



DESENVOLVIMENTO DE APLICATIVO EDUCACIONAL AUXILIAR EM PYTHON PARA O ENSINO E APRENDIZAGEM DA OTIMIZAÇÃO COMBINATÓRIA NAS ENGENHARIAS

DOI: 10.37702/2175-957X.COBENGE.2024.5111

Autores: THIAGO ROBERTO DOS SANTOS

Resumo: O presente artigo teve como objetivo desenvolver um aplicativo educacional em Python destinado ao ensino de Otimização Combinatória, matéria comumente abordada em disciplinas de Pesquisa Operacional na Engenharia. O aplicativo, construído utilizando Python em conjunto com as bibliotecas PuLP e Flet, visa encontrar os valores ótimos das variáveis que maximizam ou minimizam funções lineares presentes em exercícios de Programação Linear Inteira, um componente essencial da Otimização Combinatória. A interface do aplicativo foi projetada para ser intuitiva e de fácil utilização pelo usuário. Os resultados obtidos através dos exercícios propostos demonstram que o aplicativo representa uma ferramenta promissora para a resolução de problemas de Programação Linear Inteira, oferecendo uma interface acessível que pode facilitar o ensino de Otimização Combinatória para estudantes de Engenharia. Destaca-se, portanto, sua relevância no contexto do desenvolvimento de profissionais na área de Pesquisa Operacional.

Palavras-chave: Otimização combinatória, Ensino de engenharia, Python, Aplicativo educacional.

DESENVOLVIMENTO DE APLICATIVO EDUCACIONAL AUXILIAR EM PYTHON PARA O ENSINO E APRENDIZAGEM DA OTIMIZAÇÃO COMBINATÓRIA NAS ENGENHARIAS

1 INTRODUÇÃO

No cenário atual, é crucial que os estudantes de Engenharia tenham domínio dos recursos computacionais e dos conceitos envolvidos na modelagem de problemas de otimização em suas grades curriculares, pois isso possibilita sua aplicação em projetos reais. Como destacado por Silva e Brasil (2019), durante todo o processo de concepção do projeto, busca-se otimizar o desempenho em diversas áreas, como análise, projeto, fabricação, vendas, pesquisa e desenvolvimento.

Este trabalho apresenta o desenvolvimento de um aplicativo em Python que otimiza funções lineares e utiliza a biblioteca Flet para a criação da interface gráfica para o usuário. O objetivo do aplicativo é encontrar os valores das variáveis que maximizam ou minimizam uma função linear, sujeita a um conjunto de restrições lineares. Ele oferece uma interface de fácil operação para o usuário iniciante, o que permite a ele inserir a função objetivo, as restrições e a quantidade de variáveis, além de escolher o tipo de otimização desejada (maximização ou minimização).

Para validar o aplicativo desenvolvido, foram realizados testes em problemas de Programação Linear Inteira (PLI) para maximização e minimização de funções objetivo. Além disso, foi descrito o funcionamento do aplicativo e exemplos de uso.

1.1 Ensino da Otimização Combinatória para a Engenharia

Dentro do contexto da Otimização Combinatória, Miyazawa e Souza (2015) destacam sua aplicabilidade em uma ampla gama de problemas, que visam encontrar soluções que otimizem o uso dos recursos disponíveis. Esses problemas abrangem diversas áreas, como projetos de redes, logística e produção.

Entretanto, os estudantes frequentemente enfrentam desafios ao compreender conceitos complexos presentes nos problemas de Programação Linear (PL) o que os afasta do conteúdo de PL, negligenciando uma área com grande aplicabilidade na engenharia.

1.2 Otimização Combinatória

“Em termos gerais, otimização é o problema de minimizar ou maximizar uma função sujeita a uma série de restrições.” (Guenin, könemann e Tunçel, 2014, tradução nossa)

Na Otimização Combinatória, há a minimização ou maximização de uma função aplicada a um conjunto finito de elementos, demandando o uso de técnicas para encontrar soluções ótimas. Em um problema de otimização, destacam-se uma função objetivo e um conjunto de restrições, ambos associados às variáveis de decisão.

