GRADIENT-BOOSTED TREES COMO ESTRATÉGIA PARA METAMODELAGEM EM PROBLEMAS DE OTIMIZAÇÃO VIA SIMULAÇÃO

João Victor Soares do Amaral (UNIFEI)
joao.victoramaral96@hotmail.com

José Arnaldo Barra Montevechi (UNIFEI)
montevechi@unifei.edu.br

Rafael de Carvalho Miranda (UNIFEI)
rafael.miranda@unifei.edu.br

No contexto da indústria 4.0, a otimização via simulação (OvS) surge como uma das mais potentes ferramentas da indústria moderna, permitindo aos decisores alocarem seus recursos de forma mais assertiva. Todavia, em sistemas muito complexos, o uso de técnicas convencionais de OvS demandam um tempo computacional que, muitas vezes, inviabiliza sua aplicação. Nos últimos anos, o desenvolvimento na área de machine learning surgiram algoritmos com alta capacidade de aprendizado, tornando o uso das técnicas de metamodelagem para solucionar problemas complexos de OvS um campo de estudo promissor. O presente estudo busca integrar técnicas de machine learning e OvS, utilizando o Gradient-Boosted Trees para metamodelagem em um problema real de OvS. A partir da construção, otimização e treinamento e aplicação do metamodelo obteve-se uma redução de 99,18% no tempo gasto na geração de todo espaço de solução, encontrando o mesmo valor que o bechmark obtido pelo software de otimização SimRunner.

Palavras-chave: Machine learning, Otimização via Simulação, Simulação a Eventos Discretos, Metamodelagem.

1. Introdução

Assegurar a competitividade das organizações, satisfazer seus clientes e reduzir custos requer que os sistemas produtivos lancem mão de ferramentas cada vez mais efetivas para auxílio à tomada de decisões (SALAM; KHAN, 2016). De acordo com Law (2013), soluções analíticas podem ser eficazes quando o sistema estudado é relativamente simples, todavia o autor ressalta que a grande maioria dos sistemas reais apresentam grande complexidade para serem analisados analiticamente. Neste sentido, a simulação é uma técnica desenvolvida para prever e avaliar a performance de sistemas complexos e estocásticos, os quais são, muitas vezes, analiticamente intratáveis (XU et al., 2016).

De acordo com Siebers et al. (2010), a técnica de simulação mais utilizada é a simulação a eventos discretos (SED), que busca representar computacionalmente sistemas reais, estudando o comportamento de variáveis que mudam de status em horizontes discretos de tempo (MONTEVECHI et al., 2015; SOUSA JUNIOR et al., 2019; LAW, 2013). Uma das vantagens do uso da SED é a redução do risco na tomada de decisão, uma vez que ela permite avaliar diversos cenários sem a interferência física no sistema produtivo (HELLENO et al., 2015). No entanto, quando existe uma grande quantidade de cenários possíveis, torna-se praticamente impraticável testar todos estes cenários, neste sentido, a otimização via simulação (OvS) é a técnica mais indicada para se atingir estes objetivos (BARTON, 2009)

O desenvolvimento na área de *machine learning* (ML) tem contribuído para o desenvolvimento de técnicas de otimização capazes de solucionar problemas de maior complexidade, como por exemplo, a criação de metamodelos. A metamodelagem refere-se criação de modelos representativos para modelos cuja complexidade os torna computacionalmente caros de se executar (DE LA FUENTE; ERAZO; SMITH, 2019). Um dos métodos mais conhecidos de ML é o *Gradient-Boosted Trees*, sendo inicialmente proposto por Schapire (1990) este método busca convergir uma série de árvores de decisão a fim de formar um algoritmo com alta capacidade de aprendizado.

Neste sentido, o presente trabalho visa avaliar o uso do GBT para metamodelagem em problemas de OvS. Para tanto este trabalho se propõe a construção de um modelo de SED de uma empresa real; a construção, validação e otimização dos hiper-parâmetros do metamodelo;

e otimização do metamodelo a fim de encontrar a solução de maior lucratividade para o objeto de estudo.

A fim de atingir os objetivos propostos, este artigo está organizado em mais quatro seções. A Seção 2 apresenta a revisão da literatura que baseou este trabalho, a Seção 3 descreve o método utilizado, a Seção 4 aplica o método proposto e discute seus resultados, e por fim, a Seção 5 resume as conclusões do estudo e direções para trabalhos futuros.

