

SIMO: Uma Ferramenta Online para Auxiliar o Ensino de Otimização

André F. R. Malta, Daniel G. de Oliveira, Elias L. S. Júnior, João M. M. C. Cota

Graduandos em Engenharia da Computação

andrimalta@gmail.com, daniel.gdo@hotmail.com,
eliasluizjr@gmail.com, joao_marcos_cota@hotmail.com

André Rodrigues da Cruz

Departamento de Computação e Construção Civil

Centro Federal de Educação Tecnológica de Minas Gerais
R. Dezenove de Novembro, 121, Centro, Timóteo, MG, Brasil
andreacruz@timoteo.cefetmg.br

RESUMO

Este trabalho apresenta a primeira versão do *Sistema Interativo para Métodos de Otimização*, SIMO, um conjunto de ferramentas criado para auxiliar o ensino e aprendizagem de técnicas e algoritmos para otimização. O SIMO possui interface gráfica simples e intuitiva que pode ser acessada gratuitamente no endereço <https://jmmccota.github.io/Otimizacao> através de um computador ou dispositivo móvel. Ele apresenta de forma didática todos os passos da execução dos algoritmos, ressaltando as informações relevantes. Foram implementados algoritmos aos quais incluem variações do Simplex para programação linear, métodos para resolver problemas de transporte e problemas de programação inteira usando a técnica Branch-and-Bound. Além de apresentar a ferramenta, são ilustrados alguns exemplos do uso do SIMO. O sistema se mostra como uma opção de material didático de apoio para estudantes, já que não requer o conhecimento de uma linguagem de modelagem, como ocorre em outras ferramentas.

PALAVRAS CHAVE. Sistema Web de Apoio ao Ensino. Solucionador Online. Programação Linear. Problema de Transporte. Programação Inteira.

Áreas Principais: EDU – PO na Educação, PM – Programação Matemática, OA – Outras aplicações em PO.

ABSTRACT

This paper presents the first version of *Sistema Interativo para Métodos de Otimização*, SIMO, a toolset created to assist in teaching and learning of techniques and algorithms for optimization. SIMO has a simple and intuitive graphical user interface which can be freely accessed at the address <https://jmmccota.github.io/Otimizacao> through a computer or a mobile device. It features in a didactic way all the steps of the execution of algorithms, highlighting relevant information. The algorithms that have been implemented include variations of Simplex for linear programming, methods for solving transportation problems and integer programming problems using the Branch-and-Bound technique. In addition to present the tool, this paper also illustrates some examples of the use of SIMO. The system shows itself as an option of support learning resource for students, since it does not require knowledge of a modelling language as in other tools.

KEYWORDS. Web System to Support Teaching. Online Solver. Linear Programming. Transportation Problem. Integer Programming.

Main Area: EDU – OR in Education, PM – Mathematical Programming, OA – Other applications in OR.

1. Introdução

A Pesquisa Operacional (PO) é uma ciência aplicada voltada para a resolução de problemas reais. Ela aplica conceitos e métodos de diversas áreas científicas na concepção, planejamento ou operação de sistemas para auxiliar a tomada de decisões. As técnicas de PO são utilizadas para avaliar linhas de ação alternativas e encontrar as melhores soluções que servem aos objetivos dos indivíduos ou organizações [SOBRAPO, 2014]. Isto possibilita, por exemplo, a escolha da melhor opção viável em um conjunto de opções disponíveis. Diversas são as áreas de aplicação da mesma, como por exemplo, logística, engenharia, medicina, gestão empresarial, economia, transporte e telecomunicações. No que se refere a problemas de otimização, dependendo da estrutura do problema, uma estratégia adequada deve ser escolhida para investigar a solução ótima¹. Para isto, deve-se definir o conjunto de variáveis de decisão com os respectivos tipos e limites de intervalo, a função objetivo que deve ser maximizada ou minimizada, e o conjunto de restrições funcionais [Hillier e Lieberman, 2013].

