FRAMEWORK PARA OTIMIZAÇÃO VIA SIMULAÇÃO INTEGRANDO METAMODELAGEM, DOE E OTIMIZAÇÃO DE HIPER-PARÂMETROS

João Victor Soares do Amaral
Universidade Federal de Itajubá (UNIFEI)
Av. B P S, 1303 - Pinheirinho, Itajubá - MG, 37500-903
joao.amaral@unifei.edu.br

José Arnaldo Barra Montevechi
Universidade Federal de Itajubá (UNIFEI)
Av. B P S, 1303 - Pinheirinho, Itajubá - MG, 37500-903
montevechi@unifei.edu.br

Rafael de Carvalho Miranda
Universidade Federal de Itajubá (UNIFEI)
Av. B P S, 1303 - Pinheirinho, Itajubá - MG, 37500-903
rafael.miranda@unifei.edu.br

RESUMO

Com os desenvolvimentos na área de *machine learning* surgiram algoritmos com alta capacidade de aprendizado, tornando o uso das técnicas de otimização via simulação por metamodelagem (OvSM) para a solução de problemas complexos um campo de estudo promissor. Neste sentido, o presente estudo propõe um *framework* para OvSM que incorpora: algoritmos de *machine learning*, otimização de hiper-parâmetros via Algoritmo Genético e *Design of Experiments.* A fim de validar o *framework* proposto, este artigo testou seis algoritmos de *machine learning*, com e sem a etapa de otimização de seus hiper-parâmetros, além de considerar dois arranjos experimentais para geração de dados para metamodelagem. O *framework* proposto foi aplicado a problemas de alocação de recursos em três casos reais da indústria. A aplicação do *framework* obteve soluções que atingiram, respectivamente, 100%, 96,17%, e 100% do ótimo local *benchmark*, demandando, em média, 35,22% menos tempo. Além disso, a incorporação da etapa de otimização de hiper-parâmetros permitiu uma redução de 31,28% no erro dos metamodelos.

PALAVRAS-CHAVE. Otimização. Metamodelagem. Simulação a Eventos Discretos.

Tópicos: Simulação, Inteligência Computacional.

ABSTRACT

In recent years, the development in the machine learning area has emerged algorithms with high learning capacity, making the use of metamodel-based simulation optimization (MBSO) to solve complex problems a promising field of study. In this sense, the present study proposes a framework for MBSO that incorporates machine learning algorithms, hyper-parameter optimization via Genetic Algorithm and Design of Experiments. To validate the proposed framework, this paper tested six machine learning algorithms with and without the hyper-parameter optimization step in two experimental arrangements applied to the resource allocation problem in three real cases of industry. The application of the framework obtained solutions that reached, respectively, 100%, 96.17%, and 100% of the local optimum benchmark, demanding, on average, 35.22% less time. In addition, the incorporation of the hyper-parameter optimization step allowed a 31.28% reduction in the metamodels' error.

KEYWORDS. Optimization. Metamodeling. Discrete-event Simulation.
Paper topics: Simulation, Computational Intelligence.

1. Introdução

O campo da Otimização via Simulação (OvS) progrediu significativamente nas últimas década, com o desenvolvimento de novos algoritmos, implementações e aplicativos. Significativas contribuições para este campo de pesquisa tem surgido de pesquisadores e profissionais, principalmente nas áreas de engenharia, operações industriais, programação matemática, estatística e aprendizado de máquina, bem como nas comunidades de ciência da computação [Amaran et al. 2016].

Segundo Fu [2002], o processo de OvS é compreendido em duas etapas conduzidas recursivamente. Na primeira, o módulo de otimização indica possíveis soluções para o problema, e na segunda etapa, o modelo de simulação as avalia. Todavia, dependendo da complexidade dos modelos de simulação, estes podem demandar minutos a horas para avaliar uma única solução, tornando o processo de OvS lento e, muitas vezes, impraticável [Cai et al. 2019].

Atualmente existem diversos métodos que visam resolver problemas de OvS, dentre eles pode-se citar as metaheurísticas, *Ranking and Selection*, métodos gradientes, busca randômica, e metamodelagem, sendo esta última o foco deste estudo. A metamodelagem objetiva identificar e estimar a relação entre as entradas e saídas do modelo de simulação, na forma de uma função matemática determinística, a qual é utilizada para avaliar as possíveis soluções no processo de otimização, reduzindo, consideravelmente, o tempo dispendido na OvS [Barton e Meckesheimer 2006], [Moghaddam e Mahlooji 2017], [Parnianifard et al. 2020].

Muitos autores destacam a importância do arranjo experimental utilizado na geração da base de treino dos metamodelos [Kim et al. 2016], [Nasiri et al. 2017], [Parnianifard et al. 2020], [Smew et al. 2013], [Wade 2019], sendo o *Latin Hypercube Design* (LHD) [S. H. Kim e Boukouvala 2020], [Parnianifard et al. 2018], [Parnianifard e Azfanizam 2020], [Smew et al. 2013] e o Design Aleatório [Dengiz et al. 2009], [Kabirian e Ólafsson 2011], [Shourian et al. 2008], [Sreekanth e Datta 2011], [Yadav et al. 2016] os mais comumente utilizados em estudos desta área.

Além disto, a evolução no campo de *machine learning* (ML) proporcionou o desenvolvimento de algoritmos com alta capacidade de aprendizado, impulsionando estudos envolvendo seu uso para criação de metamodelos mais precisos e flexíveis (Batur et al. 2018). Xia et al. [2017] ressaltam, que a correta seleção dos algoritmos de ML e seus hiper-parâmetros (HP) são alguns dos fatores mais críticos na criação de metamodelos eficazes. Embora seja um dos fatores que mais impactam no sucesso de um projeto de otimização via simulação por metamodelagem (OvSM), foram encontrados na literatura poucos trabalhos que enfatizam o uso de técnicas estruturadas para seleção e/ou otimização dos HPs (OHP) em estudos de OvSM [Amiri et al. 2019], [Hou et al. 2019], [Hüllen et al. 2020], [Kim e Boukouvala 2020], [Li et al. 2019].