2. Referencial teórico

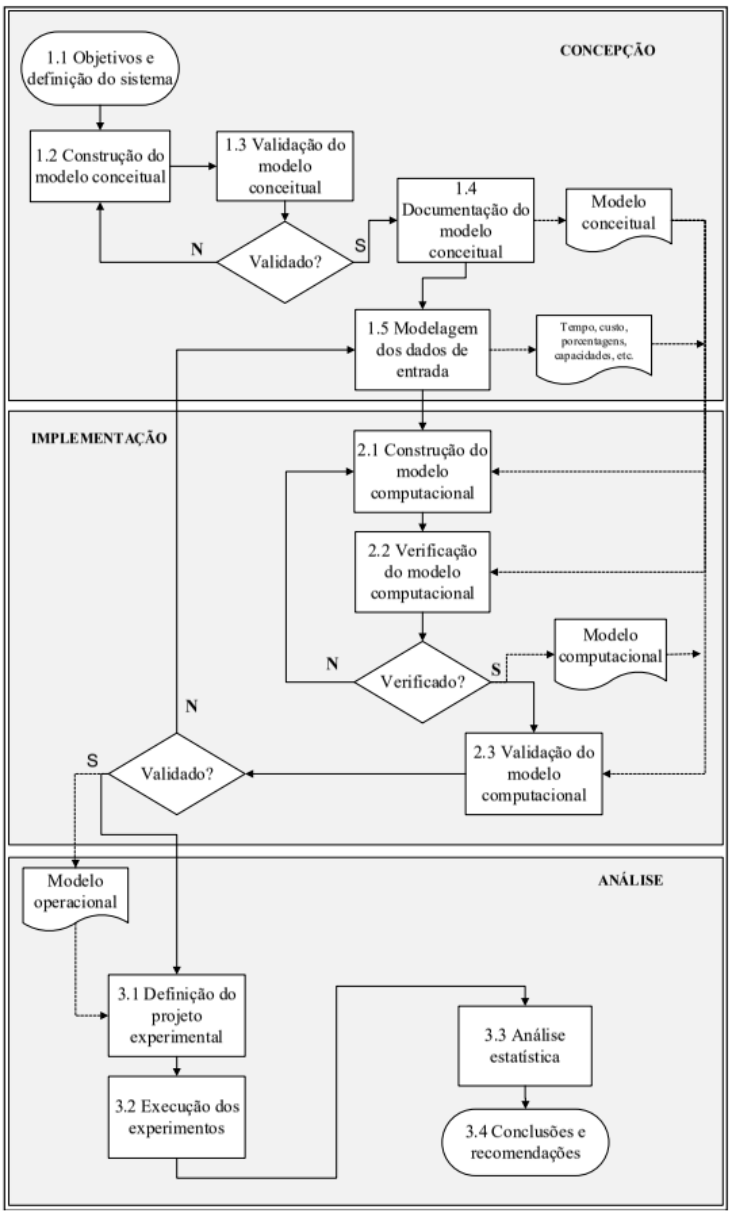
2.1. Simulação a eventos discretos

De acordo com Law (2013), a SED pode ser definida como a modelagem de um sistema que evolui ao longo de uma linha temporal, no qual eventos mudam de estado em pontos discretos no tempo. Desde sua criação nos anos de 1950, a simulação vem sendo empregada com sucesso na melhoria de processos, fornecendo uma ferramenta de modelagem e análise de sistemas complexos que são muitas vezes analiticamente intratáveis (MIRANDA et al., 2017).

Montevechi et al. (2007) descrevem as etapas de um projeto de SED, como visto na Figura 1. A primeira etapa constitui na concepção do modelo conceitual, definindo e descrevendo o objeto de estudo. A segunda etapa refere-se a etapa de implementação, verificação e validação do modelo computacional, já, na última etapa, chamada de Análise, é realizada a experimentação e análise dos resultados do modelo.

Law (2013) ressalta que a SED é uma alternativa de baixo custo para a experimentação real, evitando paradas no fluxo produtivo, bem como custos de implementação no chão de fábrica. Além disto, a SED permite ao analista experimentar diversos cenários que o levaria à melhor configuração do sistema estudado. Porém, de acordo com Xu et al. (2016), no estudo de sistemas complexos e com inúmeras possibilidades de configurações, faz-se necessário o uso de técnicas de otimização em conjunto com a SED, com o objetivo de encontrar o melhor design para o sistema.

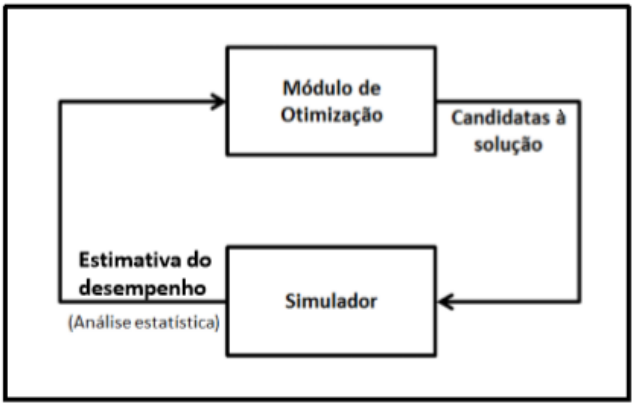
Figura 1: Etapas de um projeto de simulação a eventos discretos



Fonte: Adaptado de Montevechi et al. (2007)

Otimização via simulação refere-se ao processo de identificação dos melhores inputs para variáveis de decisão de sistemas simulados (JALALI; NIEUWENHUYSE, 2015). De acordo com Fu (2002) o processo de OvS pode ser divido em duas fases: gerar possíveis candidatas a solução e avaliar estas soluções por meio de estimativas do modelo de simulação. A Figura 2 esboça a perspectiva de Fu (2002) para este processo.

