SUPPORT VECTOR MACHINE PARA METAMODELAGEM EM PROBLEMAS DE OTIMIZAÇÃO VIA SIMULAÇÃO

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

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

RESUMO

Com o advento da indústria 4.0 os sistemas produtivos tendem a necessitar cada vez mais de ferramentas que possibilitem a análise e tomada de decisões com o menor tempo possível. Neste sentido, ferramentas de otimização via simulação baseadas em metamodelos surgem como uma promissora alternativa aos métodos convencionais de otimização em simulação. Esta pesquisa busca avaliar a integração entre as técnicas de *machine learning*, metamodelagem e OvS para solução de problemas reais da indústria, estudando a aplicabilidade do algoritmo *Support Vector Machine* para metamodelagem para otimização em um modelo de simulação a eventos discretos. Na construção, treino e otimização do metamodelo foi possível observar uma redução de 99,08% no tempo demandado para geração de todo espaço de soluções, encontrando um valor para a função objetivo superior ao *benchmark* gerado pelo *software* de otimização SimRunner®.

PALAVRAS CHAVE. **Otimização via simulação. *Machine learning*. Simulação a eventos discretos.**

Tópicos: Simulação, Inteligência Computacional, PO na Administração e Gestão da Produção

ABSTRACT

With the advent of industry 4.0, production systems tend to increasingly need tools that enable analysis and decision making in the shortest possible time. In this sense, optimization via simulation (OvS) based on metamodels emerge as a promising alternative to conventional simulation optimization methods. This research seeks to evaluate the integration between machine learning, metamodeling and OvS techniques for solving real industry problems, studying the applicability of the Support Vector Machine algorithm for metamodeling for optimization in simulation to discrete events. The construction, training, and optimization of the metamodel made it possible to observe a reduction of 99.08% in the time required to generate the entire space of solutions, finding a value for the objective function higher than the benchmark generated by the SimRunner® optimization software.

KEYWORDS. Optimization via simulation. Machine learning. Discrete event simulation.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Paper topics: Simulation, Computational Intelligence, PO in Administration and Production Management

1. Introdução

O campo da otimização via simulação progrediu significativamente na última 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].

A OvS refere-se ao processo de identificação dos melhores *inputs* para variáveis de decisão de sistemas simulados [Jalali e Nieuwenhuyse 2015]. De acordo com [Fu 2002], o processo de OvS pode ser divido em duas fases: 1) geração de possíveis candidatas à solução do problema, e 2) avaliação destas soluções por meio de estimativas em um modelo de simulação. Dentre as técnicas de otimização, a metamodelagem vem se destacando nos últimos anos devido a sua capacidade de gerar bons resultados em um tempo computacional consideravelmente menor que por métodos convencionais. Segundo De La Fuente et. al [2019], metamodelagem refere-se ao processo de geração de um modelo simplificado para um outro modelo de maior complexidade. Para estes autores, a evolução na área de *Machine learning* (ML) vem proporcionando a criação de metamodelos que esbocem maior precisão e velocidade.

Em novembro de 2019 foi desenvolvida uma análise bibliométrica nas bases de dados *Scopus* e *Web of Science*, pesquisando-se pelos termos correlatos em inglês referentes a: Otimização via simulação, metamodelagem, *machine learning*, bem como a combinação destes termos. Os resultados encontrados são apresentados na Tabela 1.

Tabela 1: Análise bibliométrica

Base de dados	"*Simulation optimization*" ou "*simulation-based optimization*" ou "*optimization via simulation*"	"*Metamodel*"	"*Machine Learning*"	"*Simulation Optimization"* ou *"Simulation-Based Optimization"* ou *"Optimization via Simulation"* e "*Metamodel*" e "*Machine Learning*"
Web of Science	3.762	4.120	100.791	2
Scopus	5.637	6.864	159.983	4

Nota-se, pela Tabela 1, um número expressivo de publicações relacionados termos separadamente. Todavia observa-se que, quando se procura pela intersecção entre estas áreas de pesquisa, ainda há um número reduzido de trabalhos publicados, revelando uma lacuna a ser explorada na literatura. Jalali e Nieuwenhuyse [2015] ressaltam, que embora a metamodelagem seja menos popular que a utilização de metaheurísticas ou métodos híbridos, ela oferece meios poderosos para solucionar problemas de OvS robustos.