Neste sentido, este estudo propõe um *framework* para condução de projetos de OvSM que incorpora algoritmos de ML, OHP e *Design of Experiments* (DOE). Além disto, objetiva-se analisar (i) se existe ganho de eficiência promovido pelo uso do arranjo LHD frente ao aleatório, (ii) qual o ganho de performance e consequente custo computacional promovido pelo uso da OHP por Algoritmo Genético (AG), e (iii) se o uso da metamodelagem permite ganhos de tempo computacional e/ou qualidade da resposta frente a otimização via software comercial baseado em metaheurísticas. A fim de responder as perguntas de pesquisa, o *framework* proposto será aplicado em três problemas reais de engenharia industrial.

Este artigo está dividido em mais quatro Seções. A Seção 2 apresenta a revisão da literatura acerca do tema, a Seção 3 expõe os materiais e métodos empregados, a Seção 4 apresenta os resultados encontrados e a Seção 5 conclui o trabalho.

2. Revisão de Literatura

2.1. Metamodelagem

De acordo com De La Fuente e Smith [2017], quanto mais complexo é o sistema estudado, mais tempo é demandado na execução da otimização. Segundo os autores, para se obter bons resultados em tempo hábil, faz-se necessário elevar o problema a um outro nível de abstração, chamado de metamodelo.

A metamodelagem é uma abordagem comumente usada para desenvolver um modelo representativo de um modelo de simulação [De La Fuente et al. 2019]. Metamodelos são aproximações matemáticas utilizadas para substituir modelos de simulação, cuja complexidade os torna caros de se executar [Barton 2009]. A metamodelagem pode ser definida como o processo de construção de modelos matemáticos capazes de capturar o relacionamento *input*/*output* da simulação, sendo capaz de gerar previsões para o valor da FO de determinada solução, em um tempo muito menor que o modelo de simulação original [Kleijnen 2014], [Parnianifard et al. 2020]

A fim de criar metamodelos mais fidedignos, nos últimos anos surgiram diversos métodos de metamodelagem na literatura. Barton [2009] comparou os principais métodos utilizados para este tipo de aplicação, a saber: *Response Surface Metodology*, *Radial Basis Function*, *Kriging*, Redes Neurais Artificiais (RNA), e *Spline*. Entretanto, com o advento da inteligência artificial, muitos pesquisadores buscam incor

porar técnicas de ML na simulação a fim de melhorar a velocidade de execução, precisão e confiabilidade, sendo que muitas destas aplicações estão relacionadas com: metamodelo e análise, predição de parâmetros, e tomada de decisão inteligente [De La Fuente et al. 2019].

Considerando seu treinamento por meio de dados amostrais rotulados, os metamodelos podem ser classificados como uma técnica de aprendizado supervisionado, que busca estabelecer um relacionamento de causa e efeito entre as variáveis de entrada e saída do modelo [Amaran et al. 2016], [Dunke e Nickel 2020]. Hassayebi et al. [2019] destacam que as aplicações de metamodelagem em problemas da indústria e serviços tem crescido devido a sua capacidade de reduzir o número de simulações necessárias no processo de otimização.

Neste trabalho serão utilizados os algoritmos de ML mais usados na literatura para problemas de OvS para modelos de SED, sendo eles: *Gaussian Process* (GP), Regressão Polinomial (RP), RNA, *Ensembles* (*Randon Forest* – RF, e *Gradient Boosted Regression Tree* - *GBT*) e *Support Vector Machine* (SVM), assim como sugerido por Amaral [2021].

2.2. Otimização de Hiper-parâmetros

A escolha adequada dos hiper-parâmetros dos algoritmos de ML é uma etapa fundamental quando se deseja aumentar a capacidade de aprendizagem destes algoritmos. Perante a inviabilidade da definição manual destes parâmetros, torna-se imprescindível o uso de técnicas de otimização para encontrar a configuração que extraia a melhor performance do algoritmo [XIA et al. 2017].

Atualmente existem diversos métodos de otimização para seleção de hiper-parâmetros, tais como AG, colônia de formigas, *grid search*, *bayesian optimization*, entre outros (Hüllen et al. 2020], [Kim e Boukouvala 2020], [Li, Tan e Ng 2019]. Para este trabalho optou-se pelo uso de AG devido a sua velocidade de execução, capacidade de adaptação perante diferentes tipos de variáveis e não requerer a definição de valores iniciais para performar a otimização.

O AG é uma técnica heurística inspirada na teoria biológica da seleção natural. Proposto inicialmente por De Jong [1975] e Holland [1975], é baseada no princípio da sobrevivência do mais forte de Darwin, sendo utilizado com sucesso em problemas de programação linear e não linear [Bashiri e Geranmayeh 2011], [Lessmann et al. 2006]. No AG, o espaço de busca é composto por soluções candidatas, sendo cada uma representada por um cromossomo [Alabas et al. 2002]. O cromossomo é composto por genes, representando uma codificação dos parâmetros de uma solução candidata e um valor de aptidão, ou FO, para alcançar a melhor solução do

problema. O objetivo do AG é promover a combinação dos genes de modo a obter novos cromossomos com melhores valores de FO [Li et al. 2010].
O uso do AG para OHP pode ser observado em diversos trabalhos (*e.g.*, [Hou et al. 2019] e [Hüllen et al. 2020]). Este método avalia diversas combinações possíveis para os parâmetros {a_1, ..., a_i} do algoritmo de ML, e para cada combinação é utilizada a técnica de *cross-validation* para calcular o erro associado a esta.
Considerada uma das mais importantes técnicas de avaliação de métodos de regressão e classificação, a *k-fold cross-validation* divide igualmente e aleatoriamente os dados de treino em k partes, utilizando-se k-1 partes para treinamento do algoritmo e o restante para teste. Cada arranjo tem seu erro calculado k vezes, sendo que em cada interação é selecionado uma parte k_i diferente para teste, com i = {1, ..., k}. Então, a performance é medida com base na média dos erros, nas k interações de validação, performando uma métrica mais robusta do que se medida em apenas uma observação [Bergmeir e Benítez 2012]. Roy e Datta [2019] sugerem o uso de *10-fold cross-validation*.