Figura 2: Processo de otimização via simulação



Fonte: Adaptado de Fu (2002)

2.2. Metamodelagem por *gradiente-boosted trees*

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.

Jalali e Nieuwenhuyse (2015) explica que o metamodelo busca uma aproximação da função objetivo através da captura da relação existentes entre os *inputs x* e os *outputs y* do modelo de simulação, tendo como principal vantagem a velocidade de execução. Ao longo dos anos, diversos algoritmos foram desenvolvidos a fim de melhorar a performance do metamodelo, tais como *Kriging*, Metodologia de Superfície de Resposta, *Splines*, Redes Neurais, entre outros (BARTON, 2009). Em termos de técnicas mais modernas, diversos autores destacam o GBT como sendo um promissor algoritmo para construção de metamodelos (DE LA FUENTE; SMITH, 2017; GANJISAFFAR; CARUANA; LOPES, 2011)

Boosting é uma método proposto inicialmente por Schapire (1990) e visa construir um modelo de alta acurácia com base na convergência de algoritmos com baixa aprendizagem, que no caso de GBT são árvores de decisão. Friedman (2002) propôs uma variação deste método chamado Stochastic Gradient Boosting, sendo introduzido estocasticidade no procedimento proposto por Schapire (1990). Nele, Friedman (2002) ensina que a cada interação do algoritmo as árvores são treinadas utilizando uma amostra aleatória $\widetilde{N}$ dos dados de treino, sem reposição, de modo que $\widetilde{N} < N$, onde N é o tamanho dos dados de treino. O algoritmo GBT se diferencia de métodos

de *bagging*, pois o primeiro visa construir modelos (árvores) sequencialmente, enquanto o segundo é ajustado com base em modelos construídos em paralelo (XIA et al., 2017).

Este algoritmo tem como base conjuntos de árvores de decisão, sendo que a cada interação uma nova árvore é gerada com base nos resíduos das anteriores (DE LA FUENTE; SMITH, 2017). Dado um conjunto de dados $\{x_i; y_i\}$ de tamanho N, os atributos de entrada são representados por x_i e y_i é a variável de resposta, sendo o objetivo do algoritmo determinar a função F*(x) que expressa a relacionamento entre *x* e *y*, de forma a minimizar a função de perda Eq. (1). Este relacionamento aproximado de forma aditiva pela Eq.(2) (XIA et al., 2017).

$$F^*(x) = \arg min_f \Psi\left(y, F(x)\right) \tag{1}$$

$$F_m^*(x) = \sum_{m=0}^{M} \beta_m h(x; \Theta_m) \tag{2}$$

Sendo os *base learners* ideais ($h(x; \Theta_m)$) representados por $\beta_m h(x; \Theta_m)$, sendo β_m e Θ_m os coeficientes e parametros ótimos do *base learner*, respectivamente. Frieman (2002) apresenta o pseudocódigo para o GBT, assim como descrito no Algoritmo 1:

Algoritmo 1: *Stocastic Gradient-boosted tree*

1 $F_0(x) = arg\, min_\gamma \Psi\left(y_i, \gamma\right)$

2 $For\ m = 1\ to\ M\ do$:

3 $\{\pi(i)\}_1^N = rand_{perm}\{i\}_1^N$

4 $\tilde{y}_{\pi(i)m} = -\left[\frac{\partial \Psi\left(\left(y_{\pi(i)}, F\left(x_{\pi(i)}\right)\right)\right)}{\partial F\left(x_{\pi(i)}\right)}\right]_{F(x)=F_{m-1}(x)}, i = 1, \tilde{N}$

5 $\{R_{lm}\}_1^N = L - terminal\ node\ tree(\{\tilde{y}_{\pi(i)m}, x_{\pi(i)}\}_1^{\tilde{N}})$

6 $\gamma_{lm} = arg\, min_\gamma \sum_{x_{\pi(i)} \in R_{lm}} \Psi\left(y_{\pi(i)}, F_{m-1}\left(x_{\pi(i)}\right) + \gamma\right)$

7 $F_m(x) = F_{m-1}(x) + \nu\, .\gamma_{lm}\, l(x \in R_{lm})$

8 *End For.*

Fonte: Friedman (2002)

Sendo: $\Psi\left(y_i, \gamma\right)$ representa a função de perda, R_{lm} representa cada região formada por l nós das m-ésimas árvores e $\pi(i)\}_1^N$ a permutação dos inteiros (1, ... N) dos dados de treino $\{y_i, x_i\}_1^N$. A linha (4) denota o gradiente da função de perda (ou pseudo-resíduos) da última interação.