Nos últimos anos a utilização de metamodelos como saída para problemas de OvS complexos tem crescido, como é observado nos trabalhos de Hannah, Powell e Dunson [2014], que propuseram um método de OvS por metamodelagem baseada em regressão semi-convexa, e De La Fuente e Smith [2017], que compararam diversos métodos de metamodelagem para OvS em um hospital. Também no setor de *health care*, Yousefi et al. [2018] utilizaram métodos de *ensemble* em um modelo de simulação baseada em agentes para redução do tempo médio de permanência de pacientes, e Yousefi e Yousefi [2019] aplicaram a mesma metodologia para alocação ótima de recursos, com a finalidade de reduzir o tempo médio de *Door to Doctor* em uma emergência

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

hospitalar. Souza Junior et al. [2019] estudaram a integração entre metamodelagem por ML, processamento paralelo e metaheurísticas aplicados em problemas de engenharia de produção. E Woldemariam e Lemu [2019] utilizaram metamodelagem por redes neurais e algoritmo genético em OvS para análise dinâmica de fluidos

Visto isto, este tema justifica-se por estudar a integração entre técnicas de ML e OvS, áreas ainda pouco exploradas na literatura como comprovadas pela análise bibliométrica. Neste sentido, o presente trabalho visa explorar a utilização do algoritmo de ML *Support Vector Machine* (SVM) para a otimização via simulação de problemas complexos de simulação a eventos discretos.

O presente trabalho contribui com a literatura por meio da integração de técnicas de ML com a OvS como estratégia para redução do tempo computacional dispendido na execução da otimização. Além disto, este trabalho reflete sua contribuição tecnológica no que tange ao desenvolvimento de um método de OvS que poderá ser utilizado na solução de problemas complexos no contexto industrial. Desta forma, espera-se que esta pesquisa sirva de base para que pesquisadores e práticos possam explorar a potencialidade da integração entre estes dois campos de pesquisa (Simulação e Inteligência Artificial).

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 Otimização via simulação

De acordo com Lin e Chen [2015], a otimização via simulação é uma poderosa ferramenta de tomada de decisões, pois esta técnica é capaz de capturar o relacionamento entre várias entidades de um sistema complexo do mundo real, identificando a melhor solução para o problema. Segundo Souza Junior et al. [2019], a OvS é utilizada para auxiliar tomadores de decisão a investir ou alocar recursos em sistemas de produção novos ou existentes.

Amaran et al. [2016] definem OvS como sendo a otimização de uma função objetivo (FO) sujeita a restrições, a qual é avaliada por meio de um modelo de simulação estocástico. Carson e Maria [1997] e Xu et al. [2016] referem-se a OvS como o processo de identificação dos melhores valores de entrada para variáveis de um sistema simulado, avaliando as soluções por meio de um loop criado entre o algoritmo de otimização e o modelo de simulação.

O campo do estudo da OvS evoluiu significantemente na última década, com o desenvolvimento de diversos algoritmos, softwares e aplicações. Para Amaran et al. [2016], a formulação geral para um problema de OvS, pode ser representado pela Eq. (1):

$$\begin{gathered} \operatorname{Min} \mathbb{E}_{\omega}[f(x,y,\omega)] \\ s.a.\ \ \mathbb{E}_{\omega}[g(x,y,\omega)] \leq 0 \\ h(x,y) \leq 0 \\ x_l \leq x \leq x_u \\ x \in \mathbb{R}^n, y \in \mathbb{D}^m \end{gathered} \tag{1}$$

Sendo: $\mathbb{E}_{\omega}$ representa o valor esperado para a função *f*, que é avaliada através da simulação a partir de entradas contínuas *x*, ou discretas *y*, submetidas a um vetor de números aleatórios ω. Da mesma forma as restrições são definidas pelos valores da função *g*, que são avaliados a cada simulação. O problema ainda pode conter outras restrições (representadas por *h*) que não envolvem variáveis aleatórias, bem como restrições vinculadas as variáveis de decisão.

