueiro *et al.* Economic lot-size using machine learning, parallelism, metaheuristic and simulation. **International Journal of Simulation Modelling**, [s. l.], v. 18, n. 2, p. 205–216, 2019.

SOUSA JUNIOR, Wilson Trigueiro *et al.* Shop floor simulation optimization using machine learning to improve parallel metaheuristics. **Expert Systems with Applications**, [s. l.], 2020.

SRINIVAS, M.; PATNAIK, L.M. Genetic algorithms: a survey. **Computer**, [s. l.], v. 27, n. 6, p. 17–26, 1994. Disponível em: <http://ieeexplore.ieee.org/document/294849/>.

STORTI, Bruno A. *et al.* Improving the efficiency of a Savonius wind turbine by designing a set of deflector plates with a metamodel-based optimization approach. **Energy**, [s. l.], v. 186, p. 115814, 2019. Disponível em: <https://doi.org/10.1016/j.energy.2019.07.144>.

TAGUCHI, Genichi; TSAI, Shih-Chung. Quality Engineering (Taguchi Methods) For The Development Of Electronic Circuit Technology. **IEEE TRANSACTIONS ON RELIABILITY**. [S. l.: s. n.], 1995.

TAHRAOUI, Hichem *et al.* Modeling the organic matter of water using the decision tree coupled with bootstrap aggregated and least-squares boosting. **Environmental Technology and Innovation**, [s. l.], v. 27, 2022.

TALBI, El Ghazali. **Machine Learning into Metaheuristics: A Survey and Taxonomy**. [S. l.]: Association for Computing Machinery, 2021.

TALEB, Mushtaq *et al.* An integrated approach of discrete event simulation and a non-radial super efficiency data envelopment analysis for performance evaluation of an emergency department. **Expert Systems with Applications**, [s. l.], v. 220, 2023.

TAY, Timothy; OSORIO, Carolina. Bayesian optimization techniques for high-dimensional simulation-based transportation problems. **Transportation Research Part B: Methodological**, [s. l.], v. 164, p. 210–243, 2022.

TEIXEIRA, Rui *et al.* Metamodel-based metaheuristics in optimal responsive adaptation and recovery of traffic networks. **Sustainable and Resilient Infrastructure**, [s. l.], v. 7, n. 6, p. 756–774, 2022.

TURRIONI, J.; MELLO, C. Metodologia De Pesquisa Em Engenharia De Produção: Estratégias, Métodos E Técnicas Para Condução De Pesquisas Quantitativas E Qualitativas. **Programa de Pós-graduação em Engenharia de Produção**, [s. l.], v. 1, n. Programa de Pós-graduação em Engenharia de Produção, p. 191, 2012.

UNGREDDA, Juan; PEARCE, Michael; BRANKE, Juergen. Bayesian Optimisation vs. Input Uncertainty Reduction. **ACM Transactions on Modeling and Computer Simulation**, [s. l.], v. 32, n. 3, 2022.

UR REHMAN, Samee; LANGELAAR, Matthijs. Efficient infill sampling for unconstrained robust optimization problems. **Engineering Optimization**, [s. l.], v. 48, n. 8, p. 1313–1332, 2016.

URQUHART, Magnus; LJUNGSKOG, Emil; SEBBEN, Simone. Surrogate-based optimisation using adaptively scaled radial basis functions. **Applied Soft Computing Journal**, [s. l.], v. 88, p. 106050, 2020. Disponível em: <https://doi.org/10.1016/j.asoc.2019.106050>.

VILLARREAL-MARROQUÍN, María G. *et al.* A comparison of two metamodel-based methodologies for multiple criteria simulation optimization using an injection molding case study. **Journal of Polymer Engineering**, [s. l.], v. 33, n. 3, p. 193–209, 2013.

VU, Ky Khac *et al.* Surrogate-based methods for black-box optimization. **International Transactions in Operational Research**, [s. l.], v. 24, n. 3, p. 393–424, 2017.

WANG, Tianxiang *et al.* Efficient estimation of a risk measure requiring two-stage simulation optimization. **European Journal of Operational Research**, [s. l.], v. 305, n. 3, p. 1355–1365, 2023.

WANG, Yi *et al.* Multi-Fidelity Surrogate-Based Optimization for Electromagnetic Simulation Acceleration. **ACM Transactions on Design Automation of Electronic Systems**, [s. l.], v. 25, n. 5, p. 1–21, 2020. Disponível em: <https://dl.acm.org/doi/10.1145/3398268>.

WANG, Lien; DEMEULEMEESTER, Erik. Simulation optimization in healthcare resource planning: A literature review. **IISE Transactions**, [s. l.], 2022.

WANG, Zilong; IERAPETRITOU, Marianthi. A Novel Surrogate-Based Optimization Method for Black-Box Simulation with Heteroscedastic Noise. **Industrial and Engineering Chemistry Research**, [s. l.], v. 56, n. 38, p. 10720–10732, 2017.

WANG, Zilong; IERAPETRITOU, Marianthi. Constrained optimization of black-box stochastic systems using a novel feasibility enhanced Kriging-based method. **Computers and Chemical Engineering**, [s. l.], v. 118, p. 210–223, 2018.

WANG, Haowei; YUAN, Jun; NG, Szu Hui. Gaussian process based optimization algorithms with input uncertainty. **IISE Transactions**, [s. l.], v. 52, n. 4, p. 377–393, 2020. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/24725854.2019.1639859>.