3. Materiais e Métodos

A Figura 1 apresenta o *framework* de OvSM proposto neste trabalho, que incorpora tanto atividades estabelecidas no ambiente de simulação, à direita, quanto as desenvolvidas no ambiente de metamodelagem, à esquerda. Esta abordagem assume que o modelo de simulação já foi construído, verificado e validado (Passo 0), etapa fundamental para sua otimização.

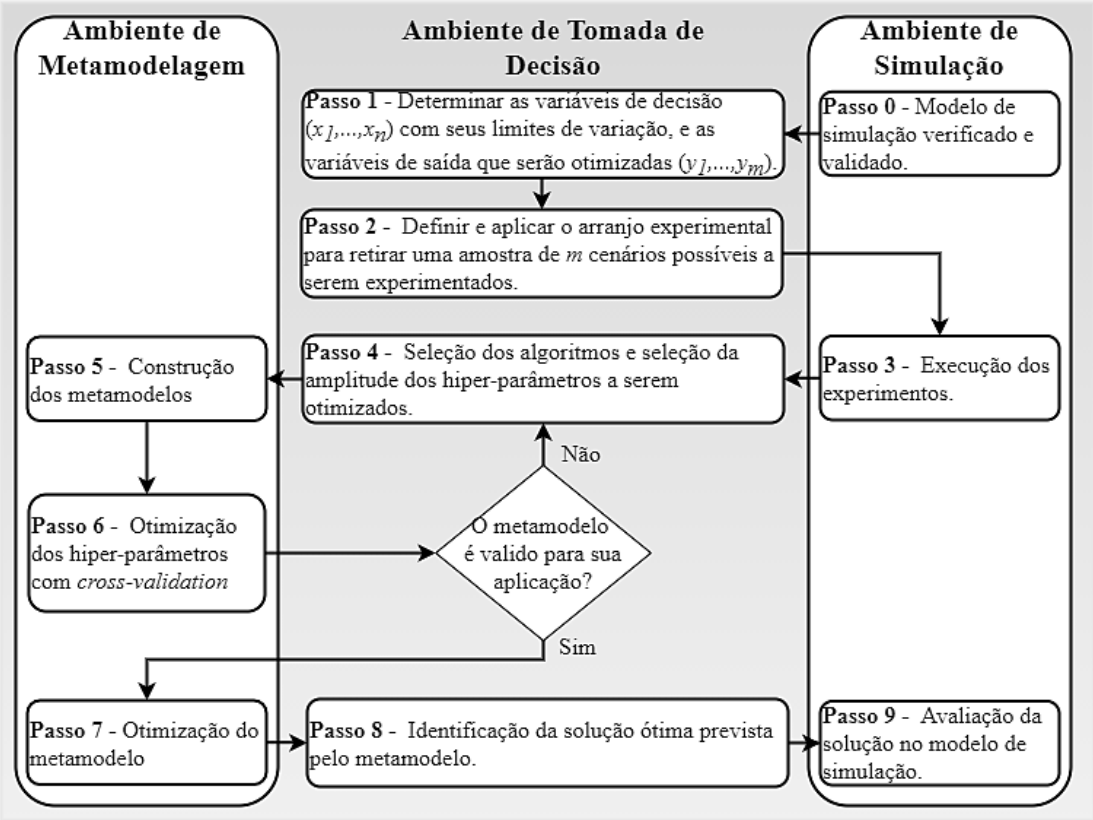


Figura 1 - *framework* proposto para otimização via simulação por metamodelagem

Os passos do *framework* são descritos como se segue:

1. Determine as variáveis de decisão (x_i, ..., x_n) e suas amplitudes de variação, bem como a função objetivo (FO) a ser otimizada (y);

2. Defina o arranjo experimental e selecione uma amostra de *m* cenários que irão compor a base de treinamento dos metamodelos. Neste trabalho foi testado o arranjo LHD nos tamanhos amostrais 20*x*, 40*x*, 60*x*, 80*x* e 100*x* e o arranjo Aleatório no tamanho 100*x* (no qual *x* é o número de variáveis de decisão do problema);
3. Execute os experimentos no modelo de simulação e armazene os dados simulados;
4. Selecione os algoritmos de ML e os seus hiper-parâmetros a serem otimizados, assim como sua amplitude de variação;
5. Construa a estrutura do metamodelo no software ou linguagem de programação escolhida;
6. Execute a OHP via AG com *k-fold cross-validation*. O AG foi estabelecido com um tamanho de população de 5 indivíduos, torneio como estratégia de seleção, mutação Gaussiana, 0,9 de probabilidade de cruzamento e o número de geração de 50;
7. Execute a otimização do metamodelo para encontrar os valores das variáveis de decisão que otimizam o valor predito da FO definido na etapa 1. Nesta etapa, o tomador de decisão pode utilizar as técnicas de otimização que julgar convenientes. Considerando o aspecto discreto das variáveis dos problemas deste artigo, foi usada a otimização por *Ranking and Selection*;
8. Identifique a solução que otimiza a FO prevista pelo metamodelo;
9. Avalie a solução encontrada no modelo de simulação.

Para o desenvolvimento dos modelos de simulação, foi utilizado o software *ProModel* (versão 18.2), enquanto para a criação dos metamodelos optou-se pelo software *RapidMiner* versão (9.6.000). Como referência, usou-se o otimizador comercial *SimRunner*. O *SimRunner* é um software de otimização integrado ao *ProModel* e baseado em uma metaheurística, que combina AG e estratégias evolutivas, sendo uma variante dos algoritmos evolutivos.