Em contrapartida, Miranda et al. [2017] ressaltam que em problemas de OvS em sistemas complexos e com espaço de soluções muito grandes, demanda-se um longo tempo computacional para que o algoritmo de otimização convirja para um bom resultado. Barton [2009] cita que estas características levaram pesquisadores a desenvolverem métodos especializados para OvS, dentre

eles, destacam-se: *Ranking* e Seleção [Swisher e Jacobson 1999], *Simulated Annealing* [Kirkpatrick et al. 1983], Algoritmos Genéticos [Holland 1975], Metaheurísticas [Rao 2009], Busca Tabu [Glover et al. 2019] e Metamodelagem [Barton 2009].

2.2. Metamodelagem por *Support Vector Machine*

A metamodelagem é uma abordagem comumente usada para desenvolver um modelo representativo de um modelo de simulação [De La Fuente et al. 2019]. O metamodelo é capaz de capturar a relação entre os valores das variáveis de decisão e os outputs da simulação, provendo uma aproximação da função objetivo em um tempo de execução muito menor que a simulação [Jalali e Nieuwenhuyse 2015].

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: Metodologia de Superfície de Resposta, *Radial Basis Function, Kriging*, Redes Neurais e *Spline*. Entretanto, com o advento da inteligência artificial, muitos pesquisadores buscam incorporar 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].

Proposto por Vapnik [1999], o SVM é um dos métodos de ML mais populares dos últimos anos. Foi inicialmente construído para problemas de classificação, mas também vem sendo muito utilizado em problemas de regressão [Hastie et al. 2008], [Li et al. 2010].

Em problemas de classificação, o método busca o limite que separa os dados de treinamento em suas respectivas classes, com o mínimo de erro possível. Neste caso, é definido o hiperplano com base em vetores construídos pelas observações que estão localizadas nos limites, logo os dados que se encontram em lados distintos do hiperplano são alocados em classes diferentes [Østergard et al. 2018].

Jeng [2006] explica que neste tipo de problema, o SVM tenta encontrar a margem máxima entre as duas classes através da combinação linear dos vetores, sendo solucionado por uma programação quadrática com restrições lineares (QP). Para os autores, com a introdução do parâmetro $\mathcal{E}$, que representa a função de perda, este algoritmo pode ser estendido à solução de problemas de regressão não lineares.

Li et al. [2010] apresenta a formulação matemática para o SVM (Eq. 2):

$$\hat{y} = \hat{f}(X) = w\phi(X) + b \tag{2}$$

Sendo $X \in \Re^k$ é o input k-dimensional, ϕ é a transformação não-linear de $\Re^k$ para $\Re^h$, com h > k, e $\hat{f}(X)$ descreve o hiperplano em $\Re^h$. Os autores ressaltam que os coeficientes w e b são obtidos pela minimização da função de risco, dada pela Eq. (3).

$$R_{svm}(C) = C\sum_{i=1}^{n} C\left(\hat{f}(X_i), y_i\right) + \frac{1}{2}\|w\|^2 \tag{3}$$

Sendo:

$$C\left(\hat{f}(X_i), y_i\right) = \begin{cases} \left|\hat{f}(X) - y\right| - \mathcal{E} & \text{se } \left|\hat{f}(X) - y\right| \geq \mathcal{E} \\ 0 & \text{Caso Contrário} \end{cases} \tag{4}$$

Solucionado o problema de minimização, a função de previsão final do método SVM é obtida pela Eq. (5) [Vapnik 1999].

$$f(X, a_i, a_i^*) = \sum_{i=1}^{n} (a_i - a_i^*) K(X, X_i) + b \quad (5)$$

Sendo X o novo *input* do modelo, $K(X, X_i)$ a função Kerniana que representa a distância entre X e X_i e a_i, a_i^* e b são os coeficientes obtidos pela minimização da função de risco [Li et al. 2010]. Oliveira et al. [2010] afirmam que a boa eficiência do SVM depende da escolha correta dos hiper-parâmetros *C*, *epsilon* ($\mathcal{E}$) e Kernel. De acordo com Cherkassky e Ma (2004) pode-se calcular o valor ideal de *C* segundo a Eq. (6):

$$C = \max\left(\left|\bar{y} + 3\sigma_y\right|, \left|\bar{y} - 3\sigma_y\right|\right) \quad (6)$$

No qual $\bar{y}$ e σ_y são a média e o desvio padrão dos valores de y nos dados de treino. Para o cálculo de ε (Eq. 7), os autores recomendam uma primeira execução do algoritmo com os parâmetros *default*, com isto pode-se calcular o erro de previsão δ associado a base de dados, no qual $\delta = \hat{y} - y_i, (i = 1, \ldots, n)$.