WOLLMANN, Dewey; TORTATO, Ubiratã. Proposal for a model to hierarchize strategic decisions according to criteria of value innovation, sustainability and budgetary constraint. **Journal of Cleaner Production**, [s. l.], v. 231, p. 278–289, 2019.

XIA, Yufei *et al.* A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. **Expert Systems with Applications**, [s. l.], v. 78, p. 225–241, 2017. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2017.02.017>.

XIAO, Hui; LEE, Loo Hay; CHEN, Chun Hung. Optimal budget allocation rule for simulation optimization using quadratic regression in partitioned domains. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, [s. l.], v. 45, n. 7, p. 1047–1062, 2015.

XIAO, Yu; WATSON, Maria. Guidance on Conducting a Systematic Literature Review. **Journal of Planning Education and Research**, [s. l.], v. 39, n. 1, p. 93–112, 2019.

XIE, Jing; FRAZIER, Peter I.; CHICK, Stephen E. Bayesian optimization via simulation with pairwise sampling and correlated prior beliefs. **Operations Research**, [s. l.], v. 64, n. 2, p. 542–559, 2016.

XIE, Wei; YI, Yuan; ZHENG, Hua. Global-local Metamodel-assisted Stochastic Programming via Simulation. **ACM Transactions on Modeling and Computer Simulation**, [s. l.], v. 31, n. 1, 2021.

XIONG, Zhipeng *et al.* Distillation process optimization: A screening-clustering assisted kriging optimization method. **Chemical Engineering Science**, [s. l.], v. 238, 2021.

YANG, Li; SHAMI, Abdallah. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, [s. l.], v. 415, p. 295–316, 2020.

YAOHUI, Li. A Kriging-based global optimization method using multi-points infill search criterion. **Journal of Algorithms and Computational Technology**, [s. l.], v. 11, n. 4, p. 366–377, 2017.

YIN, Ruyang; LIU, Zhiyuan; ZHENG, Nan. A Simulation-Based Model for Continuous Network Design Problem Using Bayesian Optimization. **IEEE Transactions on Intelligent Transportation Systems**, [s. l.], v. 23, n. 11, p. 20352–20367, 2022.

YOSEFIPOOR, Parisa *et al.* An adaptive surrogate-based, multi-pollutant, and multi-objective optimization for river-reservoir system management. **Ecological Engineering**, [s. l.], v. 175, 2022.

ZHAN, Dawei *et al.* A cooperative approach to efficient global optimization. **Journal of Global Optimization**, [s. l.], 2023.

ZHANG, Shuangsheng *et al.* A construction strategy for conservative adaptive Kriging surrogate model with application in the optimal design of contaminated groundwater extraction-treatment. **Environmental Science and Pollution Research**, [s. l.], v. 29, n. 28, p. 42792–42808, 2022a.

ZHANG, Shuangsheng *et al.* An Adaptive Dynamic Kriging Surrogate Model for Application to the Optimal Remediation of Contaminated Groundwater. **Water Resources Management**, [s. l.], v. 36, n. 13, p. 5011–5032, 2022b.

ZHANG, Jianxia *et al.* Estimation of the Pareto front in stochastic simulation through stochastic Kriging. **Simulation Modelling Practice and Theory**, [s. l.], v. 79, p. 69–86, 2017.

ZHANG, Y *et al.* Injection Molding Warpage Optimization Based on a Mode-Pursuing Sampling Method Injection Molding Warpage Optimization Based on a Mode-Pursuing Sampling Method. **Polymer-Plastics Technology and Engineering**, [s. l.], v. 2559, 2009.

ZHAO, Tingting; MINSKER, Barbara. Efficient Metamodel Approach to Handling Constraints in Nonlinear Optimization for Drought Management. **Journal of Water Resources Planning and Management**, [s. l.], v. 147, n. 12, 2021.

ZHENG, Liang *et al.* Network-wide signal timing stochastic simulation optimization with environmental concerns. **Applied Soft Computing Journal**, [s. l.], v. 77, p. 678–687, 2019.

ZHENG, Liang; LI, Xiaoru. Simulation-based optimization method for arterial signal control considering traffic safety and efficiency under uncertainties. **Computer-Aided Civil and Infrastructure Engineering**, [s. l.], v. 38, n. 5, p. 640–659, 2023.

ZHONG, Shaopeng *et al.* Active learning for multi-objective optimal road congestion pricing considering negative land use effect. **Transportation Research Part C: Emerging Technologies**, [s. l.], v. 125, 2021.

ZHOU, Jian; TURNG, Lih Sheng. Process optimization of injection molding using an adaptive surrogate model with Gaussian process approach. **Polymer Engineering and Science**, [s. l.], v. 47, n. 5, p. 684–694, 2007.

ANEXO A – Detalhamento da Aplicação do *Framework*

A1. Apresentação do Problema

Conforme apresentado na Seção 6.1, o OE1 representa uma fábrica têxtil cujo objetivo é a alocação dos recursos fabris (operadores e máquinas) para o atendimento da demanda semanal da empresa. Este problema possui oito variáveis de decisão conforme disposto na Tabela A1.