Esses conceitos foram fundamentais para o desenvolvimento do aplicativo, principalmente para as entradas de dados necessárias para a resolução dos problemas de Programação Linear Inteira.

1.3 Problemas de Programação Linear Inteira (PLI)

Problemas de Programação Linear Inteira (PLI) são problemas de Otimização Combinatória nos quais a função objetivo e as restrições são todas lineares e restritas a valores inteiros.

Exemplos incluem o design de redes de comunicação para minimizar custos, a programação de produção em fábricas considerando recursos e demanda, o corte eficiente de estoque para minimizar o desperdício, o roteamento otimizado de veículos para reduzir custos e a programação de horários de trabalho ou aulas para maximizar a eficiência e atender às preferências.

No contexto deste estudo, o aplicativo foi especificamente desenvolvido para resolver problemas de Programação Linear Inteira (PLI), embora possa ser adaptado para abordar uma variedade de outros problemas de Programação Linear.

2 REFERENCIAL TEÓRICO

Para abordar os aspectos propostos por este estudo, é necessário inicialmente conduzir uma pesquisa sobre conceitos fundamentais da Programação Linear.

2.1 Uso de tecnologias digitais no ensino

O uso de tecnologias no ensino é uma prática cada vez mais comum e necessária atualmente. Com o avanço constante da tecnologia, surgiram diversas ferramentas e recursos que têm o potencial de transformar a forma como aprendemos e ensinamos. Bacich e Moran (2018, p.9) destacam que "A metodologia ativa se caracteriza pela inter-relação entre educação, cultura, sociedade, política e escola, sendo desenvolvida por meio de métodos ativos e criativos, centrados na atividade do aluno com a intenção de propiciar a aprendizagem".

No entanto, é crucial não apenas adotar tecnologias educacionais, mas também explorar como essas ferramentas podem ser aplicadas de forma eficaz para melhorar o processo de ensino-aprendizagem. O aplicativo desenvolvido neste trabalho proporciona não apenas uma abordagem ativa e criativa, mas também facilita a compreensão de conceitos complexos de maneira dinâmica e acessível para os estudantes iniciantes no assunto. Ao simplificar a entrada de dados e o processo de resolução de problemas, espera-se promover um ambiente educacional mais interativo e engajador.

Portanto, a integração de tecnologias digitais no ensino, como o aplicativo desenvolvido, é fundamental para facilitar a compreensão de conceitos complexos e aprimorar o processo de aprendizagem, tornando-o mais dinâmico e acessível para os alunos.

2.2 Programação Linear (PL)

Um engenheiro que atua como gestor em um processo de produção de qualquer empresa é frequentemente requisitado a tomar decisões baseadas na escassez de recursos, o que representa limitações significativas na situação.

De acordo com Corrar e Garcia (2001), "O modelo de Programação Linear é utilizado como auxílio para a resolução de problemas que envolvem alocação de recursos escassos, visando alcançar um certo objetivo." Em outras palavras, a Programação Linear auxilia o engenheiro, ou outro profissional, na tomada de decisões visando à ótima utilização dos recursos disponíveis.

Esse problema pode ser representado por uma função linear, chamada função objetivo, que deve ser maximizada ou minimizada ao alterar os valores das variáveis de

modelagem matemática permitiu a representação formal dos problemas de otimização abordados, enquanto a implementação computacional viabilizou a solução prática desses problemas.

3.2 Linguagem Python

“Python possui toda a potência e flexibilidade que você esperaria de uma linguagem de programação moderna e orientada a objetos. Mas mesmo com todo o seu poder, você pode se surpreender com a rapidez com que pode construir programas” (Dawson, 2010, p. 5, tradução nossa). A citação de Downey destaca a potência e a flexibilidade da linguagem Python, enfatizando como é rápido construir programas com Python. Essa observação reforça a reputação de Python como uma linguagem acessível e eficiente, especialmente para iniciantes na programação.