$$\mathcal{E} = 3\sigma \sqrt{\frac{\ln n}{n}} \quad (7)$$

Sendo σ o desvio padrão do erro δ, e n o número de exemplos da base de dados. Para o presente estudo, o range escolhido para os parâmetros a serem utilizados são $C = \left\{\left|\bar{y} - 3\sigma_y\right|, \left|\bar{y} + 3\sigma_y\right|\right\}$; $epsilon = \{0, \mathcal{E}\}$; e Kernel {linear, RBF, polinomial} [Cherkassky e Ma 2004].

2.3. Otimização dos hiper-parâmetros via *grid search*

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 [Syarif et al. 2016].

Atualmente existem diversos métodos baseados em heurísticas (ex: algoritmo genético, colônia de formigas, etc.) que esboçam velocidade de execução e capacidade de adaptação, porém são suscetíveis a ficarem presos em ótimos locais. Como alternativa, destaca-se o algoritmo *grid search*, que apesar de ser mais lento, possui maior facilidade em encontrar o ótimo global [Huang et al. 2012]. Este método testa todas combinações possíveis para os parâmetros $\{a_1, \ldots, a_i\}$ do algoritmo de ML, e para cada combinação é utilizada a técnica de *cross-validation* para calcular o erro associado a esta.

2.4. Métricas de performance

Um importante passo na construção de metamodelos é a avaliação dos mesmos, a fim de se escolher o modelo que melhor se adeque aos dados, assim como guiar a seleção e/ou otimização de seus parâmetros. Botchkarev [2019] destaca o *Mean Square Error* (MSE) ou *Root MSE* (RMSE), o *Mean Absolut Error* (MAE) e o *Mean Absolute Percentage Error* (MAPE) como as principais métricas de erro utilizadas na literatura. Segundo Hu, Palta e Shao (2006) outra importante métrica em problemas de regressão é a correlação quadrática (R^2).

Considerando y_i como sendo o valor observado (real) no ponto *i* e $\hat{y}_i$ o valor previsto, e *n* o tamanho dos dados de teste, podemos calcular o RMSE, MAE e MAPE, assim como demostrado nas Eqs. (7-9), respectivamente [Bergmeir e Benítez 2012; Chai e Draxler 2014].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (7)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (8)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|100\ \frac{y_i - \hat{y}_i}{y_i}\right| \quad (9)$$

Quando se trata de métricas de avaliação de metamodelos, não se possui um consenso universal sobre valores para a acurácia de um modelo. Todavia, é plausível definir que um metamodelo com $R^2 > 0{,}90$ é razoável, $R^2 > 0{,}95$ possui boa acurácia e $R^2 > 0{,}99$ é considerado quase perfeito [Østergard et al. 2018]. O R^2 é uma métrica de acurácia com uma amplitude que varia de 0 a 1. Como indicado na Eq.(10), seu cálculo é dado pela divisão da soma dos quadrados dos resíduos $(y_i - \hat{y}_i)^2$ pela soma total dos quadrados $(y_i - \bar{y}_i)^2$ [Pereira et al. 2020].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (10)$$

Sendo:

$$\bar{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i \quad (11)$$

3. Método de pesquisa

Turrioni e Mello (2012) classificam as pesquisas cientificas na área de engenharia de produção em quatro dimensões básicas, a saber: quanto à natureza, objetivos, abordagem e método. Seguindo classificação proposta por estes autores, pode ser considerada de natureza aplicada, objetivo normativo e abordagem quantitativa utilizando-se do método de modelagem e simulação.

Esta pesquisa objetiva avaliar a aplicabilidade da metamodelagem por ML em problemas de OvS, contemplando problemas reais do grupo de pesquisa do pesquisador. O objeto de estudo selecionado corresponde a uma célula de produção de uma empresa do setor de telecomunicações.