Tabela A1: Variáveis de decisão e limites (OE1)

Variáveis	Descrição	Limite Inf. (\mathbf{l})	Limite Sup. (\mathbf{u})	Custo (\mathbf{c})	
x_1	Operadores GrupoA	2	10	R\$	800,00
x_2	Operadores GrupoB	2	10	R\$	800,00
x_3	Máquinas Reta	1	3	R\$	500,00
x_4	Máquina Overlook	2	10	R\$	500,00
x_5	Máquina Debrum	2	10	R\$	500,00
x_6	Estação de Arremate	2	10	R\$	300,00
x_7	Estação de Bainha	2	10	R\$	350,00
x_8	Estação de Passadoria	2	10	R\$	300,00

O objetivo é encontrar uma solução que maximize a função lucro da fábrica, que é dado pela Eq. (A1)(15).

$$L(\mathbf{x}) = \psi v - \sum_{i=1}^8 x_i c_i \quad (\text{A1})$$

Na Eq. (15), $L(\mathbf{x})$ representa o lucro, ψ é a produção semanal alcançada pelo modelo e v é a margem de contribuição média dos produtos. A função objetivo compreende a maximização da expectativa para a função lucro, $\max f(\mathbf{x}) = \mathbb{E}_\omega\{\tilde{\phi}[L(\mathbf{x}, \omega)]\}$, que pode ser obtida pela média de $L(\mathbf{x})$ das replicações do modelo de simulação ($\tilde{\phi}$), sujeitas a sementes aleatórias ω distintas. Para este modelo foram utilizadas 15 replicações (quantidade mínima de replicações considerando a variabilidade do modelo). O modelo matemático que representa o OE1 é dado pela maximização da função Eq.(A2), sujeito as restrições (Eq.(A3) e Eq.(A4)) referentes ao espaço de busca.

$$\max f(\mathbf{x}) = \mathbb{E}_\omega\{\tilde{\phi}[L(\mathbf{x}, \omega)]\} \quad (\text{A2})$$

Sujeito a:

$$l_i \leq x_i \leq u_i \quad (\text{A3})$$

$$x_i \in \mathbb{Z}, \quad \forall i \in \{1, \dots, 8\} \quad (\text{A4})$$

Vale ressaltar que o valor da FO é obtido apartir da produção encontrada (e consequentemente o valor de $L(\mathbf{x})$) pelo modelo de SED (representado por $\tilde{\phi}$), que compreende diversas restrições operacionais, estocasticidade, interdependências e complexidades não esboçadas no modelo matemático.

Uma vez definido o modelo matemático do problema e em posse do modelo de simulação, que tem como *inputs* o valor das variáveis de decisão (x_1 a x_8) e *output* o valor estimado para o Lucro ($\mathbb{E}_\omega\{\tilde{\phi}[L(\mathbf{x}, \omega)]\}$), pode-se aplicar o *framework* AMSO para otimizar este problema. A Seção A2 apresenta o passo a passo da aplicação do AMSO para o OE1.

A2. Aplicação do *Framework*

Conforme detalhado no Capítulo 5, o AMSO possui duas macro fases (inicialização e adaptativa) divididas em 7 etapas. Nesta seção será apresentado os resultados obtidos em cada etapa do *framework*. A Figura A1 apresenta estas etapas do *framework* proposto.