γ_{lm} é definido como o conjunto de coeficientes de expansão ótimos que correspondem a cada nó da árvore na m-ésima interação. Um importante parâmetro do GBT é a taxa de aprendizagem (v), que controla o fenômeno de *overfitting* do modelo (XIA et al., 2017; FRIEDMAN, 2002). Ganjisaffar, Caruana e Lopes (2011) sugerem que os principais parâmetros a serem definidos pelo usuário, assim como suas amplitudes, são: o número de folhas por árvore = [2; 25] ; a porcentagem mínima de observações por folha = [0,12; 0,50]; e a já comentada taxa de aprendizagem [0,05; 0,30].

No processo de construção do metamodelo, a definição correta dos hiper-parâmetros é imprescindível para o sucesso do projeto. Neste sentido diversos métodos de otimização foram utilizados com o objetivo de encontrar o melhor conjunto de parâmetros, destacando os métodos heurísticos como algoritmo genético e colônia de formigas. Embora velozes, um ponto negativo é que apresentam certa dificuldade em encontrarem os ótimos globais. A fim de resolver este dilema, a literatura sugere o uso do método *Grid Search*, que apesar de mais lento apresenta grande capacidade de encontrar o ótimo global do problema (HUANG; MAO; LIU, 2012). Neste método é testado todas possíveis combinações do hiper-parâmentro $\{a_1, ..., a_i\}$ do algoritmo, sendo que cada solução é avaliada recursivamente pela técnica de *cross-validation.* Mais detalhes sobre esse método pode ser observado em Syarif, Prugel-Bennett e Wills (2016).

3. Método de pesquisa

Em termos de pesquisa em engenharia de produção, um trabalho pode ser classificado quanto a sua natureza, objetivos, abordagem e método (TURRIONI; MELLO, 2012). Com base nessa classificação, esta pesquisa pode ser considerada de natureza aplicada, objetivo normativo que utiliza uma abordagem quantitativa regida pelo método de modelagem e simulação.

O objetivo desta pesquisa é avaliar a aplicabilidade do GBT para metamodelagem de um problema real trabalhado no grupo de pesquisa dos autores. O objeto de estudo em questão é uma célula de produção do setor de telecomunicações e será descrito na próxima seção deste artigo.

Seguindo a estratégia proposta por Barton (2009), esta pesquisa se dará em seis etapas principais, a saber: Modelar o metamodelo; Planejar o experimento de geração da base de dados para treino; conduzir o experimento no modelo de simulação; otimizar os hiper-parâmetros do

metamodelo; treinar, validar e aplicar o metamodelo; verificar no modelo de SED a resposta ótima do metamodelo.

Quanto as métricas de desempenho do metamodelo, optou-se pelo uso do Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) e a Correlação Quadrática (R^2). Tomando y_i como o valor observado (real) no ponto i, $\hat{y}_i$ o valor previsto, e n o tamanho amostral, a formulação matemática para o RMSE, MAE, MAPE (BERGMEIR; BENÍTEZ, 2012; CHAI; DRAXLER, 2014) e R^2 (PEREIRA et al., 2020) estão representadas nas Eqs. (3-6), respectivamente.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|100\ \frac{y_i - \hat{y}_i}{y_i}\right| \tag{5}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{6}$$

Sendo:

$$\bar{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{7}$$

Para criação do modelo de SED será utilizado o software de simulação ProModel® (versão 18.2), sendo utilizado o otimizador SimRunner® para geração do valor benchmark. Para construção do metamodelo, escolheu-se o software RapidMiner® (versão 9.6.000), pois o mesmo possui características importantes a este trabalho, como operadores de algoritmos de ML, otimização e processamento paralelo. A máquina escolhida para realização dos experimentos foi um Notebook Dell Inspirion Série 3000 com 8 GB de RAM equipado com processador Intel i5 7200U de 4 *threats* e 2 núcleos com velocidade de 2.70 GHz cada.

4. Discussão e resultados

4.1. Apresentação do objeto de estudo

O objeto de estudo selecionado representa uma célula produtiva do setor de telecomunicações. A escolha deste case deu-se por ser um sistema real, complexo e com alto nível de manufatura, esboçando assim alta variabilidade em seus processos produtivos, além de variáveis de decisão com diferentes níveis. Neste estudo foram utilizadas 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 e x_7).

