Otimização para Cientistas de Dados

Um Guia Prático para Entender Otimização Linear

Versão 1.0

Alexandre Soares

Otimização para Cientistas de Dados

Um Guia Prático para Entender Otimização Linear

por

Alexandre Soares

Notas Sobre o Estudo

Todos os programas fonte e dados utilizados para a elaboração deste estudo, bem como o PDF deste material, estão disponíveis para download no repositório GitHub, no endereço: https://github.com/Alxsoa/Artigos.

Conteúdo

Lista de Figuras

Lista de Tabelas

1

Objetivo do Estudo

1.1. Estratégia de Desenvolvimento

Este estudo apresentará de maneira introdutória o conteúdo de otimização linear, baseando-se em exemplos para facilitar a compreensão bem como a aplicação prática dos conceitos teóricos. A jornada se inicia com uma introdução às fundações da programação linear, incluindo a definição de variáveis de decisão, a formulação da função objetivo e as restrições.

Este segmento introdutório é essencial para estabelecer uma base sólida e garantir a compreensão dos componentes fundamentais que compõem um problema de otimização linear.

Para solidificar o entendimento teórico, utilizamos exemplos práticos que ilustram como esses conceitos são aplicados na resolução de problemas reais. Cada exemplo é escolhido para representar uma variedade de cenários comuns, como a otimização de produção, alocação de recursos e planejamento de logística.

Através de uma abordagem passo a passo na formulação de problemas, transformação em modelos matemáticos e aplicação de métodos de solução, esclareceremos o processo de modelagem e também a relevância e utilidade da programação linear em contextos do mundo real.

Este método introdutório não só ajuda na aprendizagem e aplicação dos princípios da otimização linear, mas também destaca sua eficácia em uma variedade de aplicações práticas, reforçando assim a importância deste campo de estudo no mundo atual.

2

Programação Linear e Otimização

2.1. Introdução

A Programação Linear é uma técnica matemática amplamente utilizada para a resolução de problemas de otimização em diversos campos, tais como engenharia, economia, logística, entre outros. Este método baseia-se na modelagem matemática de um problema, a fim de encontrar a melhor solução possível.

A otimização é fundamental em diferentes aspectos, pois permite a maximização de recursos e a minimização de custos, tempo e esforço. Por esse motivo, é essencial que os alunos de nível avançado em disciplinas de matemática aplicada e engenharia aprendam os conceitos básicos de Programação Linear e Otimização.

Uma das principais razões pelas quais devemos estudar Programação Linear é a sua capacidade de simplificar problemas complexos em modelos matemáticos mais simples e compreensíveis. Por meio da Programação Linear, é possível representar de forma precisa as restrições e objetivos de um problema, o que facilita a busca pela solução ótima.

A Programação Linear é uma ferramenta poderosa para tomada de decisão em ambientes empresariais e industriais, é uma técnica que envolve a maximização ou minimização de uma função objetivo linear, sujeita a um conjunto de restrições lineares. Esses problemas são comuns em diversas áreas, como economia, engenharia, logística e ciência de dados, onde a eficiência e a otimização são cruciais para o desempenho e a tomada de decisões.

Outra vantagem da Programação Linear é a possibilidade de utilizar algoritmos eficientes para resolver problemas de grande porte. Com o avanço da tecnologia, é possível realizar cálculos complexos de forma rápida e precisa, o que torna a aplicação da Programação Linear viável em diversos cenários.

Para garantir o sucesso na utilização da Programação Linear e Otimização, é importante seguir algumas boas práticas. Em primeiro lugar, é fundamental compreender detalhadamente o problema em questão, identificando todas as variáveis, restrições e objetivos envolvidos. A correta formulação do modelo matemático é essencial para obter resultados precisos.

Além disso, é importante utilizar softwares especializados em Programação Linear, como o CPLEX, o Gurobi, entre outros, que facilitam a resolução de problemas complexos e a análise dos resultados obtidos. Essas ferramentas permitem a otimização dos processos e a obtenção de soluções eficientes.

Uma típica formulação de programação linear inclui uma função objetivo, que se deseja maximizar ou minimizar, e um conjunto de restrições que definem os limites dentro dos quais a solução deve ser encontrada. Por exemplo, em um problema de alocação de recursos, a função objetivo pode representar o lucro total a ser maximizado, enquanto as restrições podem incluir limitações de recursos disponíveis, como materiais, tempo ou mão-de-obra.