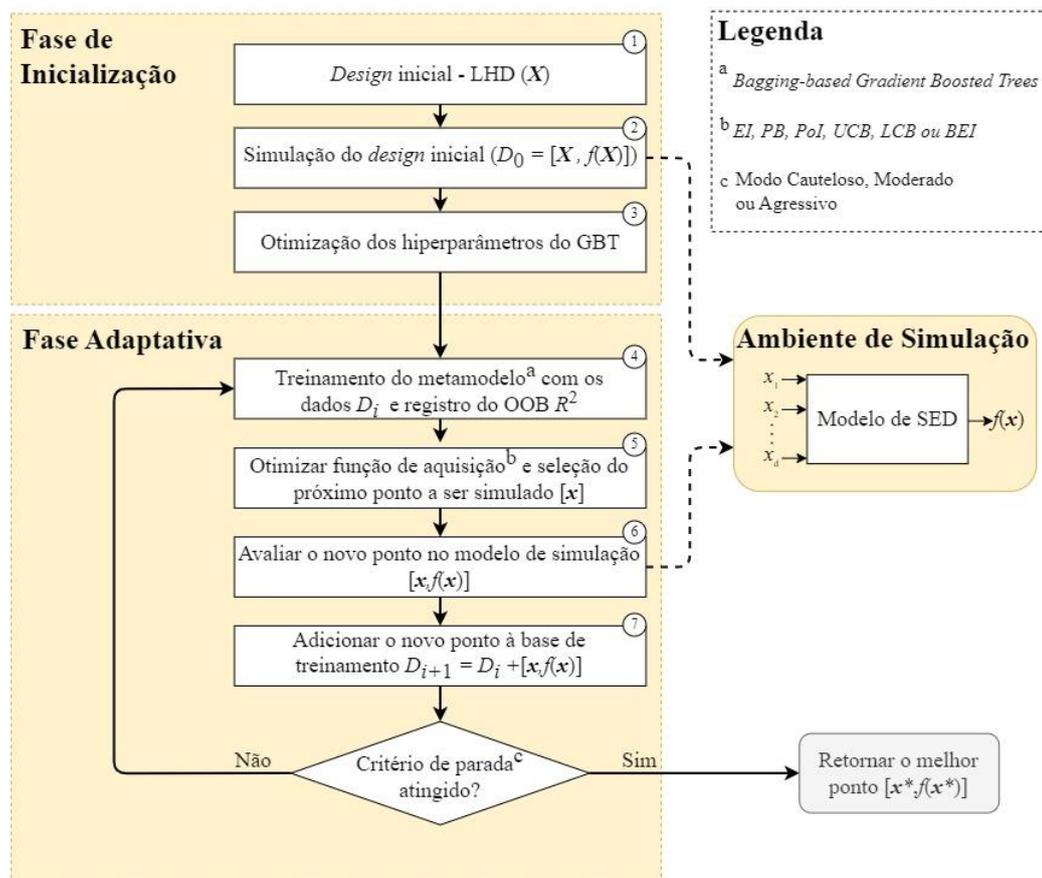


Figura A1: *Framework Adaptive Metamodeling-based Simulation Optimization*

A2.1. Fase de Inicialização

Parâmetros de entrada (definidos pelo usuário):

- Tamanho do arranjo inicial: 40 experimentos;
- Número de iterações na fase adaptativa: 40 iterações;
- Número de replicações: 15;
- Função de aquisição: BEI;
- Critério de parada (Modo): Cauteloso;
- Limites das variáveis de decisão: $l = [2 \ 2 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2]^T$ e $u = [10 \ 10 \ 3 \ 10 \ 10 \ 10 \ 10 \ 10]^T$;
- Tipo das variáveis de decisão: Todas inteiras;
- Modelo de simulação: OE1.fsm;
- Sentido da otimização: Maximização.

Etapa 1: *Design Inicial*

Nesta etapa foi delineado, pelo método LHD, o arranjo inicial a ser simulado. A Tabela A2 apresenta o arranjo gerado.

Tabela A2: *Design Inicial*

N	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	4	6	2	4	8	10	7	4
2	6	9	2	3	8	5	10	6
3	7	6	2	3	9	4	4	2
4	3	6	1	10	4	9	6	5
5	10	8	2	3	8	9	3	5
6	5	3	3	3	6	8	3	7
7	3	8	2	9	5	3	7	2
8	10	5	3	3	3	7	8	6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
37	5	5	2	8	9	2	3	7
38	3	7	2	4	2	8	9	8
39	7	2	2	6	4	6	9	2
40	9	5	2	10	10	5	7	3

Etapa 2: Simulação do *Design Inicial*

Na segunda etapa o *design* inicial gerado anteriormente é imputado no modelo e simulado, conforme esquemático da Figura A2, e os resultados são armazenados no ambiente de otimização em *Python*. Os resultados da simulação estão dispostos na Tabela A3.

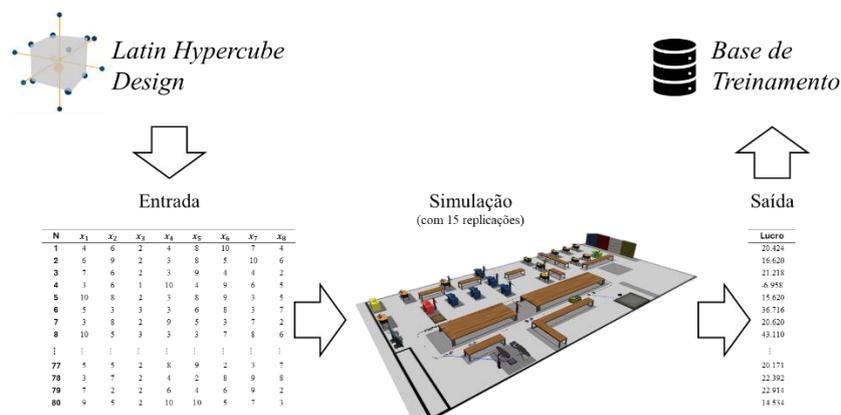


Figura A2: Esquemático para geração da base de treinamento

Tabela A3: *Design* inicial com resultado da simulação

N	X								Y
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Lucro
1	4	6	2	4	8	10	7	4	20.424
2	6	9	2	3	8	5	10	6	16.620
3	7	6	2	3	9	4	4	2	21.218
4	3	6	1	10	4	9	6	5	-6.958

5	10	8	2	3	8	9	3	5	15.620
6	5	3	3	3	6	8	3	7	36.716
7	3	8	2	9	5	3	7	2	20.620
8	10	5	3	3	3	7	8	6	43.110
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
37	5	5	2	8	9	2	3	7	20.171
38	3	7	2	4	2	8	9	8	22.392
39	7	2	2	6	4	6	9	2	22.914
40	9	5	2	10	10	5	7	3	14.534

Etapa 3: Otimização dos Hiperparâmetros

Nesta etapa os hiperparâmetros do GBT serão selecionados, via otimização com AG, para serem utilizados nas demais etapas do *framework*. Os hiperparâmetros do GBT adotados e sua amplitude foram: número de estimadores (1 – 100); taxa de aprendizagem (0,05 – 0,30); e amostras mínimas por folha (0,12 – 0,50).

Conforme discutido no Capítulo 5, a base de dados de entrada (\mathbf{X} e \mathbf{Y}) foi normalizada para média 0 e desvio padrão igual a 1. Na sequência, foi aplicado o AG com *cross-validation* (5-folds) para minimizar o erro do metamodelo. A que o AG retornou após 10 gerações foi número de estimadores igual a 79, taxa de aprendizado igual a 0,098 e amostras mínimas por folha igual a 2. Esta configuração resultou em um R^2 de 0,91, considerado um bom ajuste inicial para o metamodelo. Portanto, esta será a configuração do GBT que será utilizado como *base-learner* no treinamento do metamodelo *Bagging* na fase adaptativa.