Diante disso, é possível destacar a flexibilidade da linguagem Python, juntamente com as diversas bibliotecas disponíveis para o desenvolvimento de aplicativos por iniciantes na programação. Neste projeto, foram utilizadas duas bibliotecas da linguagem Python, a biblioteca PuLP, a biblioteca Flet e o código foi escrito na IDE Microsoft Visual Studio Code.

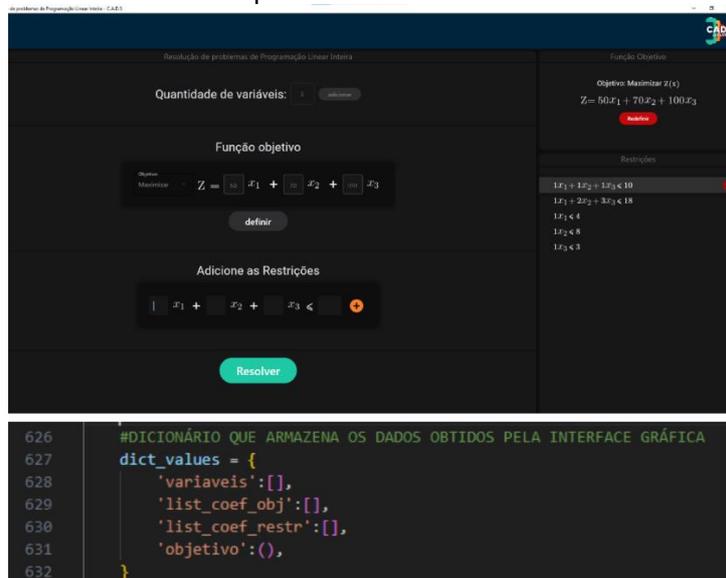
Biblioteca Flet

Na metodologia deste estudo, empregamos o framework Flet como uma ferramenta fundamental para o desenvolvimento de aplicativos em Flutter utilizando a linguagem Python, proporcionando uma abordagem inovadora que permite a criação de aplicativos práticos e de interface objetiva, mesmo sem conhecimento prévio da linguagem Dart. Destacamos que o Flet possibilita o rápido desenvolvimento de aplicativos, aproveitando sua funcionalidade de coleta e apresentação de dados intuitivamente.

Utilizamos o Flet durante todo o processo de desenvolvimento do aplicativo, desde a obtenção dos dados do modelo matemático até a apresentação dos resultados gerados pelo PuLP, garantindo uma experiência eficiente para o usuário final.

A Figura 1 mostra a interface de inserção de dados apresentada ao usuário desenvolvida com o Flet, bem como o dicionário *dict* que armazena esses dados do modelo de Programação Linear Inteira por meio de listas e matrizes em Python.

Figura 1 – Interface gráfica apresentada ao usuário e dicionário *dict* que armazena os dados inseridos.



Fonte: Autoria própria

A interface Flet apresenta aos usuários a entrada da quantidade de variáveis, que corresponde ao número de variáveis de decisão a serem otimizadas, bem como os campos para inserção dos coeficientes da função objetivo e das restrições. Optar por inserir os coeficientes em vez da função objetivo completa direciona o foco do estudante para o processo de modelagem, em detrimento da formatação de equações e funções. Isso simplifica o aprendizado, especialmente no escopo deste estudo. Com o uso da ferramenta Flet, conseguimos uma interface limpa e direta, ideal para o ensino e aprendizado de modelos de PLI.

No dicionário *dict*, temos quatro chaves: *variaveis*, *list_coef_obj*, *list_coef_restr* e *objetivo*. Cada uma dessas chaves de dicionário recebe, respectivamente, a quantidade de variáveis que serão estudadas, os coeficientes da função objetivo, os coeficientes da restrição (cada vez que for adicionada uma restrição) e uma *string* de Python “*Maximizar*” ou “*Minimizar*”.