Neste trabalho será utilizado o software ProModel® (versão 18.2), para o desenvolvimento do modelo de simulação, já para a criação dos metamodelos optou-se pelo uso do software RapidMiner® (versão 9.6.000), uma vez que o mesmo apresenta interface amigável, processamento paralelo, e uma ampla variedade de algoritmos de ML e operadores de otimização, ambos necessários neste trabalho. 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. Com base na estratégia apresentada por Barton (2009), o presente trabalho será desenvolvido segundo as seguintes etapas:

- Modelagem da SED: Modelar em ambiente de simulação (software ProModel®) um sistema produtivo real;
- Definição do *benchmark*: Realizar uma OvS utilizando-se o software SimRunner® a fim de gerar um valor *benchmark* para a solução ótima;
- Dados para treinamento: Selecionar aleatoriamente, dentre todas soluções possíveis, uma amostra de 1.000 cenários a ser experimentado no ambiente de SED. Neste estudo serão descartadas da base de treino, soluções com yi ≥ valor ótimo encontrado pelo SimRunner®, caso existam;
- Metamodelagem: Construir no software RapidMiner® o metamodelo com base no algoritmo SVM. Além disto, o metamodelo deverá contemplar uma estrutura de

otimização via *grid search* dos hiper-parâmetros de algoritmo, bem como sua validação pela técnica de *cross-validation*, com k = 10;

- Previsão: Aplicar metamodelo com hiper-parâmetros ótimos para prever o valor da função objetivo em todo espaço de solução do problema;
- Encontrar o ótimo: Ranquear as soluções de modo a encontrar a que esboce o maior valor predito e posteriormente testá-la em ambiente de simulação;
- Avaliação: Avaliar o resultado do metamodelo em relação as métricas de erro RMSE, MAE, MAPE e R^2, e a solução ótima encontrada perante o valor do *benchmark*.

4. Resultados e discussões

4.1. Apresentação do objeto de estudo

O objeto de estudo trabalhado nesta pesquisa representa uma célula de produção da indústria de telecomunicações, sendo um modelo real, complexo e com alta variabilidade no processo. Este problema 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). O objetivo da otimização é encontrar o valor que maximiza o lucro (y_1) da empresa.

Dadas as variáveis de decisão e sua amplitude de variação 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. A Figura 1 representa a modelagem conceitual da célula produtiva estudada sob a ótica da técnica IDEF-SIM [Montevechi et al. 2010].

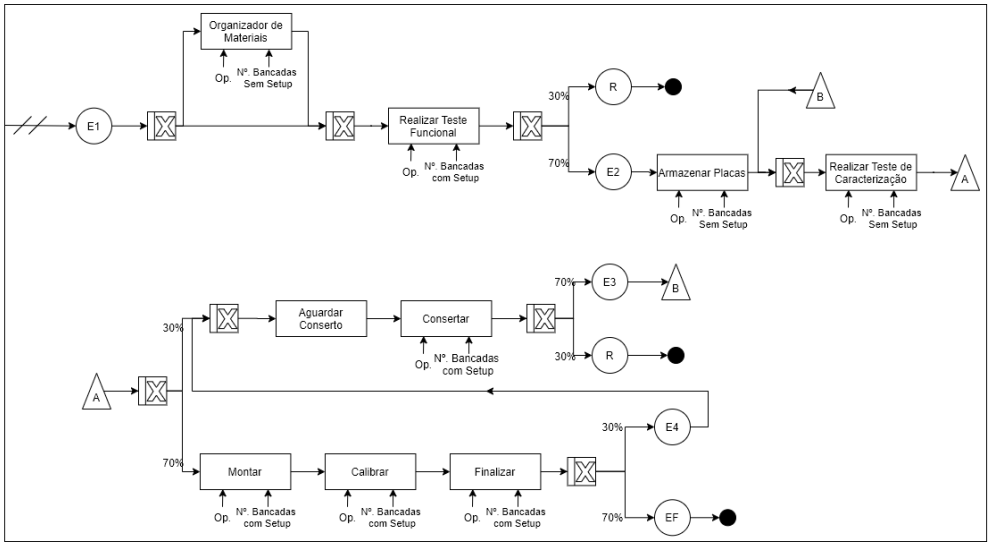


Figura 1: IDEF-SIM do objeto de estudo

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. (12), sujeito as restrições (Eq. (13)) dos valores das variáveis de decisão x_i, com $i \in I$ (1, ..., 7).

$$MaxE\left\{f(\tilde{\phi}(\sum_{i=1}^{7} x_i))\right\} \tag{12}$$

Sujeito a:

$$L_i \leq x_i \leq U_i, \quad \forall i \in I$$
$$x_i \in I, \quad \forall i \in I \tag{13}$$

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*, foram realizadas 245 iterações com um tempo gasto de 8.087 segundos para obtenção de uma solução ótima. A solução ótima encontrada pode ser observada na Tabela 3, refletindo em um lucro médio de 54.933,00 reais. Ressalta-se que, por o *software* utilizar técnicas metaheurísticas, não se garante que este seja o ótimo global do problema, porém uma boa solução que servirá de *benchmark* no momento de avaliação do metamodelo.