Para fins didáticos, os problemas de otimização com poucas variáveis e restrições podem ser resolvidos manualmente em um tempo hábil. Porém, quando a dimensão do problema é maior, a busca da solução ótima deve ser realizada através de um solucionador computacional. O solucionador é um software, com algoritmos especializados, para resolver problemas de otimização [Hillier e Lieberman, 2013]. Os solucionadores podem ser encontrados em versões comerciais, livres e didáticas. Alguns exemplos de versões comerciais, com licença gratuita para fins acadêmicos, são CPLEX da IBM ILOG [IBM, 2010], Gurobi da Gurobi Optimization, Inc. [GUROBI, 2014], Lingo e Lindo da Lindo System, Inc. [LINDO, 1999, 2014]. Alguns gratuitos são GNU Linear Programming Kit GLPK [Makhorin, 2010a] e o lp_solve [LP_SOLVE, 2015]. Algumas versões didáticas que vem acompanhando livros são o TORA [Taha, 2008] e o OR Tutor [Hillier e Lieberman, 2013]. Estas são aplicações locais baseados em uma interface cujas informações dos modelos são preenchidas dentro de campos definidos. Em outras palavras, o usuário não necessita conhecer uma linguagem formatada para o modelo. Por último, tem-se também o PROLIN [DPI-UFV, 2012] e o GNU MathProg [MATH-PROG, 2015], que são sistemas Web que recebem como entrada um modelo em formato de texto e, depois, geram a solução ótima do problema.

Em um primeiro curso que ensina técnicas de modelagem e solução de problemas de otimização, em geral, são disponíveis para os alunos poucas ferramentas gratuitas de cunho didático, com fácil acesso e com uma interface amigável. Em geral, os mesmos devem aprender, em um primeiro momento, a manusear formatos de modelo em texto, como por exemplo em MPS, AMPL, GMPL dentre outros [GUROBI, 2014; IBM, 2010; Hillier e Lieberman, 2013; Fourer et al., 2002; Makhorin, 2010b]. Desta maneira, gasta-se um tempo para se ter familiaridade com a ferramenta, para assim depois, poder obter os resultados do processo de modelagem. O solucionador didático, em um primeiro momento, pode ser mais interessante para o aprendizado do aluno. Pois, de imediato, o mesmo já possui uma interface intuitiva para informar os dados do problema, para então solucioná-lo instantaneamente.

Este trabalho apresenta a primeira versão do *Sistema Interativo para Métodos de Otimização* (SIMO). Ele possui a proposta de suprir a necessidade de uma ferramenta Web para auxiliar no aprendizado da PO, com uma interface amigável e com boa usabilidade. O SIMO é uma ferramenta online gratuita, com acesso através do endereço <https://jmmccota.github.io/Otimizacao> via navegadores modernos de dispositivos computacionais, móveis ou não. Neste texto são apresentadas as funcionalidades que solucionam problemas de Programação Linear (PL), Programação Inteira (PI) e Transporte (PT). Para isso, foram implementados três métodos Simplex (duas fases, grande M e generalizado), um Branch-and-Bound e de três algoritmos de transporte (canto noroeste, menor custo e aproximação de Vogel). Tais algoritmos foram codificados à partir da obra [Taha, 2008].

¹Ou conjunto de soluções ótimas, se for o caso.

O objetivo principal da interface do SIMO é facilitar o entendimento do aluno. Desta maneira, buscou-se implementar interfaces interativas, autoexplicativas e de fácil manuseio. O projeto de software seguiu um modelo em cascata [Pressman, 2009]. Para a implementação, foram utilizadas na construção desta ferramenta as tecnologias HTML5 [Lubbers et al., 2011] para estruturar as páginas, CSS3 [Frain, 2012] para inserir efeitos visuais, e a linguagem JavaScript [Flanagan, 2011] para programar os métodos. Foram utilizadas também as bibliotecas glpk.js [Gourvest, 2015] para solucionar os modelos relaxados no método Branch-and-Bound, MathJax [Cervone et al., 2015] para formatar equações, JQuery [Duckett, 2014] para facilitar a execução dos efeitos de transição e visuais, Bootstrap [Megosinarso, 2014] para criar uma interface responsiva, e vis.js [Almende, 2016] para a construção de árvores para representar a execução do Branch-and-Bound.

A principal contribuição deste trabalho para a comunidade de PO é a oferta gratuita de um solucionador online, com interface amigável, para auxiliar no aprendizado. Para utilizar este sistema não é necessário saber uma linguagem de modelagem, mas deve-se ter apenas os parâmetros do modelo. O uso do SIMO será difundido no Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, nas disciplinas que ensinam modelagem e solução de problemas de otimização.

O trabalho está organizado da seguinte forma: a Seção 2 apresenta um conjunto de trabalhos relacionados; a Seção 3 apresenta detalhes da construção do sistema SIMO; a Seção 4 exemplifica o funcionamento da ferramenta com exemplos de uso; e por fim, a seção 5 apresenta as considerações finais e sugestões de trabalhos futuros.