A2.2. Fase Adaptativa

Nesta fase o algoritmo executará recursivamente as Etapas 4 a 7, até que o critério de parada seja satisfeito (número máximo de iterações neste caso). Nesta seção será apresentado, para fins explicativos, apenas os resultados das primeiras e últimas iterações do algoritmo.

Iteração 1

Etapa 4: Treinamento do Metamodelo

O metamodelo Bagged-Gradient Boosted Trees (B-GBT) foi treinado com a base de treinamento inicial normalizada de tamanho $\xi_0 = 40$. Após o treinamento, o metamodelo exibiu um $R^2 = 0,92$ e está pronto para realizar previsões sobre o problema de OvS estudado.

Etapa 5: Otimização da Função de Aquisição

Como nesta aplicação está sendo utilizada a função BEI, o primeiro passo é definição do critério de busca λ . Este critério é baseado no valor obtido do sorteio aleatório

$\rho \sim \text{Uniform}(0,85; 1,00)$, que nesta iteração resultou em $\rho = 0,89$. Como $\rho \leq R^2$, o valor de λ é igual a 1, ou seja, nesta iteração o algoritmo pesará por uma busca exploratória. Logo, a função BEI a ser otimizada pelo AG será representada pelo segundo termo da Eq.(A5).

$$BEI(\mathbf{x}) = (0)\sigma\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma}\right) + (1)\hat{f}(\mathbf{x}) \quad (A5)$$

É importante ressaltar que, o padrão da biblioteca *geneticalgorithm2* é minimização, foi requisitado que o AG minimizasse o negativo da função BEI ($-BEI(\mathbf{x})$). O AG convergiu após 59 gerações e retornou a solução $\mathbf{x} = [6 \ 6 \ 3 \ 3 \ 2 \ 3 \ 4 \ 6]^T$, com um valor da função BEI igual a 43.240,83. O gráfico de convergência é apresentado pela Figura A3.

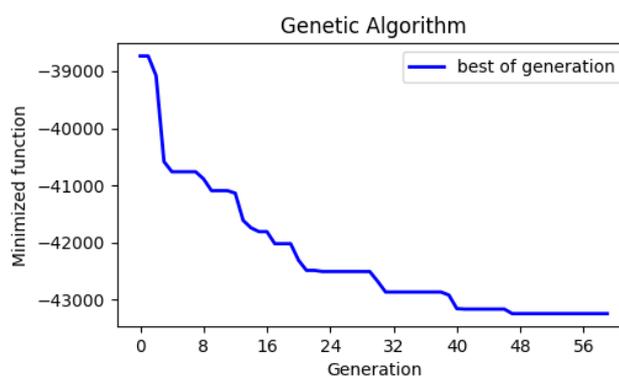


Figura A3: Gráfico de convergência do AG para a iteração 1.

Etapa 6: Avaliar Solução no Modelo de Simulação

Nesta Etapa, a solução $\mathbf{x} = [6 \ 6 \ 3 \ 3 \ 2 \ 3 \ 4 \ 6]^T$ foi imputada no modelo de simulação, o qual foi executado com 15 replicações, e retornou um valor de lucro médio ($f(\mathbf{x})$) igual a R\$48.680,80.

Etapa 7: Incremento da base de treinamento

Esta nova solução $\mathbf{x} = [6 \ 6 \ 3 \ 3 \ 2 \ 3 \ 4 \ 6]^T$, com seu respectivo valor de $f(\mathbf{x}) = 48.680,80$, foram incorporados à base de treinamento inicial (\mathbf{D}_0) formando a nova base \mathbf{D}_1 , com tamanho $\xi_1 = 41$ experimentos.

Critério de Parada

Como o número de iterações até o momento é menor que o definido pelo usuário, o algoritmo segue para a próxima iteração.

Iteração 2

Etapa 4: Treinamento do Metamodelo

O metamodelo B-GBT foi treinado com a base de treinamento inicial normalizada de tamanho $\xi_1 = 41$. Após o treinamento, o metamodelo exibiu um $R^2 = 0,93$ e está pronto para realizar previsões sobre o problema de OvS estudado.

Etapa 5: Otimização da Função de Aquisição

Considerando o valor sorteado de $\rho = 0,86$ e como $\rho \leq R^2$, o valor de λ é novamente igual a 1(explotatória). Logo, a função BEI a ser otimizada pelo AG será representada pelo segundo termo da Eq.(A6).

$$BEI(\mathbf{x}) = (0)\sigma\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma}\right) + (1)\hat{f}(\mathbf{x}) \quad (A6)$$

O AG convergiu após 26 gerações e retornou a solução $\mathbf{x} = [6 \ 5 \ 3 \ 2 \ 2 \ 2 \ 4 \ 7]^T$, com um valor da função BEI igual a 46.007,12. O gráfico de convergência é apresentado pela Figura A4.