Tabela 2: Solução encontrada pelo SimRunner®

	Variável de Decisão	Limite inferior	Limite superior	Solução
x_1	Número de operadores	1	3	3
x_2	Número de bancadas com *setup*	1	4	3
x_3	Número de bancadas sem *setup*	1	4	2
x_4	Atividade organizar material	0	1	1
x_5	Tamanho do estoque intermediário 1	5	15	12
x_6	Tamanho do estoque intermediário 2	5	15	13
x_7	Tamanho do estoque intermediário 3	5	15	5
Lucro Total (R$)			54.933,00	
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 para a construção do metamodelo é a geração da base de dados para treinamento. Neste caso, foram selecionados aleatoriamente um conjunto de 1.000 possíveis cenários, sendo demandado 33.010 segundos para sua simulação. A fim de se evitar o enviesamento dos dados e testar a capacidade do metamodelo em extrapolar o espaço amostral, foram mantidos apenas os cenários que obtiveram um lucro menor que o valor *banchmark* encontrado. O histograma dos valores de lucro em ambos casos estão apresentados na Figura 2.

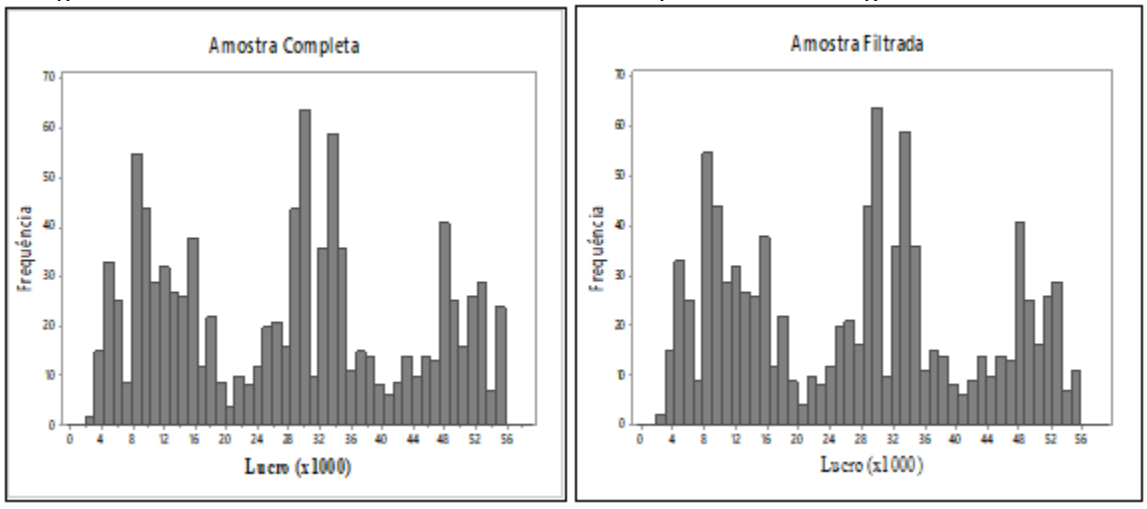


Figura 2: Histograma das respostas dos cenários

A segunda e terceira etapas da construção do metamodelo correspondem, respectivamente, ao treinamento e otimização dos hiper-parâmetros do metamodelo a fim de minimizar o RMSE, e a sua aplicação para previsão do restante do espaço de soluções. A Tabela 3 apresenta os parâmetros ótimos do SVM encontrado via *grid search*, assim como o tempo (T – Opt.) dispendido para a otimização e treinamento. Ressalta-se que os valores para *C* e *epsilon* foram calculados a partir de uma execução preliminar do algoritmo com parâmetros *default.*