4. Resultados

4.1. Apresentação dos Objetos de Estudo

O primeiro objeto de estudo (OE1) representa uma célula produtiva do setor de telecomunicações, no qual o objetivo da otimização foi encontrar o valor que maximiza o lucro (y_1) da empresa. O OE1 possui 7 variáveis de decisão inteiras, a saber: número de operadores (x_1), número de bancadas tipo 1 (x_2), número de bancadas tipo 2 (x_3), variável binária para atividade de organizador material (x_4), e tamanho dos estoques intermediários (x_5, x_6, x_7). Considerando os limites inferiores para as variáveis, *L* = [1 1 1 0 5 5 5], e limites superiores, *U* = [3 4 4 1 15 15 15], o problema tem um total de 127.000 possíveis soluções. O valor *benchmark* obtido pelo *SimRunner* foi de 54,93 mil dólares, demandando 245 experimentos (8.087 segundos) para performar a otimização.

O segundo objeto de estudo (OE2) representa uma célula de controle de qualidade de uma empresa produtora de transmissores, cujo principal objetivo é a maximização da função lucro. Este caso possui 6 variáveis de decisão, correspondendo ao número de operadores responsáveis por cada teste de qualidade (x_1, x_2, x_3) e ao número de equipamentos para realização dos testes dos tipos 1 2 e 3 (x_4, x_5, x_6). Considerando os limites de variação dos recursos, *L* = [1 1 1 1 1 1] e *U* = [10 10 10 10 10 10], o OE2 possui um total de 1.000.000 de soluções possíveis. Para este problema, a solução ótima *benchmark* encontrada pelo *SimRunner* levou a um lucro de 330,12 mil dólares, demandando 451 experimentos (68.552 segundos).

Por fim, o terceiro objeto de estudo (OE3) visa a maximização da quantidade de peças produzidas em uma célula de produção de autopeças. Neste sistema foram estudadas 11 variáveis de decisão, que correspondem ao número de operadores do tipo 1 2, e 3 (x_1, x_2, x_3) e a quantidade de máquinas de cada um dos 8 tipos existentes $(x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11})$. Os valores dessas variáveis podem oscilar entre os limites inferiores, *L* = [1 1 1 1 1 1 1 1 1 1 1], e superiores, *U* = [5 4 6 3 3 3 3 4 4 4 4]. Devido a amplitude das variáveis, o problema possui um total de 2.488.320

soluções possíveis. O melhor valor encontrado pelo *SimRunner* neste caso foi de 880,97 mil peças produzidas, demandando 787 experimentos e 75.552 segundos.
O *framework* proposto na Seção 3 foi aplicada a cada um dos OEs apresentados. Para cada um, analisou-se 6 arranjos experimentais (LHD 20*x*, LHD 40*x*, LHD 60*x*, LHD 80*x*, LHD100*x*, e Aleatório 100*x*) e 6 algoritmos de ML (GP, RP, RNA, RF, GBT e SVM), treinados com a etapa de otimização de hiper-parâmetros e sem esta etapa (hiper-parâmetros *default*). Os dados coletados de cada uma das configurações de cada OE foram padronizados e analisados, sendo os resultados apresentados na Seção 4.2.

4.2. Análise dos Resultados

Esta seção apresentará e discutirá os principais resultados obtidos no trabalho. O primeiro ponto a ser destacado é que o uso da metamodelagem pode contribuir significativamente no processo de OvS. Ressalta-se que o principal objetivo deste trabalho não foi concorrer em tempo com o *software SimRunner*, uma vez que o metamodelo, quando treinado e validado, pode ser utilizado inúmeras vezes pelo tomador de decisões para performar otimizações em um curto espaço de tempo. Neste sentido o resultado obtido com o *software SimRunner* foi tomado como uma referência para determinar se o metamodelo obteve, ou não, bons resultados. A Tabela 1 resume os principais resultados do melhor cenário encontrado para cada um dos três casos analisados.

Tabela 1 - Resultado dos testes estatísticos para os objetos de estudo

OE	Método	Arranjo	Nº de experimentos	FO (mil)	Teste de Anderson-Darling	Teste F	Teste T	Teste Mann-Whitney
1	SVM	LHD20*x*	140	54,93	0,644	1,000	1,000	-
	Referência	-	245	54,93	0,644			
2	RF	LHD40*x*	240	317,45	0,391	0,568	0,001	-
	Referência	-	451	330,12	0,167			
3	RF	LHD60x	660	879,20	<0,005	-	-	0,000
	Referência 1	-	325	859,30	<0,005			
	Referência 2	-	787	880,97	<0,005	-	-	0,971

De acordo com a Tabela 1, o primeiro OE foi capaz de atingir o valor referência demandando 42,85% menos experimentos. Já o OE2 não foi capaz de atingir a FO referência, obtendo um lucro 3,83% menor, porém demandando 46,78% menos experimentos. Por outro lado, o OE3 superou a FO referência, produzindo 19.900 peças a mais, porém demandando mais que o dobro de experimentos. Para este último caso, vale ressaltar que, devido o metamodelo ter alcançado um valor acima do encontrado pelo *SimRunner*, este foi executado novamente, agora no modo cauteloso, e encontrou uma solução com FO estatisticamente igual ao encontrado pelo metamodelo, porém demandando 16,04% mais tempo.
Outro ponto a ser ressaltado é que no segundo e terceiro OE, o metamodelo que melhor performou foi o RF, obtendo também bons resultados para o OE1 nos arranjos LHD 40*x*, 60*x*, 80*x*, 100*x* e aleatório 100*x*. Estes resultados apontam o RF como o algoritmo mais adaptável perante as diferentes características dos objetos de estudo.
Os resultados de FO e RMSE (média dos três OEs padronizados entre 0 e 1) dos metamodelos obtidos perante cada arranjo estudado estão apresentados nas Tabelas 2 e 4 (para metamodelos com hiper-parâmetros *default*) e Tabelas 3 e 5 (para metamodelos com a otimização de hiper-parâmetros).