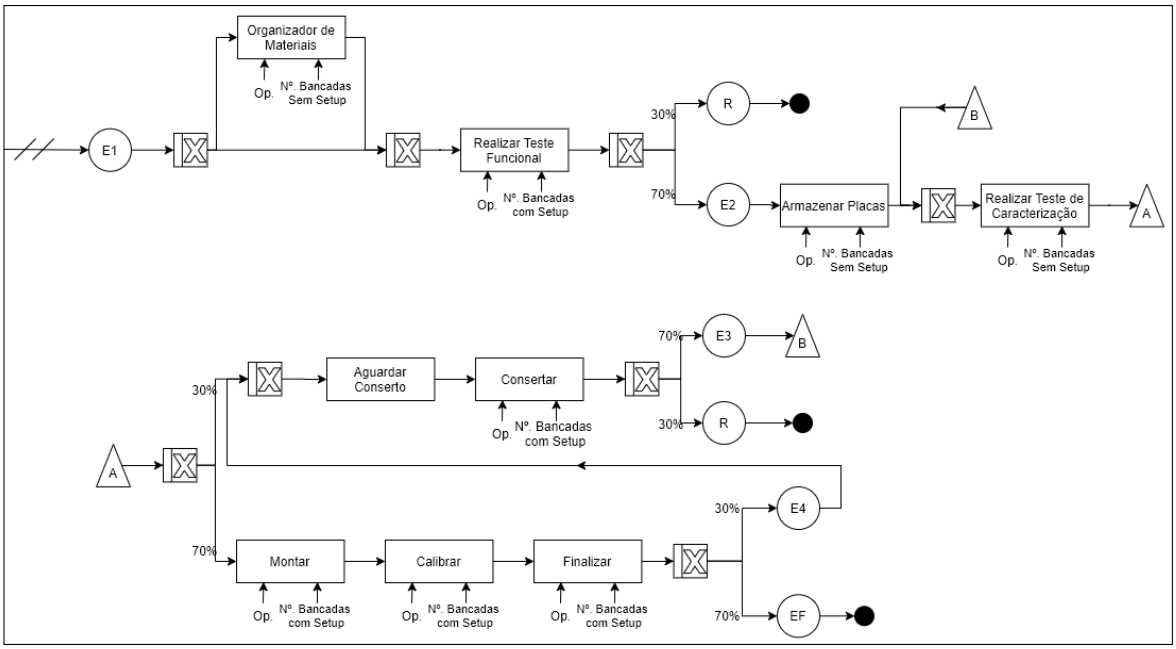
O objetivo da otimização é encontrar o valor que maximiza o lucro (y_i) da empresa. Dadas as variáveis de decisão e sua amplitude de variação, apresentadas na Tabela 1, o problema possui um espaço de solução de 127.000 configurações possíveis, e, considerando que para se gerar uma solução (com 30 replicações) demanda-se um tempo computacional de 33 segundos, a geração de todo espaço de solução gastaria 4.191.000 segundos, ou 48,51 dias.

Tabela 1: Variáveis de decisão do problema

	Variável de Decisão	Tipo	Limite Inferior	Limite Superior
x_1	Número de operadores	Inteiro	1	3
x_2	Número de bancadas com setup	Inteiro	1	4
x_3	Número de bancadas sem setup	Inteiro	1	4
x_4	Atividade organizar material	Binário	0	1
x_5	Tamanho do estoque intermediário 1	Inteiro	5	15
x_6	Tamanho do estoque intermediário 2	Inteiro	5	15
x_7	Tamanho do estoque intermediário 3	Inteiro	5	15

A Figura 3 representa a modelagem conceitual da célula produtiva estudada sob a ótica da técnica IDEF-SIM (MONTEVECHI et al., 2010).

Figura 3: IDEF-SIM do objeto de estudo



A fim de se estabelecer um valor *bechmark* a ser utilizado na avaliação dos metamodelos, foi construído um modelo de otimização com um horizonte de simulação de um mês e com um número de replicações igual a 30. A formulação matemática do problema objetiva maximizar o lucro, Eq. (8), sujeito as restrições (Eq. 9) dos valores das variáveis de decisão x_i, com $i \in I$ (1, ..., 7).

$$MaxE\left\{f(\tilde{\phi}(\sum_{i=1}^{7} x_i))\right\} \tag{8}$$

Sujeito a:

$$\begin{gathered} L_i \leq x_i \leq U_i, \quad \forall i \in I \\ x_i \in I, \quad \forall i \in I \end{gathered} \tag{9}$$

Sendo:

x_i = Quantidade do recurso i;

E = Valor esperado da função objetivo comtemplando o lucro;

f = Valor do vetor função objetivo;

ϕ = Modelo estocástico representado pelo modelo de simulação;

L_i = Limite inferior de variação do recurso i, L_i = [1 1 1 0 5 5 5];

U_i = Limite superior de variação do recurso i, U_i = [3 4 4 1 15 15 15].

Para execução da OvS utilizou-se o software comercial SimRunner®. A otimização deu-se no modo "moderado" do software necessitando 245 interações e um tempo de 8.087 segundos para convergir em um valor de ótimo, assim como mostrado na Figura 4. A solução ótima encontrada pode ser observada na Tabela 2, refletindo em um lucro médio de 54.933,00 reais.