Tabela 3: Seleção dos hiper-parâmetros do SVM

Hiper-parâmetro	**Variação**	**Nº. *Steps* (Linear)**	**Valor Ótimo**
C^*	[18,30; 72,68]	10	62,60
Epsilon (ε)*	[0,00; 1,77]	10	0,00
Kernel Type	RBF, polinomial, linear	3	RBF
T – Opt (s)	-	-	6.549

Finalizada a construção do metamodelo com seus parâmetros ótimos, o próximo passo foi a sua aplicação para previsão dos resultados para as 127.000 soluções possíveis. Estas soluções foram ranqueadas e a Tabela 4 apresenta o valor do lucro para melhor solução encontrada pelo metamodelo (média das réplicas 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.), ambos sem considerar o tempo gasto na geração da base de dados no modelo de simulação.

Tabela 4: Aplicação do metamodelo ao objeto de estudo

Descrição	Metamodelo	*Benchmark* (SimRunner®)
RMSE	1,186	-
MAE	0,125	-
MAPE	0,229%	-
R^2	0,994	-
T-Opt. (s)	6.129	8.087,45
T-Prev. (s)	13	-
Solução ótima	(3; 3; 2; 0; 8; 8; 6)	(3; 3; 2; 1; 12; 13; 5)
Lucro (previsto)	59.290,00	-
Lucro (observado)	55.370,00	54.933,00
IC 95%	(53.490,00 ; 57.240,00)	(53.080,00 ; 56.780,00)

O metamodelo construído com base no SVM gastou um tempo de 6.129 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 13 segundos. Isto torna-se importante em aplicações que necessitam de uma boa resposta em um intervalo de tempo que torna proibitivo o uso de técnicas convencionais de OvS, tais como os emergentes modelos de gêmeos digitais [Salama e Eltawil 2018].

Este método também se mostra benéfico em aplicações que necessitam trabalhar com o ótimo global, sendo possível a avaliação de todo o espaço de soluções. 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 SVM demandou-se 6.129 segundos, que se somados aos 33.010 segundos necessários para geração da base de treino, conseguiu prever o lucro máximo dos 127.000 cenários em um tempo 99,06% inferior se comparado a simulação para geração de todo o espaço de busca.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Em termos de assertividade, o método proposto mostrou-se capaz de extrapolar a base de treino, uma vez que o valor da FO encontrada de 55.370,00 reais foi maior que o *beckmark* encontrado pelo otimizador SimRunner®, de 54.930,00 reais.

5. Conclusão

O principal intuito desta pesquisa foi avaliar a utilização do SVM 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, cuja 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 possui 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 13 segundos, que se somados aos 6.129 segundos demandados para otimização e treinamento do metamodelo e aos 33.010 segundo gastos para geração da base de treino, apresenta uma redução de 99,06% se comparado à simulação de todo o espaço de busca.

Quando se trata da performance do metamodelo, verifica-se que o algoritmo SVM apresentou um ajuste com $R^2 > 0{,}994$, sendo considerado "quase perfeito" pela literatura. Em relação ao MAPE, verifica-se que as previsões do metamodelo se diferenciam do valor real, na média, em apenas 0,229%, valor consideravelmente baixo em relação a variabilidade do modelo de simulação. Além disto, após o ranqueamento das previsões e avaliação da melhor resposta no modelo de simulação, observou-se uma solução com o lucro médio de 55.370,00 reais, valor superior ao *benchmark* encontrado pelo otimizador SimRunner®, demostrando assim a potencialidade do uso de metamodelos com base no algoritmo SVM 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, gêmeos digitais e otimização multiobjetivo.

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

Referências

Amaran, Satyajith *et al.* (2016). Simulation optimization: a review of algorithms and applications. *Annals of Operations Research*, 240: 351–380.

Barton, Russell R. (2009) Simulation optimization using meramodels. In *Anais do 2009 Winter Simulation Conference*, p. 230–238. Austin, WSC.

Bergmeir, Christoph e Benítez, José M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, v. 191, p. 192–213.

Botchkarev, Alexei (2019). A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, v. 14, p. 45–76.

Carson, Yolanda e Maria, Anu (1997). Simulation optimization: Methods and applications. In *Anais do 1997 Winter Simulation Conference*, p. 118–126, WSC.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Chai, T. e Draxler, R. R. (2014). 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.

Cherkassky, Vladimir e Ma, Yunqian (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, v. 17, n. 1, p. 113–126.