Tabela 2 - FO média: metamodelo vs. arranjo com hiper-parâmetros *default*

Arranjo	GP	RP	RNA	RF	GBT	SVM	Média Arranjo
LHD 20x	0,892	0,657	0,847	0,921	0,795	0,596	0,785
LHD 40x	0,933	0,775	0,762	0,944	0,860	0,538	0,802
LHD 60x	0,703	0,350	0,878	0,899	0,738	0,546	0,686
LHD 80x	0,763	0,527	0,847	0,894	0,752	0,601	0,731
LHD 100x	0,929	0,612	0,726	0,902	0,826	0,619	0,769
Aleatório 100x	0,898	0,587	0,865	0,949	0,828	0,566	0,782
Média Metamodelo	**0,853**	**0,585**	**0,821**	**0,918**	**0,800**	**0,578**	

Tabela 3 - FO média: metamodelo vs. arranjo com hiper-parâmetros otimizados

Arranjo	GP	RP	RNA	RF	GBT	SVM	Média Arranjo
LHD 20x	0,689	0,754	0,903	0,918	0,687	0,869	**0,803**
LHD 40x	0,691	0,656	0,835	0,956	0,899	0,931	**0,828**
LHD 60x	0,626	0,479	0,745	0,876	0,534	0,828	**0,681**
LHD 80x	0,619	0,590	0,837	0,855	0,738	0,781	**0,737**
LHD 100x	0,688	0,667	0,791	0,964	0,798	0,891	**0,800**
Aleatório 100x	0,774	0,635	0,971	0,959	0,909	0,806	**0,842**
Média Metamodelo	**0,681**	**0,630**	**0,847**	**0,921**	**0,761**	**0,851**	

Tabela 4 - RMSE média: metamodelo vs. arranjo com hiper-parâmetros *default*

Arranjo	GP	RP	RNA	RF	GBT	SVM	Média Arranjo
LHD 20x	0,352	0,817	0,116	0,092	0,153	0,184	0,286
LHD 40x	0,271	0,710	0,106	0,084	0,152	0,187	0,252
LHD 60x	0,172	0,621	0,139	0,128	0,163	0,185	0,235
LHD 80x	0,165	0,655	0,135	0,125	0,159	0,179	0,236
LHD 100x	0,218	0,594	0,076	0,063	0,140	0,182	0,212
Aleatório 100x	0,248	0,601	0,085	0,072	0,143	0,172	0,220
Média Metamodelo	**0,238**	**0,666**	**0,110**	**0,094**	**0,152**	**0,182**	

Tabela 5 - RMSE média: metamodelo vs. arranjo com hiper-parâmetros otimizados

Arranjo	GP	RP	RNA	RF	GBT	SVM	Média Arranjo
LHD 20x	0,116	0,465	0,081	0,088	0,069	0,092	**0,152**
LHD 40x	0,103	0,467	0,066	0,080	0,061	0,090	**0,144**
LHD 60x	0,141	0,515	0,132	0,125	0,124	0,130	**0,195**
LHD 80x	0,143	0,509	0,124	0,124	0,125	0,129	**0,192**
LHD 100x	0,093	0,482	0,043	0,063	0,052	0,380	**0,186**
Aleatório 100x	0,105	0,456	0,052	0,070	0,061	0,084	**0,138**
Média Metamodelo	**0,117**	**0,482**	**0,083**	**0,092**	**0,082**	**0,151**	

Ao analisar as Tabelas 2 e 3, cujos dados representam a média da FO dos metamodelos dos três casos estudados, percebe-se que dentre os metamodelos treinados com hiper-parâmetros *default* o RF, RNA e GP foram os que atingiram os melhores valores médios de FO, sendo o SVM o de

pior desempenho. Todavia quando se trata dos metamodelos com hiper-parâmetros otimizados, o SVM torna-se o segundo melhor, ficando claro o impacto desta etapa no desempenho deste algoritmo. Vale ressaltar que o RF possuiu o melhor desempenho nas duas situações analisadas e maior flexibilidade entre os arranjos experimentais utilizados.

Ao comparar o RMSE médio dos metamodelos apresentados nas Tabelas 4 e 5, é possível observar que os metamodelos baseados em RF, GBT e RNA foram os de melhores resultados, ou seja, aqueles que melhor representaram os modelos de simulação estudados. Destaca-se que os algoritmos RP apresentaram, em todos os arranjos, os piores resultados em termos de FO e RMSE. Em contrapartida, o RF, RNA, e GBT apresentaram, em média, bons resultados.

Um ponto a ser destacado é que, considerando o RMSE dos metamodelos com hiper-parâmetros otimizados, houve uma boa performance nos arranjos LHD 20*x*, LHD 40*x*, LHD 100*x* e aleatório 100*x*, sendo o LHD 60*x* e 80*x* os arranjos de piores desempenhos médios. Já quando se observa os metamodelos com hiper-parâmetros *default*, as melhores métricas foram encontradas nos experimentos dos arranjos LHD 100*x* e Aleatório 100*x*. As Figuras 2(a) e 2(b) apresentam o comparativo entre os arranjos para o FO e RMSE.

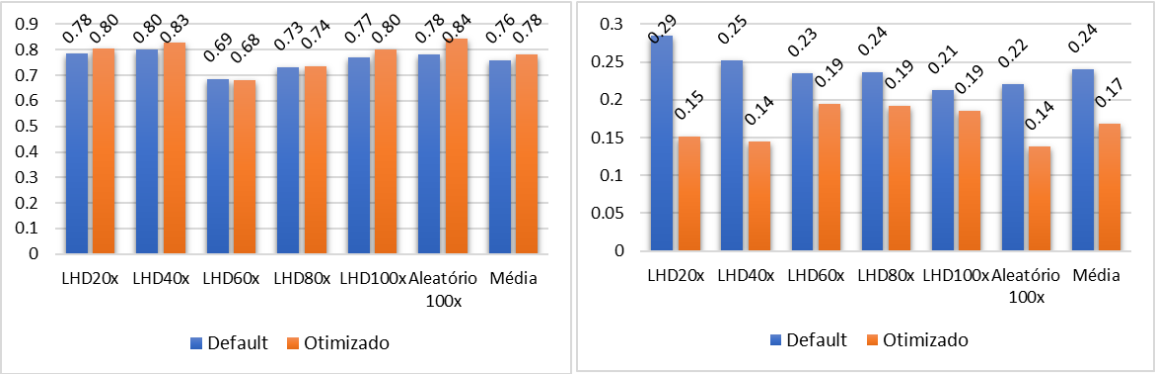


Figura 2 – Comparativo entre os arranjos para a (a) FO e o (b) RMSE.