Ao utilizar o Flet, conseguimos criar uma interface gráfica, facilitando a interação dos usuários na modelagem e resolução de problemas de Otimização Combinatória, como demonstrado no contexto do Problema de Programação Linear Inteira (PLI).

Este estudo destaca o potencial da biblioteca Flet como uma ferramenta para o desenvolvimento de aplicativos para Otimização Combinatória, ampliando as possibilidades de aplicação dessa tecnologia nas engenharias.

Biblioteca PuLP

PuLP é um modelador de Programação Linear escrito em Python. Ele é utilizado para formular e resolver problemas de otimização linear e inteira. A Figura 2 evidencia parte da função do Python chamada *solve_problema*, destinada a definir o modelo matemático do PPL através do PuLP.

Figura 2 - Função *solve_problema* de formulação e resolução de problemas lineares através do PuLP.

```

34 #BACKEND-PULP-RESOLUÇÃO-DE-PROBLEMAS-LINEARES-----#
35 def solve_problema(dic):
36     # Lista de coeficientes
37     coeficientes_obj = dic["list_coef_obj"]
38
39     # Lista de coeficientes para restrições
40     coeficientes_restr = dic["list_coef_restr"]
41
42     # Criar um problema de programação linear
43     if dic["objetivo"]=="Minimizar":
44         problema = LpProblem("MeuProblema", LpMinimize)
45     else:
46         problema = LpProblem("MeuProblema", LpMaximize)
47
48     # Criar variáveis dinamicamente
49     variaveis = [LpVariable(name=f"x{i+1}", lowBound=0, cat="Integer") for i in range(len(coeficientes_obj))]
50
51     # Construir a equação do objetivo
52     equacao_objetivo = lpSum(coef * var for coef, var in zip(coeficientes_obj, variaveis))
53
54     # Adicionar a equação ao problema como a função objetivo
55     problema += equacao_objetivo, "Funcao_Objetivo"
56
57     # Adicionar restrições ao problema
58     for i in range(len(coeficientes_restr)):
59         equacao_restricao = lpSum(coef * var for coef, var in zip(coeficientes_restr[i]:len(coeficientes_obj)], variaveis)
60         if dic["objetivo"]=="Minimizar":
61             problema += equacao_restricao >= coeficientes_restr[i][len(coeficientes_obj)], f"Restricaoo{i+1}"
62         else:
63             problema += equacao_restricao <= coeficientes_restr[i][len(coeficientes_obj)], f"Restricaoo{i+1}"
64
65     #Resolvendo o modelo
66     status = problema.solve()
67

```

Fonte: Autoria própria

A função *solve_problema* foi implementada para resolver problemas de PLI utilizando a biblioteca PuLP em Python. Primeiramente, a função extrai os coeficientes da função objetivo e das restrições a partir do dicionário *dict*. Os coeficientes da função

objetivo são armazenados na lista *coeficientes_obj*, enquanto os coeficientes das restrições são armazenados na lista *coeficientes_rest*.

Em seguida, a função cria um problema de Programação Linear Inteira utilizando a classe *LpProblem* da PuLP. Dependendo da especificação do objetivo no dicionário *dict*, o problema é configurado para minimizar ou maximizar a função objetivo.

As variáveis de decisão do problema são criadas dinamicamente por meio de uma lista de compreensão, em que cada variável é nomeada sequencialmente (x_1, x_2, x_3, \dots) e restrita a valores inteiros não negativos. Portanto, não são necessárias restrições adicionais de não-negatividade, uma vez que o problema já é intrinsecamente restrito a valores não negativos, conforme definido no modelo PuLP.

A equação da função objetivo é então construída utilizando a função *lpSum* da PuLP, que permite somar os produtos dos coeficientes e variáveis correspondentes. Essa equação é adicionada ao problema como a função objetivo a ser minimizada ou maximizada.