De La Fuente, Rodrigo, Erazo, Ignacio e Smith, Raymond L. (2019). Enabling intelligent processes in simulation utilizing the tensorflow deep learning resources. In *Anais do* 2019 Winter Simulation Conference, p. 1108-1119, National Habour. IEEE.

De La Fuente, Rodrigo e Smith, Raymond. (2017). Metamodeling a system dynamics model: A comparison methods. In *Anais do 2017 Winter Simulation Conference*, p. 1689-1699, Las Vegas. WSC.

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

Glover, Fred, Laguna, Manuel e Martí, Rafael. (2019). Principles and Strategies of Tabu Search. *Handbook of Approximation Algorithms and Metaheuristics,* 2: 361–377.

Hannah, Lauren A., Powell, Warren B. e Dunson, David B. (2014). Semiconvex regression for meramodeling-based optimization. *Society for Industrial and Applied Mathematics*, v. 24, n. 2, p. 573–597.

Hastie, Trevor, Tibshirani, Robert e Friedman, Jerome. (2008). The Elements of Statistical Learnning. Springer, Stanford.

Holland, John H. (1975). Adaptation in natural and artificial systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge.

Huang, Qiujun, Mao, Jingli e Liu, Yong (2012). An improved grid search algorithm of SVR parameters optimization. *Applied Mechanics and Materials*, n. 2, p. 1022–1026.

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

Jeng, Jin Tsong (2006). Hybrid approach of selecting hyperparameters of support vector machine for regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, v. 36, n. 3, p. 699–709.

Kirkpatrick, Scott *et al.* Optimization by Simulated Annealing. *Science*, v. 220: p. 671 - 680.

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

Lin, James T. e Chen, Chien Ming (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and*

Theory, v. 51, p. 100–114.

Miranda, Rafael de Carvalho *et al.* (2017). 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.

Montevechi, José Arnaldo Barra *et al.* (2010). Conceptual modeling in simulation projects by mean adapted IDEF: An application in a brazilian tech company. In *Anais do Winter Simulation Conference*, p. 3469-3480, Baltimor, WSC.

N. Vapnik, Vladimir (1999). *The Nature of Statistical Learning Theory*. New York, Springer.

Oliveira, Adriano L.I. *et al.* (2010). GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Information and Software Technology*, v. 52, n. 11, p. 1155–1166.

Pereira, Leandro Duarte *et al.* (2020). Short-term forecasting models for automated data backup system : segmented regression analysis. *Acta Scientiarum*, v. 42, p. 1–13

Østergard, Torben, Jensen, Rasmus Lund e Maagaard, Steffen Enersen (2018). A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy*, v. 211, p. 89–103.

Rao, Singiresu S. (2009). *Engineering Optimization: Theory and Practice: Fourth Edition*. JOHN WILEY & SONS, Hoboken.

Salama, Shady; Elatawil, Amr B. (2018). A Decision Support System Architecture Based on Simulation Optimization for Cyber-Physical Systems. *Procedia Manufacturing*, v. 26, p. 1147–1158.

Souza Junior, W. T. De *et al.* (2019). Economic Lot-Size Using Machine Learning, Parallelism, Metaheuristic and Simulation. *International Journal of Simulation Modelling*, v. 18, n. 2, p. 205–216.

Swisher, James R. Lotus e Jacobson, Sheldon H. (1999). A survey of ranking, selection, and multiple comparison procedures for discrete-event simulation. In *Anais do 1999 Winter Simulation Conference*. p. 188–195, Phoenix, WSC.

Turrioni, J. e Mello, C. (2012). 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, Itajubá.

Woldemariam, Endashaw Tesfaye e Lemu, Hirpa G. (2019). A machine learning based framework for model approximation followed by design optimization for expensive numerical simulation-based optimization problems. In *Anais do XXIX International Offshore and Polar Engineering Conference*, p. 943–950, Honolulu, ISOPE.

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

Yousefi, Milad *et al.* (2018). Chaotic genetic algorithm and Adaboost ensemble metamodeling

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

approach for optimum resource planning in emergency departments. *Artificial Intelligence in Medicine*, v. 84, p. 23–33.

Yousefi, Milad e Yousefi, Moslem (2019). Human resource allocation in an emergency department: A metamodel-based simulation optimization. *Kybernetes*, v. 49, n. 3, p. 779–796.