2. Trabalhos Relacionados

É possível encontrar solucionadores cujo funcionamento possui a mesma inspiração que a do SIMO para resolver problemas de otimização.

O TORA [Taha, 2008] e o OR Tutor [Hillier e Lieberman, 2013] são aplicações desktop que possuem diversas rotinas para solucionar problemas de PO, como por exemplo, PL, PI, caminho mínimo, árvore geradora mínima, fluxo máximo, análise de filas e soma zero. A interface das mesmas são autoexplicativas e somente é necessário fornecer os parâmetros do problema. A principal vantagem destes dois é a facilidade de uso. A desvantagem é o fato de não ser um sistema Web, capaz de ser acessado por um navegador moderno, em um sistema operacional qualquer. Além do mais para se ter acesso ao TORA é necessário comprar o livro. Já o instalador do OR Tutor pode obtido no endereço <http://www.mhhe.com/engcs/industrial/hillier/iortutorial/install/instruction.html> ou com o CD que acompanha o livro.

Dois sistemas Web conhecidos para solucionar problema de otimização são o PROLIN [DPI-UFV, 2012] e o MathProg [MATH-PROG, 2015].

O PROLIN, que pode ser acessado pelo endereço <http://www.prolin.ufv.br/>, foi desenvolvido pela Universidade Federal de Viçosa por uma equipe em 1987. Ele utiliza o simplex revisado como algoritmo base. Uma adaptação da codificação para a Web foi realizada em 2003. No PROLIN somente é possível resolver problemas de PL com até 50 restrições e 100 variáveis. Para que um problema seja solucionado, é necessário editar o modelo em uma caixa de texto, ou carregar um arquivo no servidor, no formato especificado. Depois, clica-se no devido link para se gerar a solução.

O MathProg, que pode ser acessado em <http://www3.nd.edu/~jeff/mathprog/>, é uma aplicação Web para otimização de modelos de PL, PI e PI misto. É utilizado um código de fonte aberto em Javascript para os modelos. O MathProg, assim como PROLIN, aceita o carregamento de problemas modelados e salvos no computador no formato gmpl. Também é possível escrever e editar um modelo direto no navegador.

Ambos PROLIN e MathProg disponibilizam para os usuários um manual, bem como exemplos de utilização. E também, os dois necessitam do conhecimento prévio da sintaxe do modelo em formato de texto.

O SIMO tem como intuito ser um solucionador didático Web para os mais diversos problemas de otimização com interface amigável, sendo somente necessário informar, ou carregar, os

Algoritmos de Pesquisa Operacional

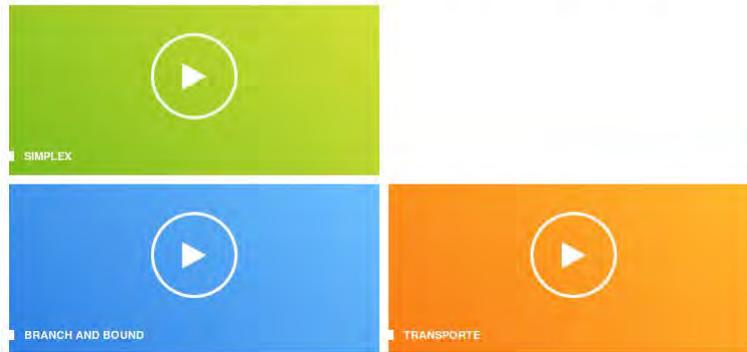


Figura 1: Recorte da interface da página inicial do SIMO.

dados do problema. Em comparação com as ferramentas existentes, o SIMO os supera em relação a facilidade de acesso por ser gratuito e acessível por qualquer navegador moderno em um computador ou dispositivo móvel. Também a interface proposta possui uma preocupação com a oferta de informações relevantes dos métodos, algo que dificilmente é encontrado nos outros trabalhos. Além do mais, é oferecido ao usuário um suporte sobre como manipular o SIMO.

3. O Sistema

O SIMO é uma ferramenta desenvolvida com o intuito de suprir a necessidade de uma ferramenta online que possua uma interface amigável, intuitiva e com uma boa usabilidade para solucionar problemas de otimização. Ele foi pensado para facilitar o entendimento em relação ao comportamento dos métodos de PO. Em particular, na versão atual há implementações de algoritmos que solucionam problemas de PL, PI e PT. Tais módulos podem ser acessados pela interface da página inicial, <https://jmmccota.github.io/Otimizacao>, que possui um menu conforme apresentado pela Figura 1.