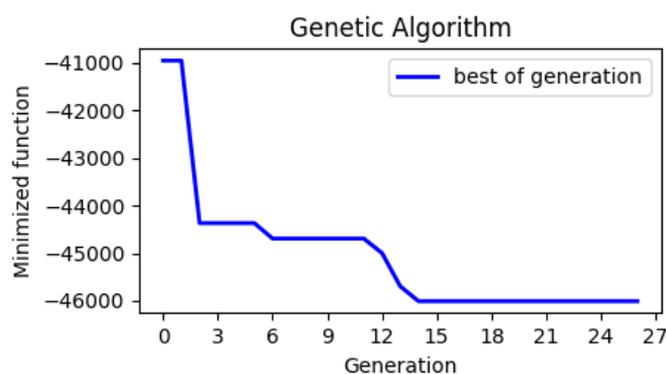


Figura A4: Gráfico de convergência do AG para a iteração 2.

Etapa 6: Avaliar Solução no Modelo de Simulação

Nesta Etapa, a solução $\mathbf{x} = [6 \ 5 \ 3 \ 2 \ 2 \ 2 \ 4 \ 7]^T$ foi imputada no modelo de simulação, o qual foi executado com 15 replicações, e retornou um valor de lucro médio ($f(\mathbf{x})$) igual a R\$50.316,80.

Etapa 7: Incremento da base de treinamento

Esta nova solução $\mathbf{x} = [6 \ 5 \ 3 \ 2 \ 2 \ 2 \ 4 \ 7]^T$, com seu respectivo valor de $f(\mathbf{x}) = 50.316,80$, foram incorporados à base de treinamento (\mathbf{D}_1) formando a nova base \mathbf{D}_2 , com tamanho $\xi_2 = 42$ experimentos.

Critério de Parada

Como o número de iterações até o momento é menor que o definido pelo usuário, o algoritmo segue para a próxima iteração.

Iteração 3

Etapa 4: Treinamento do Metamodelo

O metamodelo B-GBT foi treinado com a base de treinamento inicial normalizada de tamanho $\xi_2 = 42$. Após o treinamento, o metamodelo exibiu um $R^2 = 0,92$ e está pronto para realizar previsões sobre o problema de OvS estudado.

Etapa 5: Otimização da Função de Aquisição

Considerando o valor sorteado de $\rho = 0,95$ e como $\rho \geq R^2$, o valor de λ é igual a 0 (busca exploratória). Logo, a função BEI a ser otimizada pelo AG será representada pelo segundo termo da Eq.(A7).

$$BEI(\mathbf{x}) = (1)\sigma\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma}\right) + (0)\hat{f}(\mathbf{x}) \quad (A7)$$

O AG convergiu após 31 gerações e retornou a solução $\mathbf{x} = [6 \ 5 \ 3 \ 2 \ 2 \ 2 \ 2 \ 9]^T$, com um valor da função BEI igual a 688,82. O gráfico de convergência é apresentado pela Figura A5.

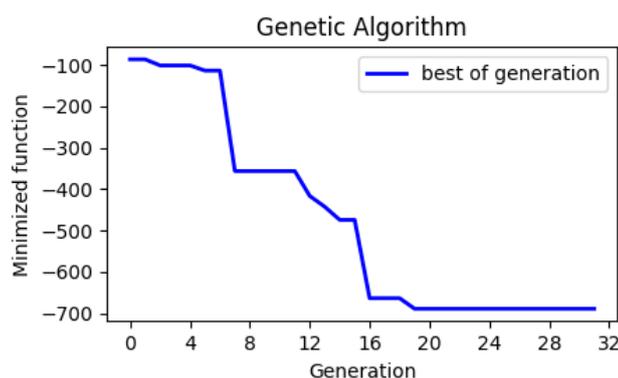


Figura A5: Gráfico de convergência do AG para a iteração 3.

Etapa 6: Avaliar Solução no Modelo de Simulação

Nesta Etapa, a solução $\mathbf{x} = [6 \ 5 \ 3 \ 2 \ 2 \ 2 \ 2 \ 9]^T$ foi imputada no modelo de simulação, o qual foi executado com 15 replicações, e retornou um valor de lucro médio ($f(\mathbf{x})$) igual a R\$50.468,00.

Etapa 7: Incremento da base de treinamento

Esta nova solução $\mathbf{x} = [6 \ 5 \ 3 \ 2 \ 2 \ 2 \ 2 \ 9]^T$, com seu respectivo valor de $f(\mathbf{x}) = 50.468,00$, foram incorporados à base de treinamento (\mathbf{D}_2) formando a nova base \mathbf{D}_3 , com tamanho $\xi_3 = 43$ experimentos.

Critério de Parada

Como o número de iterações até o momento é menor que o definido pelo usuário, o algoritmo segue para a próxima iteração.

Iteração 4 a 39

Para fins didáticos e espaço não serão apresentados os dados das iterações 4 a 39.

Iteração 40

Etapa 4: Treinamento do Metamodelo

O metamodelo B-GBT foi treinado com a base de treinamento inicial normalizada de tamanho $\xi_{39} = 79$. Após o treinamento, o metamodelo exibiu um $R^2 = 0,98$ e está pronto para realizar previsões sobre o problema de OvS estudado.

Etapa 5: Otimização da Função de Aquisição

Considerando o valor sorteado de $\rho = 0,91$ e como $\rho \leq R^2$, o valor de λ é igual a 1 (busca exploratória). Logo, a função BEI a ser otimizada pelo AG será representada pelo segundo termo da Eq.(A8).

$$BEI(\mathbf{x}) = (0)\sigma\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{\sigma}\right) + (1)\hat{f}(\mathbf{x}) \quad (A8)$$

O AG convergiu após 64 gerações e retornou a solução $\mathbf{x} = [3 \ 9 \ 3 \ 2 \ 3 \ 3 \ 2 \ 2]^T$, com um valor da função BEI igual a 50.359,49. O gráfico de convergência é apresentado pela Figura A6.