As restrições do problema são adicionadas ao mesmo utilizando um loop *for*. Para cada restrição, uma equação de restrição é construída e adicionada ao problema, juntamente com uma condição de restrição adequada dependendo do objetivo especificado (\geq para minimização e \leq para maximização).

Finalmente, o modelo de otimização é resolvido utilizando o método *solve* do problema. Portanto, a biblioteca PuLP desempenhou um papel fundamental no projeto, fornecendo as ferramentas necessárias para formular e resolver problemas de PLI de forma eficiente e precisa.

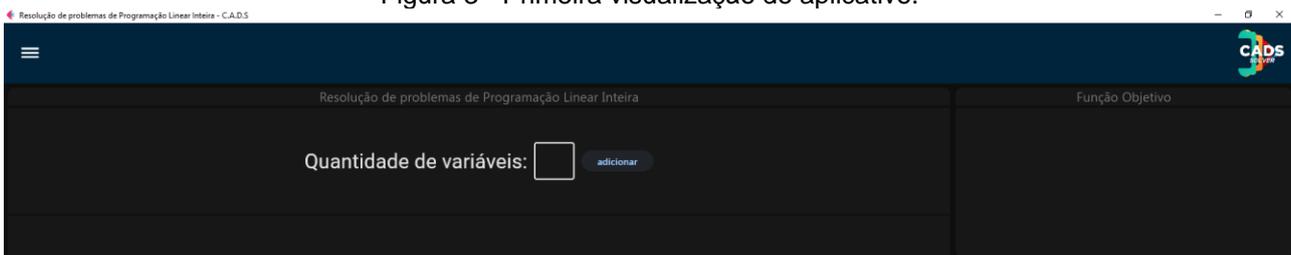
IDE Visual Studio Code

O IDE (Ambiente de Desenvolvimento Integrado) utilizado para escrever os códigos foi o Visual Studio Code, devido à sua interface intuitiva e eficiente, que facilita o processo de escrita e desenvolvimento do aplicativo. Como afirmado por Speight (2021), o Visual Studio Code é reconhecido por ser um editor de código simplificado, ideal para um ciclo rápido de criação de código e depuração.

3.3 Descrição e funcionamento do aplicativo

O aplicativo possui uma única interface gráfica que, inicialmente, apresenta apenas uma entrada: a quantidade de variáveis. Conforme o usuário insere a quantidade de variáveis a serem modeladas no problema de Programação Linear Inteira, as entradas para a função objetivo surgem automaticamente, conforme a primeira entrada. Espera-se que isso auxilie os usuários em seu primeiro contato com o aplicativo, evitando que se percam no processo. A Figura 3 mostra a primeira visualização da interface gráfica do aplicativo.

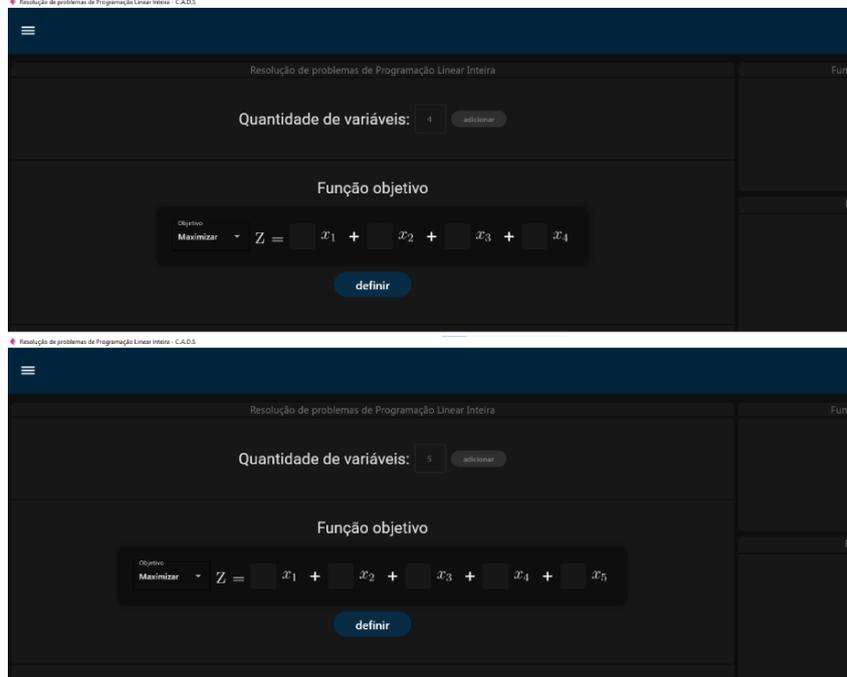
Figura 3 - Primeira visualização do aplicativo.