O projeto SIMO teve origem em um trabalho da disciplina de Otimização 1, do curso de Engenharia da Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo. À partir daí, teve-se a ideia de se criar uma ferramenta online gratuita com algoritmos para solucionar problemas de otimização que fosse de fácil manuseio e de característica didática. Assim levantou-se os requisitos do sistema, no que diz respeito à todos os algoritmos a serem implementados e como seriam apresentados ao usuário final, e elaborado um modelo em cascata para a produção do mesmo. Pretende-se oferecer as diversas implementações do Simplex; método gráfico; algoritmos de pontos interiores para PL; ferramentas auxiliares para realizar análise de sensibilidade, conversão dual-primal, análise de pós-otimização; algoritmos para problemas de transporte; métodos para otimização em modelos de grafos; algoritmos para programação inteira e programação dinâmica. Para este trabalho, são apresentados três implementações do Simplex, três algoritmos para transporte e um algoritmo Branch-and-Bound para programação inteira.

Foram utilizadas na construção do SIMO as tecnologias HTML5 para estruturar as páginas, que possui novas funcionalidades como semântica, multimídia, interação e portabilidade; CSS3, que ficará responsável pela definição de estilos visuais; e a linguagem JavaScript para implementar os algoritmos e tratar a interação com o usuário final. Assim, todo o processamento de cálculos será realizado na máquina cliente, ao invés de ser no servidor. Algumas bibliotecas auxiliares foram utilizadas, que são glpk.js, que possui estrutura de dados e rotinas para solucionar problemas de PL e PI; MathJax, que ficará responsável por formatar equações e outras saídas nu-

méricas e simbólica no estilo Latex; JQuery para aplicar efeitos dinâmicos de transição e visuais; Bootstrap para a criação de uma interface responsiva; e vis.js para a construção de árvores que representarão a execução do Branch-and-Bound.

A ferramenta SIMO, até o momento da escrita deste artigo pode ser acessada de qualquer computador ou dispositivo móvel com um navegador moderno. Ele se divide em três seções. A primeira, Simplex, possui as implementações das versões duas fases, grande M e generalizado. A segunda, Branch-and-Bound, possui a implementação do algoritmo para solucionar problemas de PI. E a terceira, Transporte, possui as implementações dos métodos do canto noroeste, menor custo e aproximação de Vogel. Cada uma delas contém um formulário para entrada do modelo matemático e parametrização do método, opções para salvar e carregar, e páginas de ajuda informando sobre as técnicas e sobre como utilizar a ferramenta.

Os modelos criados podem ser utilizados para visualizar os resultados da execução completa do algoritmo ou por iteração individual. Em ambos os casos, são exibidas as etapas da execução dos algoritmos, destacando as informações mais importantes que determinaram as decisões tomadas pelo método.

É possível também fazer com que o algoritmo siga um caminho diferente do usual, caso o usuário deseje avaliar o que aconteceria de forma alternativa. Essa alteração no caminho se dá de forma particular a cada método. Por exemplo, no Simplex, pode-se escolher a variável não básica que entrará na base, dentre as possíveis. E a partir desta, é definido qual variável sai. No Branch-and-Bound, escolhe-se em qual variável, com resultado final não inteiro, será feito o processo de ramificação. Ou seja, pode-se definir a adição de restrições ao valor daquela variável selecionada que cria dois modelos relaxados distintos. Nos algoritmos de transporte, o usuário pode escolher uma rota para obter recursos alocados.

4. Exemplos de Uso

Nesta seção são ilustrados alguns exemplos do funcionamento do SIMO atual, e como os resultados são exibidos para o usuário. Devido ao número limite de páginas, nem todos os algoritmos implementados terão exemplos de uso apresentados. Entretanto, fica o convite para o leitor utilizar o sistema, enviar sugestões e críticas aos autores deste trabalho.

4.1. Um Problema de Programação Linear

Inicialmente será solucionado o problema de PL apresentado pela Equação 1 pelo Simplex duas fases. Primeiramente, deve-se acessar a interface Simplex pela página inicial ou pelo endereço <https://jmmccota.github.io/Otimizacao/simplex.html>. No menu de opções “Criar Novo Problema”, deve-se informar corretamente o algoritmo Simplex desejado, o sentido da otimização, o número de variáveis e pressionar o botão “Novo Problema de Otimização”. A Figura 2 apresenta um recorte da interface com os dados do problema preenchidos.