Figura 4: Gráfico interação versus lucro

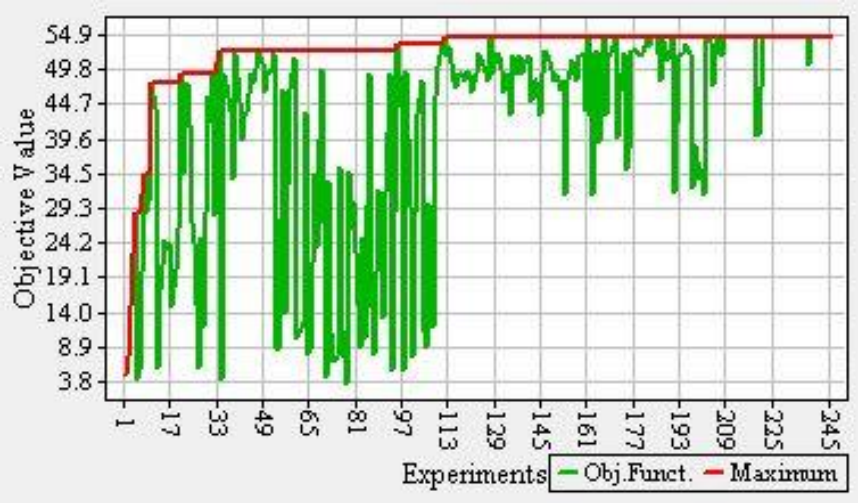


Tabela 2: Solução ótima pelo SimRunner®

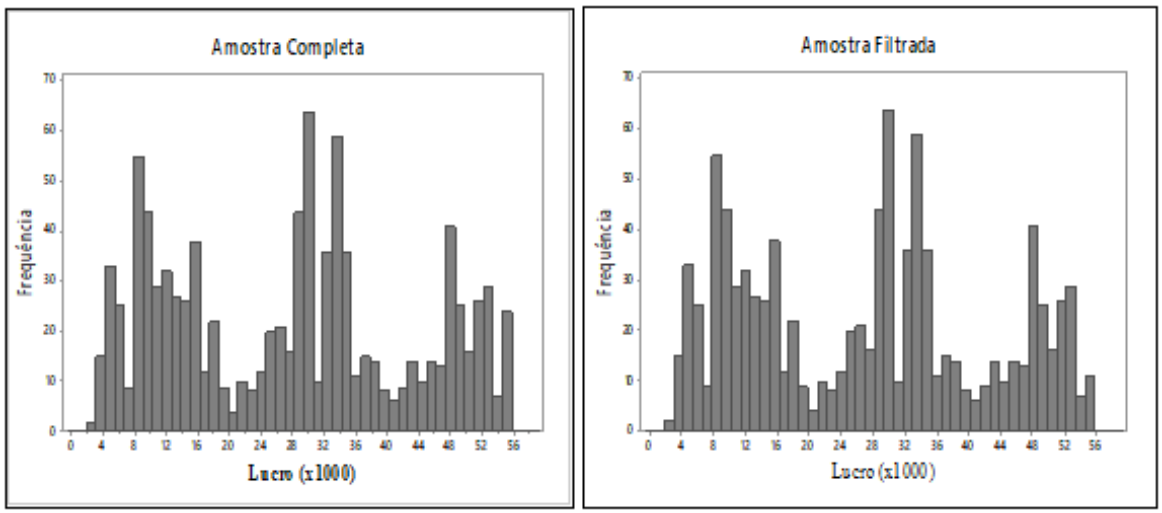
Variável de Decisão		Tipo	Solução Ótima
x_1	Número de operadores	Inteiro	3
x_2	Número de bancadas com setup	Inteiro	3
x_3	Número de bancadas sem setup	Inteiro	2
x_4	Atividade organizar material	Binário	1
x_5	Tamanho do estoque intermediário 1	Inteiro	12
x_6	Tamanho do estoque intermediário 2	Inteiro	13
x_7	Tamanho do estoque intermediário 3	Inteiro	5
Lucro Total (mil R$)			54.930,00
IC 95% para o Lucro			(53.080,00 ; 56.780,00)
Tempo Gasto na Otimização (s)			8.087,45

4.2. Construção e aplicação do metamodelo

A primeira etapa na construção dos metamodelos é a geração dos dados para treinamento. Neste trabalho foi experimentado um total de 1.000 cenários retirados aleatoriamente do espaço de

solução, demandando um tempo computacional de 33.010 segundos para simulação destes cenários, com 30 replicações cada. Após as experimentações foi retirado da base de treino as soluções com respostas iguais ou maiores que o valor *benchmark* encontrado na otimização, restando 987 dados para treinamento dos algoritmos. A Figura 5 contém o histograma da resposta para os cenários experimentados com e sem o filtro aplicado.