Fonte: Autoria própria

A Figura 4 mostra a interface do aplicativo ao inserir a quantidade de variáveis e pressionar o botão “Adicionar”, observa-se que o campo de inserção dos coeficientes da função objetivo pode variar segundo a quantidade de variáveis definidas.

Figura 4 – Inserção de 4 e 5 variáveis na interface gráfica.

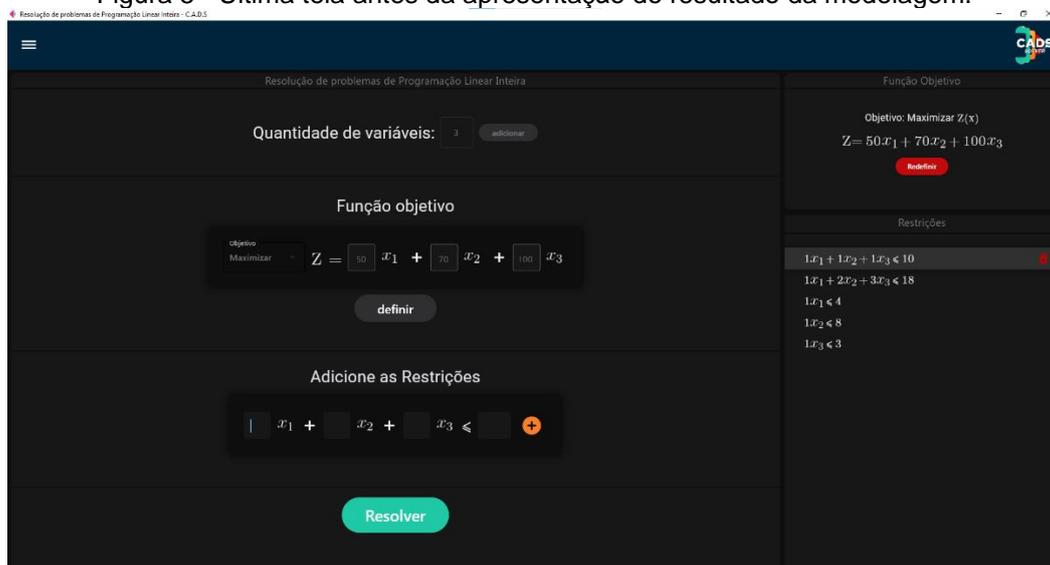


Fonte: Autoria própria

Ao inserir os coeficientes da função objetivo e pressionar o botão definir, a função será formatada e adicionada no painel superior direito para que o usuário durante o uso do aplicativo confirme a função objetivo estabelecida por ele. Além disso, surge a terceira e última entrada de dados, intitulada “Adicione as Restrições”.

A Figura 5 apresenta a visualização final da modelagem, destacando as restrições adicionadas pelo usuário no canto direito, com a opção de exclusão de qualquer restrição recém-adicionada.