Ao pressionar “Executar Completo”, obtém-se o conjunto de iterações representado pelas tabelas, com a indicação das variáveis que entram e saem da base. O usuário também possui a opção de executar somente uma iteração por vez. Além disso, ele pode selecionar qual variável pode entrar na base, caso ela contribua com a melhoria da função objetivo na próxima iteração.

A Figura 3, extraída do SIMO, apresenta as tabelas das duas últimas iterações do Simplex duas fases, quando se executa por completo. Nela encontra-se a solução ótima $x_1 = 0,4$ e $x_2 = 1,8$ com $z(x_1, x_2) = 3,4$, conforme apresentado pela tabela ótima.

$$\begin{aligned}
 \max z &= 4x_1 + x_2 \\
 \text{sujeito a: } &3x_1 + x_2 = 3 \\
 &4x_1 + 3x_2 \geq 6 \\
 &x_1 + 2x_2 \leq 4 \\
 &x_1, x_2 \geq 0
 \end{aligned} \tag{1}$$

Problema de Otimização

Escreva seu problema de programação linear abaixo ou carregue-o.

Criar Novo Problema | Carregar Problema

Tipo do Simplex: **Dois Fases**

Tipo do Problema: **Minimização**

Número de Variáveis: **2**

Novo Problema de Otimização

	x1	x2	Relação	Lado Direito
Objetivo	4	1		
Restrição1	3	1	=	3
Restrição2	4	3	>=	6
Restrição3	1	2	<=	4

+ Adicionar Restrição | **- Excluir Restrição**

	x1	x2
Limite Superior	inf	inf
Limite Inferior	0	0

Salvar Problema | **Executar Completo** | **Passo a Passo**

Limpar Tudo

Figura 2: Recorte da interface do módulo Simplex do SIMO.

Iteração 7						
Base	x1	x2	x3	x4	Resultado	Razão
z	0	0	0,2	0	3,6	
x1	1	0	0,2	0	0,6	3
x2	0	1	-0,6	0	1,2	-2
x4	0	0	1	1	1	1

Iteração 8						
Base	x1	x2	x3	x4	Resultado	Razão
z	0	0	0	-0,2	3,4	
x1	1	0	0	-0,2	0,4	
x2	0	1	0	0,6	1,8	
x3	0	0	1	1	1	

Figura 3: Tabela ótima encontrada pelo Simplex duas fases para o exemplo apresentado pela Equação 1.

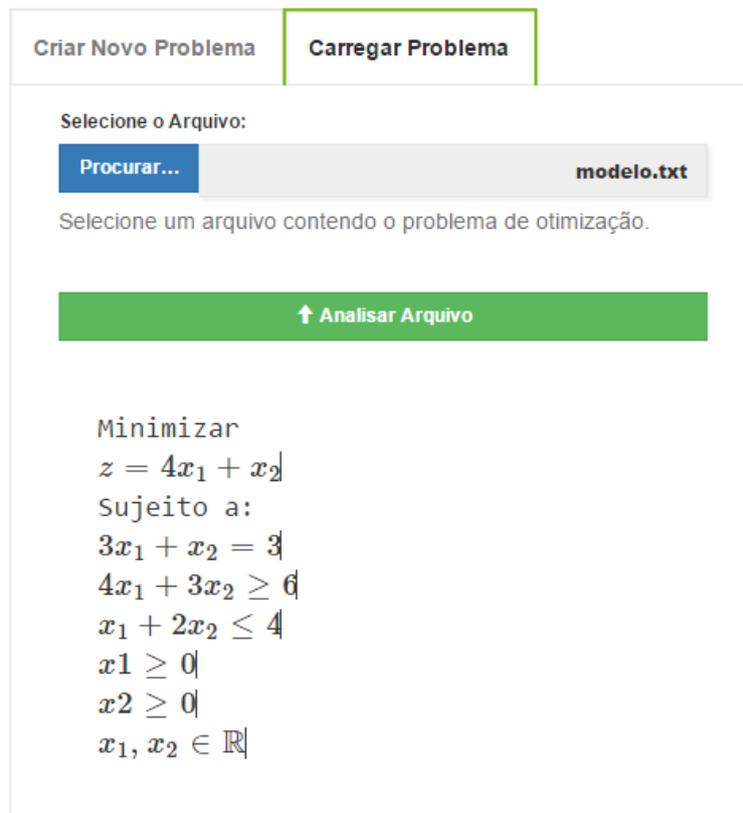


Figura 4: Interface SIMO usada para carregar o modelo, apresentado pela Equação 1, salvo.