Figura 5: Histograma das respostas dos cenários



A construção do metamodelo engloba duas fases principais: a primeira diz respeito ao treinamento e otimização dos hiper-parâmetros do algoritmo de modo a minimizar o RMSE, com base na amplitude de variação dos parâmetros identificados na literatura. A Tabela 3 contém os ranges para cada parâmetro, assim como seu valor ótimo obtido pela otimização via *grid search*, assim como o tempo dispendido para treinamento e otimização do metamodelo (T - Opt).

Tabela 3: Seleção dos hiper-parâmetros do GTB

Hiper-parâmetro	Variação	Nº. Steps (Linear)	Valor Ótimo
Nº de folhas por árvore	[2; 25]	10	18
% mín. de observações por folha	[0,12; 0,50]	10	0,196
Taxa de aprendizado	[0,05; 0,30]	10	0,225
T – Opt (s)	-	-	1.026

Tendo em mãos os parâmetros ótimos do metamodelo o próximo passo foi a sua aplicação para previsão dos resultados de todo espaço de soluções para o *case* trabalhado. A Tabela 4 apresenta o valor lucro para solução ótima encontrada pelo metamodelo (média das replicações e IC 95% resultantes do modelo de simulação), as métricas de erro RMSE, MAE, MAPE e R^2, assim como os tempos dispendidos para otimização e treinamento (T – Opt.) e previsão (T – Prev.).

Tabela 4: Aplicação do metamodelo ao objeto de estudo

Descrição	**Metamodelo**	***Benchmark* (SimRunner)**
RMSE	0,1788	-
MAE	0,143	-
MAPE	0,984%	-
R^2	0,999	-
T-Opt. (s)	1.026	8.087,45
T-Prev. (s)	3	-
Solução ótima	(3; 3; 2; 1; 8; 10; 5)	(3; 3; 2; 1; 12; 13; 5)
Lucro (previsto)	54.630,00	-
Lucro (observado)	54.930,00	54.930,00
IC95%	(53.080,00 ; 56.780,00)	(53.080,00 ; 56.780,00)

Com base nas informações da Tabela 4 pode-se observar que o metamodelo proposto foi capaz de atingir o valor *benchmark*, mostrando a potencialidade do uso do GBT em problemas de OvS. Além disto, o metamodelo demandou um tempo de 1.029 segundos para otimização, treinamento e previsão de 100% do espaço de solução do problema. Atrelado a isto, ressalta-se que, uma vez treinado o metamodelo ele pode ser utilizado inúmeras vezes demandando apenas o tempo de previsão, que para este caso foi de 3 segundos. Isto é importante em aplicações que necessitam de uma boa resposta em um intervalo de tempo que torna proibitivo o uso de técnicas convencionais de OvS, como os emergentes modelos de gêmeos digitais (SALAMA; ELTAWIL, 2018).

Este método também se mostra benéfico em aplicações que necessitam trabalhar com o ótimo global, sendo assim possível a avaliação de todo espaço de busca. Para isto, estima-se que a geração das 127.000 soluções possíveis pelo modelo de simulação demandaria 1.164,16 horas, inviabilizando sua aplicação. Todavia com o uso do metamodelo demandou-se apenas 1.029 segundos, que se somados aos 33.000 segundos necessários para geração da base de treino,

conseguiu prever o lucro dos 127.000 cenários em um tempo 99,18% inferior se comparado a simulação.

5. Conclusão

O principal intuito desta pesquisa foi avaliar a utilização do GBT para metamodelagem em problemas de OvS. Para tanto o método proposto foi aplicado em um modelo de SED que representa um sistema real da indústria de telecomunicações, cujo organização necessita decidir quantos funcionários alocar em cada atividade e definir o tamanho de cada estoque, sendo um problema encontrado em empresas de diversos setores.

O problema estudado possuía um total de 127.000 soluções possíveis, que para a avaliação completa do espaço de soluções pelo modelo de simulação demandaria aproximadamente 48,5 dias. Todavia, com o método proposto nesta pesquisa, foi possível prever o resultado para os 127.000 cenários em apenas 3 segundos, que se somados aos 1.029 segundos demandados para otimização e treinamento do metamodelo e aos 33.000 segundo gastos para geração da base de treino, apresenta uma redução de 99,18% se comparado à simulação de todo o espaço de busca do problema.

Quando se trata da performance do metamodelo, verifica-se que o algoritmo GBT apresentou um ajuste com $R^2 > 0,999$, sendo considerado "quase perfeito" pela literatura. Além disto, após o ranqueamento das previsões e avaliação da melhor resposta no modelo de simulação, observou-se uma solução com o lucro de 54.930,00 reais, exatamente o mesmo valor de *benchmark* encontrado pelo otimizador SimRunner, demostrando assim a potencialidade do uso de metamodelos com base no algoritmo GBT para otimização de modelos de SED.