Como apresentado na Figura 2, o arranjo Aleatório 100*x* foi o que esboçou o melhor resultado em termos de FO e RMSE, todavia equiparou-se estatisticamente ao LHD 40*x* (*p-value* = 0,887 para o teste de Mann-Whitney), que utiliza 60% menos experimentos. Não obstante, comparou-se o arranjo Aleatório 100*x* com o LHD 100*x* (arranjo de mesmo tamanho amostral), sendo possível inferir que não houve diferença estatisticamente significativa quando se tratou do valor da mediana da FO (*p-value* de 0,937 para hiper-parâmetros *default* e 0,788 para hiper-parâmetros otimizados para o teste de Mann-Whitney). Em respeito ao RMSE, embora possa ser observado uma diferença na Figura 2 (b), não foi encontrado uma diferença estatisticamente significativa do RMSE obtido com o arranjo LHD 100*x* e o Aleatório 100*x* (*p-value* de 0,812 para hiper-parâmetros *default* e 0,887 para hiper-parâmetros otimizados para o teste de Mann-Whitney).

Portanto, para os casos analisados, não se encontrou evidências que atestem a superioridade ou inferioridade do LHD frente ao arranjo aleatório, todavia observou-se que o LHD 40*x* atingiu resultados similares ao aleatório, mesmo executando 60% menos experimentos, podendo ser um indicativo que LHD consegue amostrar satisfatoriamente o espaço de busca, mesmo com poucos experimentos.

Outro ponto a ser destacado é que cada problema apresenta peculiaridades que podem demandar diferentes tamanhos amostrais na construção de metamodelos, não sendo possível determinar um tamanho amostral que sirva de referência para projetos desta natureza. Todavia, embora com base em apenas três casos, foi possível estabelecer uma correlação entre o tamanho do espaço de soluções dos problemas e o tamanho *N* do arranjo LHD. O modelo de regressão linear para esta relação obteve um R^2 de 0,982 e é apresentado na Eq. (1).

$$N = (19{,}675 + 1{,}728 \times 10^{-5} D)x \quad (1)$$

Sendo *N* o tamanho da amostra, *D* a tamanho do espaço de soluções e x o número de variáveis de decisão do problema.
Quanto aos resultados da otimização de hiper-parâmetros, a Figura 3 (a) apresenta a redução média no RMSE proporcionada pelo uso desta técnica em cada metamodelo e a Figura 3 (b) o comparativo de RMSE e FO para metamodelos com hiper-parâmetro *default* e otimizados.

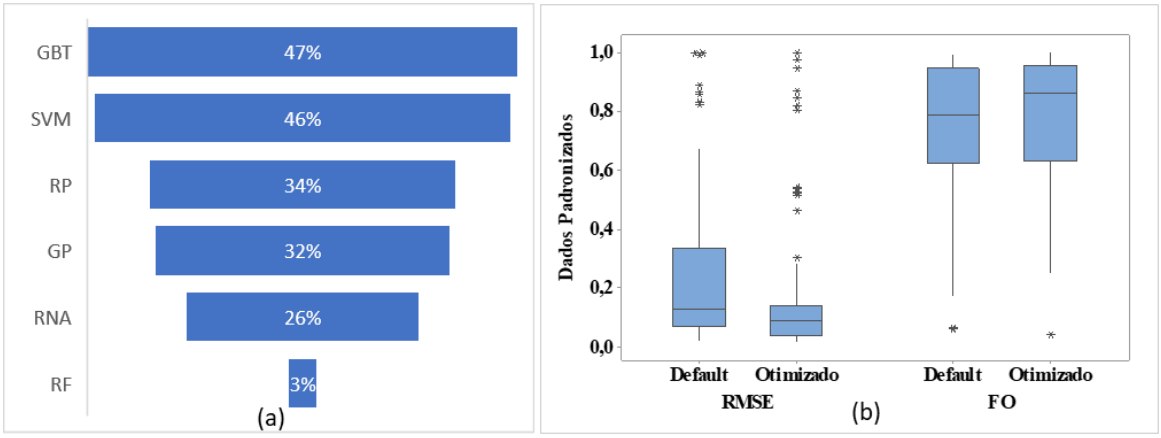


Figura 3 – (a) Redução no RMSE com o uso da otimização de hiper-parâmetros], [(b) Comparativo entre os metamodelos com hiper-parâmetros *default* e otimizados

Para os problemas estudados, esta etapa permitiu uma redução média de 31,28% no RMSE dos metamodelos, reforçando a importância da escolha adequada dos hiper-parâmetros. Dentre os metamodelos, o GBT e SVM foram os que esboçaram maior redução do RMSE, enquanto o RF apresentou uma redução de apenas 3% com o uso da otimização. Embora em termos de FO não tenha sido encontradas melhorias significativas com o uso da otimização de hiper-parâmetros (*p-value* = 0,265 para o teste de Mann-Whitney), a mediana do RMSE dos metamodelos otimizados foi estatisticamente menor que o RMSE dos metamodelos com hiper-parâmetros *default* (*p-value* = 0,000 para o teste de Mann-Whitney), ressaltando que esta técnica permite a criação de metamodelos mais precisos, e consequentemente, mais confiáveis. Vale ressaltar que o teste não paramétrico de Mann-Whitney foi escolhido devido as amostras não serem consideradas normais pelo teste de Anderson-Darling (ambas com *p-value* <0,005) e as amostras foram compostas pelos valores padronizados (entre 0 e 1) de todos metamodelos, arranjos e OEs.