Se porventura o usuário desejar salvar os dados preenchidos do modelo, pode-se clicar em “Salvar Problema”. Para carregar um conjunto de dados, deve-se clicar na aba “Carregar Problema”, apresentado na parte superior da interface. Deve-se então selecionar o arquivo correto clicando-se em “Procurar...”, que será automaticamente carregado. Se porventura os dados estiverem coerentes, então a representação em formato de equação será impressa para o usuário. Ao pressionar o botão “Analisar Arquivo” carrega-se o formulário do modelo preenchido com os parâmetros. A Figura 4 apresenta a representação do modelo carregado via arquivo para o exemplo da Equação 1.

4.2. Um Problema de Programação Inteira

O segundo exemplo, ilustrado pela Equação 2, apresenta um modelo de programação inteira cujo ótimo será obtido pelo método Branch-and-Bound.

Para solucioná-lo, deve-se acessar a interface Branch-and-Bound pela página inicial ou pelo endereço <https://jmmccota.github.io/Otimizacao/branchbound.html>. Seguindo um conjunto de passos parecidos com o exemplo anterior, no menu de opções “Criar Novo Problema”, deve-se preencher corretamente o sentido da otimização, o número de variáveis e pressionar o botão “Novo Problema de Otimização”.

Novamente, o usuário possui a opção de executar por completo ou um passo por vez. No caso de pressionar “Executar Completo”, obtém-se a árvore ramificada final do Branch-and-Bound apresentado pela Figura 5. Nesta árvore se encontra em cada nó as informações de todas as soluções ramificadas dos modelos relaxados. A solução ótima, que está no vértice 3 selecionado, possui os valores $x_1 = 0$, $x_2 = 2$, com $z(x_1, x_2) = 10$.

O conjunto de iterações é representado por cada vértice da árvore de ramificação. Para cada nó, se observa o modelo relaxado que o representa, a classificação do tipo de solução, o valor ótimo do modelo relaxado (caso exista) com os respectivos valores ótimos.

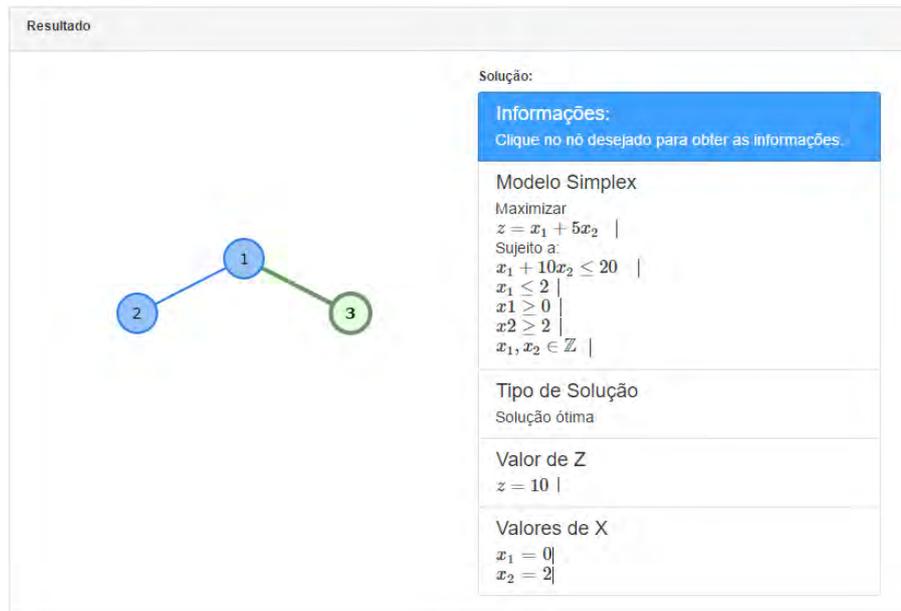


Figura 5: Árvore final do Branch-and-Bound para o exemplo apresentado pela Equação 2, com destaque para a solução ótima localizada no vértice 3.