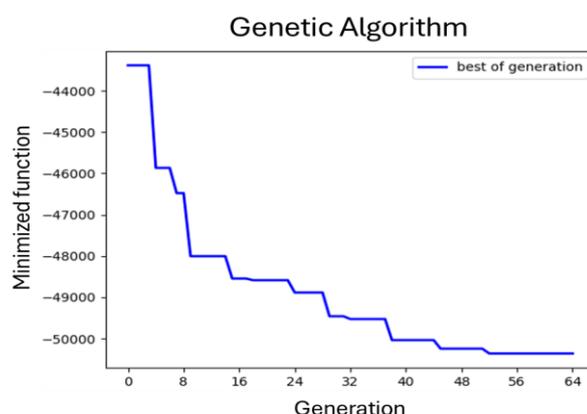


Figura A6: Gráfico de convergência do AG para a iteração 40.

Etapa 6: Avaliar Solução no Modelo de Simulação

Nesta Etapa, a solução $\mathbf{x} = [3 \ 9 \ 3 \ 2 \ 3 \ 3 \ 2 \ 2]^T$ foi imputada no modelo de simulação, o qual foi executado com 15 replicações, e retornou um valor de lucro médio ($f(\mathbf{x})$) igual a R\$50.609,60.

Etapa 7: Incremento da base de treinamento

Esta nova solução $\mathbf{x} = [3 \ 9 \ 3 \ 2 \ 3 \ 3 \ 2 \ 2]^T$, com seu respectivo valor de $f(\mathbf{x}) = 50.609,60$, foram incorporados à base de treinamento (\mathbf{D}_{39}) formando a nova base \mathbf{D}_{40} , com tamanho $\xi_{40} = 80$ experimentos.

Critério de Parada

Como o número de iterações é igual ao número máximo de iterações definido pelo usuário, o critério de parada foi satisfeito e o algoritmo é finalizado.

Após a finalização do algoritmo, o *framework* retorna como resultado a melhor solução encontrada na base de treinamento \mathbf{D}_{40} , ou seja, o $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in X} f(\mathbf{x})$. Neste sentido, a melhor solução encontrada foi $\mathbf{x}^* = [3 \ 9 \ 3 \ 2 \ 2 \ 2 \ 2 \ 2]^T$, obtendo um lucro de R\$ 51.915,20. Em termos práticos, o AMSO sugere que sejam alocados na fábrica um total de três operadores para a atividade do grupo A, nove operadores para o grupo B, três máquinas retas, duas máquinas overlook, duas estações de arremate, duas estações de bainha e duas estações de passadoria.

A Tabela A4 sintetiza a matriz de dados \mathbf{D}_{40} , composta pelos 80 experimentos realizados pelo *framework* (40 da fase de inicialização e 40 da fase adaptativa).

Tabela A4: Base de treinamento final

N	X								Y
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Lucro
1	4	6	2	4	8	10	7	4	20.424
2	6	9	2	3	8	5	10	6	16.620
3	7	6	2	3	9	4	4	2	21.218
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
71	3	6	3	2	2	2	2	4	50.672
72	3	10	3	2	2	3	3	2	50.542
73	3	9	3	2	2	2	2	2	51.915
74	3	9	3	2	2	2	2	6	50.763
75	4	9	3	2	2	10	3	2	48.407
76	2	4	3	2	2	2	2	2	32.389
77	4	10	3	2	3	2	2	3	49.253
78	4	10	3	3	2	3	2	3	48.754
79	2	9	3	3	2	2	4	2	50.674
80	3	9	3	2	3	3	2	2	50.610

APÊNDICE A - Publicações

i. Periódicos

Publicados/aceitos

AMARAL, J. V. S. Do *et al.* Adaptive metamodeling simulation optimization: Insights, challenges, and perspectives. **Applied Soft Computing**, v. 165, 2024.

AMARAL, João Victor Soares do *et al.* Adaptive metamodeling-based simulation optimisation. **Journal of the Operational Research Society**, 2024.

BATISTA, Ricardo Henrique *et al.* An integrated approach using discrete event simulation and design of experiments for working capital management in the fuel retail trade. **Journal of Simulation**, p. 1-14, 2024.

AMARAL, J.V.S Do *et al.* Energy Digital Twin applications: a review. **Renewable and Sustainable Energy Reviews**. 2023.

DOS SANTOS, C. H. *et al.* Simulation-based digital twins monitoring: an approach focused on models' accreditation. **The International Journal of Advanced Manufacturing Technology**, v. 124, n. 7–8, p. 2423–2435, 13 fev. 2023.

AMARAL, J. V. S. Do *et al.* Data envelopment analysis for algorithm efficiency assessment in metamodel-based simulation optimization. **The International Journal of Advanced Manufacturing Technology**, v. 121, n. 11–12, p. 7493–7507, 1 ago. 2022.

AMARAL, J. V. S. Do *et al.* Metamodeling-based simulation optimization in manufacturing problems: a comparative study. **International Journal of Advanced Manufacturing Technology**, v. 120, n. 7–8, p. 5205–5224, 1 jun. 2022.

AMARAL, J. V. S. Do *et al.* Metamodel-based simulation optimization: A systematic literature review. **Simulation Modelling Practice and Theory**, v. 114, 1 jan. 2022.