5. Conclusões

A aplicação do *framework* proposto visou responder três questões de pesquisa principais, a saber: (i) Qual é o ganho de eficiência promovido pelo uso do arranjo LHD frente ao aleatório? (ii) Qual o ganho de performance e consequente custo computacional promovido pelo uso da otimização de hiper-parâmetros por AG? (iii) O uso da metamodelagem permite ganhos de tempo computacional e/ou qualidade da resposta frente a otimização pelo software comercial *SimRunner*?
Em resposta a primeira pergunta e ao contrário do que sugeria em parte a literatura, para os casos analisados não se encontrou evidências que comprovassem a superioridade ou inferioridade do LHD sobre o arranjo aleatório, uma vez que, em média, o aleatório se igualou estatisticamente em RMSE e FO ao arranjo LHD de mesmo número de experimentos (100*x*). Todavia, ao analisar separadamente os OEs, observa-se que o arranjo aleatório obteve resultados piores ou equiparáveis a arranjos LHD de menor tamanho, como no OE1, no qual se equiparou ao LHD

20*x*, arranjo cinco vezes menor, e na média dos três casos se equiparou ao LHD 40*x*, arranjo 60% menor.

Quanto a segunda pergunta de pesquisa, pode-se afirmar que o uso da otimização de hiper-parâmetros permitiu a criação de metamodelos mais precisos, uma vez que, em média, obteve-se uma redução de 31,28% no RMSE. Além disto, o tempo computacional demandado nesta etapa foi, em média, de 624,37 segundos, sendo considerado um tempo baixo se comparado ao tempo total de execução dos experimentos na simulação.

A respeito da terceira pergunta, dentre os três problemas estudados, o primeiro atingiu o mesmo valor estatístico da solução *benchmark*, o segundo obteve um resultado de FO apenas 3,83% inferior, e o terceiro superou o *benchmark* em 2,32%. Quanto ao tempo computacional demandado, o OE1 atingiu o *benchmark* com 42,85% menos tempo e o OE2, embora não tenha conseguido atingir *benchmark*, demandou um tempo 46,78% menor. Já no OE3, o melhor valor de FO foi atingido pelo RF no arranjo LHD 60*x*, que superou o *benchmark* em 19.900 peças, porém demandando 103,4% mais tempo computacional. Vale ressaltar que, para efeito comparativo, para o *SimRunner* atingir a mesma FO encontrada pelo metamodelo foram necessários 787 experimentos, levando um tempo 16,04% maior em relação ao RF no arranjo LHD 60*x*.

Uma outra oportunidade a ser explorada em futuras pesquisas é o estudo aprofundado do tamanho do espaço amostral a ser utilizado em problemas de metamodelagem. Este estudo pode ser conduzido pela aplicação do *framework* proposto em uma gama maior de OEs, de modo a buscar identificar a correlação entre o tamanho amostral e as características do problema, tais como dimensão, espaço de soluções, variabilidade, entre outras.

Sugere-se também realizar uma comparação sistemática entre os arranjos LHD e aleatório, avaliando suas performances em diferentes tamanhos amostrais. Além disto, seria interessante a expansão desta comparação para outras estratégias de amostragem, tais como *factorial design*, CCD, *orthogonal arrays*, entre outros.

Agradecimentos

Os autores agradecem o apoio da FAPEMIG, CAPES e CNPq.

Referências

Alabas, C., Altiparmak, F., e Dengiz, B. (2002). A comparison of the performance of artificial intelligence techniques for optimizing the number of kanbans. *Journal of the Operational Research Society*, *53*(8), 907–914. https://doi.org/10.1057/palgrave.jors.2601395

Amaral, J. V. S. do. (2021). *Otimização Baseada em Metamodelos: Uma Abordagem para Metamodelagem em Simulação a Eventos Discretos*. Universidade Federal de Itajubá.

Amaran, S., Sahinidis, N. V., Sharda, B., e Bury, S. J. (2016). Simulation optimization: a review of algorithms and applications. *Annals of Operations Research* *240*(1), 351–380. https://doi.org/10.1007/s10479-015-2019-x

Amiri, F., Shirazi, B., e Tajdin, A. (2019). Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop. *Applied Soft Computing Journal*, *75*, 190–202. https://doi.org/10.1016/j.asoc.2018.11.015

Barton, R. R. (2009). Simulation Optimization Using Metamodel. *Winter Simulation Conference* 2 230–238.

Barton, R. R., e Meckesheimer, M. (2006). Chapter 18 Metamodel-Based Simulation Optimization. *Handbooks in Operations Research and Management Science*, *13*(C), 535–574. https://doi.org/10.1016/S0927-0507(06)13018-2

Bashiri, M., e Geranmayeh, A. F. (2011). Tuning the parameters of an artificial neural network using central composite design and genetic algorithm. *Scientia Iranica*, *18*(6), 1600–1608. https://doi.org/10.1016/j.scient.2011.08.031

Batur, D., Bekki, J. M., e Chen, X. (2018). Quantile regression metamodeling: Toward improved