		Destino				Fornecimento
		1	2	3	4	
Origem	1	10 x_{11}	2 x_{12}	20 x_{13}	11 x_{14}	15
	2	12 x_{21}	7 x_{22}	9 x_{23}	20 x_{24}	15
	3	4 x_{31}	14 x_{32}	16 x_{33}	18 x_{34}	15
Demanda		5	15	15	15	

Tabela 1: Dados do exemplo de um problema de transporte. Cada célula informa o custo de se transportar uma unidade da origem $i \in \{1, 2, 3\}$ para o destino $j \in \{1, 2, 3, 4\}$, na parte de cima, e a variável que representa a quantidade transportada x_{ij} , na parte de baixo. A última coluna representa a quantidade fornecida por cada origem e a última linha as unidades demandadas por cada destino.

Para o caso em que o usuário desejar executar passo a passo, ele pode selecionar a variável na qual deseja ramificar a árvore de execução.

$$\begin{aligned}
 \max z &= x_1 + 5x_2 \\
 \text{sujeito a: } &x_1 + 10x_2 \leq 20 \\
 &x_1 \leq 2 \\
 &x_1, x_2 \in \mathbb{Z}^+
 \end{aligned} \tag{2}$$

4.3. Um Problema de Transporte

O terceiro exemplo, cujos dados são apresentados pela Tabela 1, ilustra um problema de transporte que será solucionado pelo método de menor custo.

Para solucioná-lo, deve-se acessar a interface Transporte pela página inicial ou pelo endereço <https://jmmccota.github.io/Otimizacao/transporte.html>. Informando o algoritmo desejado e o número de destinos, deve-se depois pressionar “Novo Problema” para

	Coluna1	Coluna2	Coluna3	Coluna4	Suprimento
Linha1	10	2	20	11	15
Linha2	12	7	9	20	25
Linha3	4	14	16	18	10
Demanda	5	15	15	15	

+ Adicionar Coluna + Adicionar Linha - Excluir Coluna - Excluir Linha

Salvar Problema Executar Completo Passo a Passo Limpar Tudo

Figura 6: Parâmetros do exemplo apresentado pela Tabela 1 na ferramenta SIMO.

Iteração 5					
	Coluna 1	Coluna 2	Coluna 3	Coluna 4	Suprimento
Linha 1	Custo: 10 x Quantidade: 0	Custo: 2 x Quantidade: 15	Custo: 20 x Quantidade: 0	Custo: 11 x Quantidade: 0	0
Linha 2	Custo: 12 x Quantidade: 0	Custo: 7 x Quantidade: 0	Custo: 9 x Quantidade: 15	Custo: 20 x Quantidade: 10	0
Linha 3	Custo: 4 x Quantidade: 5	Custo: 14 x Quantidade: 0	Custo: 16 x Quantidade: 0	Custo: 18 x Quantidade: 5	0
Demanda	0	0	0	0	

Figura 7: Solução final do exemplo apresentado pela Tabela 1 gerado pelo método de menor custo no SIMO.

carregar a tabela na qual os parâmetros referentes ao custo unitário de transporte serão informados. Pode-se adicionar/remover linhas e colunas para determinar o número correto de variáveis/parâmetros. Na última coluna deve-se informar a quantidade fornecida pelas origens, e na última linha as quantidades demandadas pelos destinos. A Figura 6 apresenta como os dados são informados no SIMO.

Após clicar em “Executar Completo” o método de menor custo encontrou a solução $x_{12} = 15, x_{23} = 15, x_{24} = 10, x_{31} = 5, x_{34} = 5$ com o restante das variáveis assumindo o valor 0. Tal resultado é apresentado pela Figura 7 extraída do SIMO. A solução transportou todo suprimento e conseguiu atender toda a demanda.

5. Conclusões

Este trabalho apresentou a primeira versão do *Sistema Interativo para Métodos de Otimização*, ou simplesmente, SIMO. Ele é um sistema online com um conjunto de ferramentas criadas com o objetivo de auxiliar o ensino de disciplinas relacionadas com a Pesquisa Operacional. O SIMO possui interface gráfica responsiva e de fácil entendimento que pode ser acessada gratuitamente no endereço <https://jmmccota.github.io/Otimizacao> através de um computador ou dispositivo móvel.

O SIMO apresenta diretamente ou passo a passo todas as iterações dos algoritmos, ressaltando as informações relevantes. Nesta versão, foram implementados os algoritmos duas fases, grande M, e generalizado do Simplex, um algoritmo Branch-and-Bound para otimização linear inteira, e os algoritmos que solucionam o problema de transporte canto noroeste, menor custo e aproximação de Vogel.