DOS SANTOS, C. H. *et al.* Decision-making in a fast fashion company in the Industry 4.0 era: a Digital Twin proposal to support operational planning. **The International Journal of Advanced Manufacturing Technology**, v. 116, n. 5–6, p. 1653–1666, 2 set. 2021.

VILELA, F. F. *et al.* Economic feasibility analysis for co-generation of power by diesel generator at peak time: Monte carlo simulation approach in a hospital. **Acta Scientiarum - Technology**, v. 43, 20 ago. 2021.

ii. Capítulo de Livro

AMARAL, J. V. S. DO; MONTEVECHI, J. A. B.; MIRANDA, R. DE C. Metamodel-Based Simulation Optimization: A Systematic Literature Review. Em: **Uncertainty Modeling: Fundamental Concepts and Models**. Biblioteca Central da Universidade de Brasília, 2022. p. 119–171.

iii. Congressos

Publicados/aceitos

AMARAL, J. V. S. DO. *et al.* **Metamodelagem adaptativa: uma aplicação na indústria têxtil.** Simpósio Brasileiro de Pesquisa Operacional 2024. **Anais...** Fortaleza, Brasil: 2024.

MONTEVECHI, J. A. B. *et al.* **Ensemble-based infill search simulation optimization framework.** Proceeding of 2023 Winter Simulation Conference. **Anais...** San Antonio, USA: 2023.

AMARAL, J. V. S. DO. *et al.* **Metamodelagem adaptativa em otimização via simulação.** Encontro Nacional de Engenharia de Produção 2023. **Anais...** Fortaleza, Brasil: 2023.

LOPES, G. F. *et al.* **Redução do espaço de busca em problemas de Otimização via Simulação Multiobjetivo.** Encontro Nacional de Engenharia de Produção 2023. **Anais...** Fortaleza, Brasil: 2023.

PEREIRA, A. B. M. *et al.* **Gêmeos digitais para apoio a tomada de decisões na logística reversa.** Encontro Nacional de Engenharia de Produção 2023. **Anais...** Fortaleza, Brasil: 2023.

AMARAL J. V. S. DO. *et al.* **Metamodelagem adaptativa para otimização via simulação a eventos discretos.** Simpósio Brasileiro de Pesquisa Operacional 2023. **Anais...** São José dos Campos, Brasil: 2023.

AMARAL, J. V. S. DO. *et al.* **Redes neurais para otimização via simulação por metamodelagem: uma aplicação no setor industrial.** Encontro Nacional de Engenharia de Produção 2022. **Anais...** Foz do Iguaçu, Brasil: 2022.

RIBEIRO, E. C. *et al.* **Otimização de uma carteira de crédito utilizando programação linear e análise de sensibilidade.** Encontro Nacional de Engenharia de Produção 2022. **Anais...** Foz do Iguaçu, Brasil: 2022.

LIMA, Y. T. *et al.* **Desempenho Humano em Simulações: Uma Análise Sistemática da Literatura.** Encontro Nacional de Engenharia de Produção 2022. **Anais...** Foz do Iguaçu Brasil: 2022.

SILVA, C. M. P. *et al.* **Aplicação de Simulação Computacional para tomada de decisão em um Laboratório de Análises Clínicas do município de Itajubá, MG.** Encontro Nacional de Engenharia de Produção 2022. **Anais...** Foz do Iguaçu Brasil: 2022.

PINTO, W. G. M. *et al.* **Otimização via simulação aplicado a logística reversa: Uma revisão sistemática de literatura.** Encontro Nacional de Engenharia de Produção 2022. **Anais...** Foz do Iguaçu Brasil: 2022.

MONTEVECHI, A. A. *et al.* **Machine learning na análise do risco de crédito: uma aplicação da regressão logística e algoritmos de resampling.** Encontro Nacional de Engenharia de Produção 2022. **Anais...** Foz do Iguaçu, Brasil: 2022.

MARINS F. *et al.* **Network new multiple criteria data envelopment analysis model: an application on educational case.** Simpósio Brasileiro de Pesquisa Operacional 2022. **Anais...** Juiz de Fora, Brasil: 2022.

AMARAL J. V. S. DO. *et al.* **Comparação entre estratégias de metamodelagem através da análise envoltória de dados.** Simpósio Brasileiro de Pesquisa Operacional 2022. **Anais...** Juiz de Fora, Brasil: 2022.

MONTEVECHI A. A. *et al.* **Modelagem de um classificador SVM para análise de risco de crédito.** Simpósio Brasileiro de Pesquisa Operacional 2022. **Anais...** Juiz de Fora, Brasil: 2022.

AMARAL, J. V. S. DO. *et al.* **Otimização Baseada em Metamodelos: Uma Abordagem para Metamodelagem em Simulação a Eventos Discretos.** Encontro Nacional de Engenharia de Produção 2021. **Anais...** Online: 2021.

GABRIEL G. T. *et al.* **Discrete event simulation to improve an ambulatory endoscopy center in the covid-19 pandemic.** Simpósio Brasileiro de Pesquisa Operacional 2021. **Anais...** Online: 2021.

AMARAL, J. V. S. DO *et al.* **Framework para otimização via simulação integrando metamodelagem, doe e otimização de hiper-parâmetros.** Simpósio Brasileiro de Pesquisa Operacional 2021. **Anais...** Online: 2021.