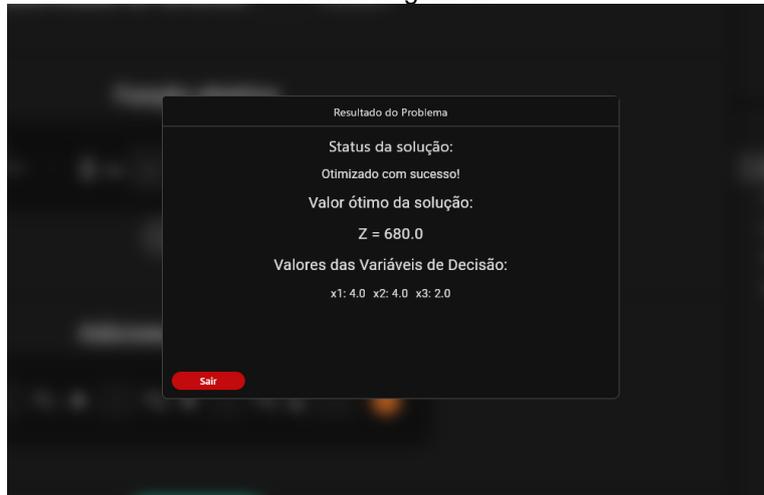
Figura 5 - Última tela antes da apresentação do resultado da modelagem.



Fonte: Autoria própria

Por fim, ao clicar em “Resolver”, obtemos a resposta final da modelagem elaborada através do aplicativo, incluindo informações como o status da solução, o valor ótimo da solução e os valores ótimos das variáveis de decisão como mostrado na Figura 6.

Figura 6 - Apresentação do resultado da modelagem na interface gráfica.



Fonte: Autoria própria

4 RESULTADOS

Na seção de resultados, apresentamos as soluções obtidas pelo aplicativo para quatro problemas específicos de Programação Linear Inteira (PLI).

4.1 Teste do aplicativo com Problemas de Programação Inteira (PI)

Cada problema foi formulado com diferentes quantidades de variáveis de decisão, funções objetivo e restrições, representando uma variedade de desafios típicos encontrados na otimização combinatória.

Esta lista de exercícios foi adaptada de materiais de ensino de Pesquisa Operacional utilizados em diferentes cursos e fontes acadêmicas. A adaptação tem como objetivo facilitar a aplicação prática dos conceitos abordados em sala de aula.

A Tabela 1 mostra os dados do exercício de problemas de Programação Inteira escolhido para o teste do aplicativo.

Tabela 1 - Dados dos problemas de Programação Linear Inteira.

Problema	Objetivo	Função Objetivo	Restrições
1	Maximizar	$Z = 4x_1 + 2x_2 + 3x_3$	$7x_1 + 3x_2 + 6x_3 \leq 150;$ $4x_1 + 4x_2 + 5x_3 \leq 200.$
2	Maximizar	$Z = 1000x_1 + 1800x_2$	$20x_1 + 30x_2 \leq 1200;$ $x_1 \leq 40;$ $x_2 \leq 30.$
3	Minimizar	$Z = 3.0x_1 + 2.5x_2$	$4x_1 + 8x_2 \geq 32;$ $6x_1 + 6x_2 \geq 36.$
4	Minimizar	$Z = 1100x_1 + 750x_2$	$2x_1 + 2x_2 \geq 16;$ $3x_1 + x_2 \geq 12.$

Fonte: Autoria própria

A Tabela 2 resume os valores ótimos das variáveis de decisão e da função objetivo para cada problema, destacando a eficácia do aplicativo na resolução desses cenários complexos.

Tabela 2 - Valores ótimos das variáveis de decisão e função objetivo dos problemas propostos.