Foi apresentado também alguns detalhes do funcionamento da ferramenta e ilustrado três exemplos de aplicação do SIMO. O sistema se mostra como uma opção interessante de material didático para estudantes e professores de graduação. O SIMO não requer conhecimento de uma linguagem de modelagem como ocorre em outras ferramentas. Além do mais, ele é eficiente e de fácil manuseio para entender a mecânica dos procedimentos computacionais que solucionam problemas de otimização.

A praticidade de uma ferramenta que oferece uma interface intuitiva e simples para o melhor entendimento dos métodos, juntamente com a exibição passo a passo dos cálculos realizados é de grande valia para estudantes. Pode-se notar pelos resultados obtidos, que um usuário com pouco conhecimento consegue entender e executar os métodos com facilidade.

Em trabalhos futuros serão implementados o método gráfico para modelos lineares com duas variáveis, algoritmos de pontos interiores, ferramentas auxiliares para realizar análise de sensibilidade, conversão dual-primal, análise de pós-otimização, métodos para otimização em modelos de grafos, e técnicas de programação dinâmica.

Referências

- Almende, B. V. (2016). vis.js: A dynamic, browser based visualization library. URL <http://visjs.org/>. Acesso em 1 de janeiro de 2016.
- Cervone, D., Sorge, V., Perfect, C., e Krautzberger, P. (2015). The mathjax consortium. URL <https://www.mathjax.org>. Acesso em 1 de dezembro de 2015.
- DPI-UFV (2012). -prolin- sistema para programação linear. URL http://www.prolin.ufv.br/arquivos/prolin_tradicional.pdf. Acesso em 1 de março de 2015.
- Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. John Wiley & Sons, Inc, 1 edition.
- Flanagan, D. (2011). *JavaScript: The definitive guide: Activate your web pages*. O'Reilly Media, Inc.
- Fourer, R., Gay, D. M., e Kernighan, B. (2002). *AMPL: A modeling language for mathematical programming*. Cengage Learning, 2 edition.
- Frain, B. (2012). *Responsive web design with HTML5 and CSS3*. Packt Publishing Ltd.
- Gourvest, H. (2015). Gnu linear programming kit for javascript. URL <https://github.com/hgourvest/glpk.js>. Acesso em 1 de março de 2015.
- GUROBI (2014). *Gurobi Optimizer Reference Manual*. Gurobi Optimization, Inc., 1 edition. URL <http://www.gurobi.com/documentation/6.0/refman.pdf>.
- Hillier, F. S. e Lieberman, G. J. (2013). *Introdução a Pesquisa Operacional*. McGraw-Hill, 9 edition.
- IBM (2010). *IBM ILOG AMPL Version 12.2 User's Guide*. International Business Machines Corporation, 1 edition. URL <http://ampl.com/BOOKLETS/amplcplex122userguide.pdf>.
- LINDO (1999). *Lingo the Modeling Language & Optimizer*. LINDO SYSTEMS INC, 1 edition. URL <http://www.lindo.com/downloads/PDF/LINGO14.PDF>.
- LINDO (2014). *LINDO API 9.0 User Manual*. LINDO SYSTEMS INC. URL <http://www.lindo.com/downloads/PDF/APIMAN9.pdf>.
- LP_SOLVE (2015). Introduction to lp_solve 5.5.2.0. URL <http://web.mit.edu/lpsolve/doc/>. Acesso em 1 de março de 2015.
- Lubbers, P., Albers, B., Salim, F., e Pye, T. (2011). *Pro HTML5 programming*. Springer.
- Makhorin, A. (2010a). Gnu linear programming kit reference manual for glpk version 4.45. URL <https://www.gnu.org/software/glpk/>.

- Makhorin, A. (2010b). Modeling language gnu mathprog language reference for glpk version 4.45. URL <https://www3.nd.edu/~jeff/mathprog/glpk-4.47/doc/gmpl.pdf>.
- MATH-PROG (2015). Mathematical programming in gnu mathprog. URL <http://www3.nd.edu/~jeff/mathprog/>. Acesso em 1 de março de 2015.
- Megosinarso, R. (2014). *Step By Step Bootstrap 3: A Quick Guide to Responsive Web Development Using Bootstrap 3*. CreateSpace Independent Publishing Platform, 1 edition.
- Pressman, R. S. (2009). *Engenharia de software*. Bookman, 7 edition.
- SOBRAPO (2014). Pesquisa operacional. URL http://www.sobrapo.org.br/o_que_e_po.php. Acesso em 1 de março de 2014.
- Taha, H. A. (2008). *Pesquisa Operacional*. Pearson Prentice Hall, 8 edition.