responsiveness in the high-tech electronics manufacturing industry. *European Journal of Operational Research* 2*64*(1) 212–224. https://doi.org/10.1016/j.ejor.2017.06.020
Bergmeir, C., e Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213. https://doi.org/10.1016/j.ins.2011.12.028
Cai, X., Qiu, H., Gao, L., Jiang, C., e Shao, X. (2019). An efficient surrogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems. *Knowledge-Based Systems*, *184*, 104901. https://doi.org/10.1016/j.knosys.2019.104901
De Jong, K. A. (1975). *Analysis of the Behavior of a Class of Genetic Adaptive Systems*.
De La Fuente, R., Erazo, I., e Smith, R. L. (2019). Enabling intelligent processes in simulation utilizing the tensorflow deep learning resources. *Proceedings - Winter Simulation Conference* 2*018-Decem*, 1108–1119. https://doi.org/10.1109/WSC.2018.8632539
De La Fuente, R., e Smith, R. (2017). Metamodeling a system dynamics model: a contemporany comparison of methods. *Winter Simulation ConferenceWinter Simulation Conference*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004
Dengiz, B., Alabas-Uslu, C., e Dengiz, O. (2009). Optimization of manufacturing systems using a neural network metamodel with a new training approach. *Journal of the Operational Research Society*, *60*(9), 1191–1197. https://doi.org/10.1057/palgrave.jors.2602620
Dunke, F., e Nickel, S. (2020). Neural networks for the metamodeling of simulation models with online decision making. *Simulation Modelling Practice and Theory*, *99*, 102016. https://doi.org/10.1016/j.simpat.2019.102016
Fu, M. C. (2002). Feature Article: Optimization for simulation: Theory vs. Practice. *INFORMS Journal on Computing*, *14*(3), 192–215. https://doi.org/10.1287/ijoc.14.3.192.113
Hassannayebi, E., Boroun, M., Alaei Jordehi, S., e Kor, H. (2019). Train schedule optimization in a high-speed railway system using a hybrid simulation and meta-model approach. *Computers and Industrial Engineering*, *138*(October), 106110. https://doi.org/10.1016/j.cie.2019.106110
Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (1º). MIT Press.
Hou, Z., Dai, Z., Lao, W., Wang, Y., e Lu, W. (2019). Application of Mixed-Integer Nonlinear Optimization Programming Based on Ensemble Surrogate Model for Dense Nonaqueous Phase Liquid Source Identification in Groundwater. *Environmental Engineering Science*, *36*(6), 699–709. https://doi.org/10.1089/ees.2018.0366
Hüllen, G., Zhai, J., Kim, S. H., Sinha, A., Realff, M. J., e Boukouvala, F. (2020). Managing uncertainty in data-driven simulation-based optimization. *Computers and Chemical Engineering*, *136*, 106519. https://doi.org/10.1016/j.compchemeng.2019.106519
Jalali, H., e Nieuwenhuyse, I. Van. (2015). Simulation optimization in inventory replenishment: A classification. *IIE Transactions*, *47*(11), 1217–1235. https://doi.org/10.1080/0740817X.2015.1019162
Kabirian, A., e Ólafsson, S. (2011). Continuous optimization via simulation using Golden Region search. *European Journal of Operational Research* 2*08*(1), 19–27. https://doi.org/10.1016/j.ejor.2010.09.002
Kim, S. H., e Boukouvala, F. (2020). Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. *Optimization Letters*, *14*(4), 989–1010. https://doi.org/10.1007/s11590-019-01428-7
Kim, W., Jeon, Y., e Kim, Y. (2016). Simulation-based optimization of an integrated daylighting and HVAC system using the design of experiments method. *Applied Energy*, *162*, 666–674. https://doi.org/10.1016/j.apenergy.2015.10.153
Kleijnen, J. P. C. (2014). Simulation-optimization via Kriging and bootstrapping: a survey. *Journal of Simulation*, *8*(4) 241–250. https://doi.org/10.1057/jos.2014.4
Lessmann, S., Stahlbock, R., e Crone, S. F. (2006). Genetic algorithms for support vector machine model selection. *IEEE International Conference on Neural Networks - Conference Proceedings*, 3063–3069. https://doi.org/10.1109/ijcnn.2006.247266

Li, G., Hwai-yong Tan, M., e Hui Ng, S. (2019). Metamodel-based optimization of stochastic computer models for engineering design under uncertain objective function. *IISE Transactions*, *51*(5), 517–530. https://doi.org/10.1080/24725854.2018.1504355

Li, Y. F., Ng, S. H., Xie, M., e Goh, T. N. (2010). A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems. *Applied Soft Computing Journal*, *10*(4), 1257–1273. https://doi.org/10.1016/j.asoc.2009.11.034

Moghaddam, S., e Mahlooji, H. (2017). A new metamodel-based method for solving semi-expensive simulation optimization problems. *Communications in Statistics: Simulation and Computation*, *46*(6), 4795–4811. https://doi.org/10.1080/03610918.2015.1134567

Nasiri, M. M., Yazdanparast, R., e Jolai, F. (2017). A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule. *International Journal of Computer Integrated Manufacturing*, *30*(12), 1239–1252. https://doi.org/10.1080/0951192X.2017.1307452

Parnianifard, A., e Azfanizam, A. S. (2020). 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*, *33*(1), 1–20. https://doi.org/10.1002/jnm.2679

Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., e Ismail, M. I. S. (2020). Comparative study of metamodeling and sampling design for expensive and semi-expensive simulation models under uncertainty. *Simulation*, *96*(1), 89–110. https://doi.org/10.1177/0037549719846988

Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., Ismail, M. I. S., Maghami, M. R., e Gomes, C. (2018). 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*, *15*(5), 1471–1479. https://doi.org/10.1166/jctn.2018.7379

Roy, D. K., e Datta, B. (2019). Adaptive Management of Coastal Aquifers Using Entropy-Set Pair Analysis–Based Three-Dimensional Sequential Monitoring Network Design. *Journal of Hydrologic Engineering* 24(3), 04018072. https://doi.org/10.1061/(asce)he.1943-5584.0001765

Shourian, M., Mousavi, S. J., Menhaj, M. B., e Jabbari, E. (2008). Neural-network-based simulation-optimization model for water allocation planning at basin scale. *Journal of Hydroinformatics*, *10*(4), 331–343. https://doi.org/10.2166/hydro.2008.057

Smew, W., Young, P., e Geraghty, J. (2013). Supply chain analysis using simulation, gaussian process modelling and optimisation. *International Journal of Simulation Modelling*, *12*(3), 178–189. https://doi.org/10.2507/IJSIMM12(3)4.239

Sreekanth, J., e Datta, B. (2011). Optimal combined operation of production and barrier wells for the control of saltwater intrusion in coastal groundwater well fields. *Desalination and Water Treatment*, *32*(1–3), 72–78. https://doi.org/10.5004/dwt.2011.2680

Wade, B. M. (2019). Creating surrogate models for an air and missile defense simulation using design of experiments and neural networks. *Journal of Defense Modeling and Simulation*. https://doi.org/10.1177/1548512919877987

Xia, Y., Liu, C., Li, Y. Y., e Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, *78* 225–241. https://doi.org/10.1016/j.eswa.2017.02.017

Yadav, B., Ch, S., Mathur, S., e Adamowski, J. (2016). Estimation of in-situ bioremediation system cost using a hybrid Extreme Learning Machine (ELM)-particle swarm optimization approach. *Journal of Hydrology*, *543*, 373–385. https://doi.org/10.1016/j.jhydrol.2016.10.013