Problema	Resultado Ótimo	Valor Ótimo
1	$x_1 = 0;$ $x_2 = 50;$ $x_3 = 0$	$Z = 100$
2	$x_1 = 15;$ $x_2 = 30$	$Z = 69000$
3	$x_1 = 0;$ $x_2 = 6$	$Z = 15.0$
4	$x_1 = 2;$ $x_2 = 6$	$Z = 6700$

Fonte: Autoria própria

5 CONCLUSÃO

O aplicativo desenvolvido revelou-se uma ferramenta educacional intuitiva e eficaz, com potencial para auxiliar no ensino de cursos de Otimização Combinatória para engenharia e outras disciplinas relacionadas à pesquisa operacional. Ele pode ser utilizado para validar valores obtidos em sala de aula e para fomentar discussões sobre os resultados. Além disso, o aplicativo foi criado com ferramentas computacionais de fácil programação. Assim, além de servir como uma ferramenta de ensino, o aplicativo proposto pode incentivar a criação de outros programas destinados a resolver uma

variedade de problemas na área de otimização, tanto para soluções exatas quanto aproximadas, utilizando diferentes algoritmos.

De maneira geral, o aplicativo atendeu plenamente às expectativas do trabalho, tornando o processo de modelagem mais dinâmico ao reduzir a quantidade de entradas de dados necessárias. Espera-se que este trabalho contribua significativamente para a promoção da área de Otimização Combinatória na Engenharia, capacitando engenheiros a atuar nesse campo de grande relevância no mercado.

AGRADECIMENTOS

Prof. Dr. Guilbert de Arruda Souza, Prof. Dr. José Carlos Thompson da Silva e Profa. Me. Elvira Padua Lovatte.

REFERÊNCIAS

BACICH, Lilian; MORAN, José. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: Penso, 2018.

BRASIL, Reyolando MLRF; DA SILVA, Marcelo Araujo. **Otimização de projetos de engenharia**. São Paulo: Blucher, 2019.

CORRAR, Luis João; GARCIA, Edinete André da Rocha. Programação linear: uma aplicação à contabilidade de custos no processo de tomada de decisão. **Revista Cruzando Fronteras: Tendencias de Contabilidad Directiva para el Siglo XXI**, 2001.

DAWSON, Michael. **Python programming for the absolute beginner**. 3.ed. Boston, MA: Course Technology, 2010.

GARCIA, Solange; GUERREIRO, Reinaldo; CORRAR, Luiz. Teoria das restrições e programação linear. **Revista de Contabilidade do CRC-SP**, v. 2, n. 4, p. 47-61, 1998.

GOLDBARG, Marco Cesar; LUNA, Henrique Pacca. **Otimização Combinatória e Programação Linear**. 2.ed. Rio de Janeiro: Elsevier, 2005.

GUENIN, Bertrand; KÖNEMANN, Jochen; TUNCEL, Levent. **A gentle introduction to optimization**. Cambridge: Cambridge University Press, 2014.

MIYAZAWA, Flávio K.; DE SOUZA, Cid C. **Introdução à Otimização Combinatória**. In: Jornadas de Atualização em Informática - Congresso da Sociedade Brasileira de Computação - JAI-SBC, 2015.

SPEIGHT, April. **Visual Studio Code for Python Programmers**. Hoboken: Wiley, 2021.

DEVELOPMENT OF AN EDUCATIONAL APPLICATION IN PYTHON FOR LINEAR FUNCTION OPTIMIZATION WITH FLET GRAPHICAL USER INTERFACE

Abstract: This article aims to develop an educational application in Python designed for teaching Combinatorial Optimization, a topic commonly covered in Operational Research courses in Engineering. The application, built using Python along with the PuLP and Flet libraries, aims to find the optimal values of variables that maximize or minimize linear functions present in Integer Linear Programming exercises, an essential component of Combinatorial Optimization. The application's interface was designed to be intuitive and user-friendly. The results obtained through the proposed exercises demonstrate that the application is a promising tool for solving Integer Linear Programming problems, providing an accessible interface that can facilitate the teaching of Combinatorial Optimization to engineering students. Therefore, its relevance in the context of developing professionals in the field of Operational Research is highlighted.

Keywords: Combinatorial optimization, Engineering education, Python, Educational application