Por fim, em respeito a recomendações para trabalhos futuros, sugere-se:

- Uso de metamodelos baseados em outros algoritmos de ML para OvS;
- Aplicação do método proposto em outros objetos de estudo, a fim de validar ou refutar sua aplicabilidade;
- Investigação da integração entre metamodelagem e outras técnicas emergentes, tais como a DEA e gêmeos digitais.

6. Agradecimentos

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

REFERÊNCIAS

BARTON, Russell R. *SIMULATION OPTIMIZATION USING METAMODELS*. *Winter Simulation Conference*. Austin, TX: [s.n.], 2009.

BERGMEIR, Christoph; BENÍTEZ, José M. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, v. 191, p. 192–213, 2012. Disponível em: <http://dx.doi.org/10.1016/j.ins.2011.12.028>.

CHAI, T.; DRAXLER, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, v. 7, n. 3, p. 1247–1250, 2014.

DE LA FUENTE, Rodrigo; ERAZO, Ignacio; SMITH, Raymond L. *Enabling intelligent processes in simulation utilizing the tensorflow deep learning resources*. *Winter Simulation Conference*. National Harbor, Maryland: IEEE, 2019.

DE LA FUENTE, Rodrigo; SMITH, Raymond. *METAMODELING A SYSTEM DYNAMICS MODEL: A CONTEMPORARY COMPARISON OF METHODS*. *Winter Simulation ConferenceWinter Simulation Conference*. Las Vegas, NY: [s.n.], 2017.

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

FU, Michael C. Feature Article: Optimization for simulation: Theory vs. Practice. *INFORMS Journal on Computing*, v. 14, n. 3, p. 192–215, 2002.

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*, n. c, p. 85–94, 2011.

HELLENO, A. L. *et al.* Integrating value stream mapping and discrete events simulation as decision making tools in operation management. *International Journal of Advanced Manufacturing Technology*, v. 80, n. 5–8, p. 1059–1066, 2015.

HUANG, Qiujun; MAO, Jingli; LIU, Yong. An improved grid search algorithm of SVR parameters optimization. *Applied Mechanics and Materials*, n. 2, p. 1022–1026, 2012.

JALALI, Hamed; NIEUWENHUYSE, Inneke Van. Simulation optimization in inventory replenishment: A classification. *IIE Transactions (Institute of Industrial Engineers)*, v. 47, n. 11, p. 1217–1235, 2015.

LAW, Averill M. *Simulation Modeling and Analysis, FIFTH EDITION*. [S.l: s.n.], 2013. Disponível em: <www.averill-law.com>.

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*, v. 262, n. 2, p. 673–681, 2017.

MONTEVECHI, Jose Arnaldo B *et al.* IDENTIFICATION OF THE MAIN METHODS USED IN SIMULATION

PROJECTS. 2015, [S.l.]: Proceedings of the 2015 Winter Simulation Conference, 2015. p. 3469–3480.

MONTEVECHI, José Arnaldo Barra *et al. APPLICATION OF DESIGN OF EXPERIMENTS ON THE SIMULATION OF A PROCESS IN AN AUTOMOTIVE INDUSTRY*. *Winter Simulation Conference*. [S.l: s.n.], 2007.

MONTEVECHI, José Arnaldo Barra *et al. CONCEPTUAL MODELING IN SIMULATION PROJECTS BY MEAN ADAPTED IDEF: AN APPLICATION IN A BRAZILIAN TECH COMPANY*. *Proceedings - Winter Simulation Conference*. [S.l: s.n.], 2010.

PEREIRA, Leandro Duarte *et al.* Short-term forecasting models for automated data backup system : segmented regression analysis. *Acta Scientiarum*, n. 2007, p. 1–13, 2020.

SALAM, Mohammad Asif; KHAN, Sami A. Simulation based decision support system for optimization: A case of Thai logistics service provider. *Industrial Management and Data Systems*, v. 116, n. 2, p. 236–254, 2016.

SALAMA, Shady; ELTAWIL, Amr B. A Decision Support System Architecture Based on Simulation Optimization for Cyber-Physical Systems. *Procedia Manufacturing*, v. 26, p. 1147–1158, 2018. Disponível em: <https://doi.org/10.1016/j.promfg.2018.07.151>.

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

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

SOUSA JUNIOR, Wilson Trigueiro De *et al.* Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. *Computers and Industrial Engineering*, v. 128, n. January, p. 526–540, 2019. Disponível em: <https://doi.org/10.1016/j.cie.2018.12.073>.

SYARIF, Iwan; PRUGEL-BENNETT, Adam; WILLS, Gary. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika (Telecommunication Computing Electronics and Control)*, v. 14, n. 4, p. 1502–1509, 2016.

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*, v. 1, n. Programa Pós-graduação em Eng. Produção, p. 191, 2012.

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

XU, Jie *et al.* Simulation optimization in the era of Industrial 4.0 and the Industrial Internet. *Journal of Simulation*, v. 10, n. 4, p. 310–320, 2016.