A otimização vai além da programação linear, abrangendo uma variedade de técnicas e métodos para resolver problemas em que a função objetivo ou as restrições podem ser não lineares. Isso inclui programação não linear, programação inteira, programação dinâmica e outros métodos especializados.

Em ciência de dados e aprendizado de máquina, técnicas de otimização são essenciais para treinar modelos, ajustar hiper parâmetros e encontrar soluções eficientes em problemas complexos. A capacidade de formular e resolver problemas de otimização de maneira eficaz é uma habilidade valiosa que pode levar a melhorias significativas em eficiência, custo e desempenho em muitas aplicações práticas.

3

Bibliotecas para Otimização

3.1. Principais Bibliotecas

A utilização de bibliotecas prontas em Python para resolver problemas de programação linear é crucial para a eficiência e eficácia na resolução desses problemas. Bibliotecas como SciPy [1], PuLP [3], OR-Tools [2] e Pyomo [4] oferecem ferramentas robustas e bem documentadas para modelar e resolver problemas de otimização linear.

Essas bibliotecas são desenvolvidas por especialistas e amplamente testadas, garantindo a precisão e confiabilidade dos resultados, o que é essencial para aplicações em áreas como pesquisa operacional, economia, engenharia e ciência de dados.

Além da precisão, o uso de bibliotecas prontas em Python economiza um tempo valioso que seria gasto na implementação de algoritmos de otimização do zero. Implementar métodos como o simplex ou métodos de pontos interiores requer um conhecimento profundo de algoritmos e matemática, além de ser um processo suscetível a erros.

As bibliotecas prontas abstraem essa complexidade, permitindo que os usuários se concentrem na formulação dos problemas e na interpretação dos resultados. Isso é particularmente importante em ambientes industriais e de negócios, onde a rapidez na tomada de decisões pode resultar em vantagens competitivas significativas.

Outro aspecto importante é a integração e a compatibilidade dessas bibliotecas com outras ferramentas e ecossistemas de Python. Muitas dessas bibliotecas podem ser facilmente integradas com pandas, NumPy e Matplotlib, facilitando o pré-processamento de dados, a análise e a visualização dos resultados. Isso cria um fluxo de trabalho eficiente e coeso, desde a entrada de dados até a solução e análise final.

Além disso, a comunidade ativa de desenvolvedores e usuários garante suporte contínuo, melhorias e a introdução de novas funcionalidades, mantendo as bibliotecas atualizadas e alinhadas com as necessidades do mercado e avanços tecnológicos.

3.2. Vantagens e Desvantagens das Biblioteca

As bibliotecas em Python, como SciPy, PuLP, Pyomo e OR-Tools, oferecem várias vantagens significativas para resolver problemas de otimização linear. Uma das principais vantagens é a simplicidade e a facilidade de uso.

Essas bibliotecas fornecem interfaces intuitivas para definir variáveis, restrições e funções objetivo, permitindo que até mesmo iniciantes possam formular e resolver problemas complexos com relativa facilidade.

Além disso, muitas dessas bibliotecas vêm com algoritmos de otimização altamente eficientes, como o método simplex e métodos de pontos interiores, que são otimizados para desempenho e podem lidar com problemas de grande escala de forma eficaz.

Outro benefício é a ampla integração dessas bibliotecas com outras ferramentas e ecossistemas Python, como pandas e NumPy, facilitando o pré-processamento de dados, a análise e a visualização dos resultados.

Bibliotecas	**Vantagens**	**Desvantagens**
SciPy	Fácil de usar	Necessita da função objetivo na forma de minimização
	Possui interface para solver de programação linear de alto desempenho	Necessita que as restrições sejam limitadas superiormente (não suporta restrições ”≥”)
		Não consegue resolver problemas de programação inteira (quando queremos que a solução seja composta por inteiros)
PuLP	Simples e fácil de usar para iniciantes	Não pode modelar problemas de otimização não linear
Pyomo	Possui mais funcionalidades	Requer a instalação de solvers
	Suporta uma ampla gama de problemas de otimização	Maior complexidade
	Função objetivo e expressões podem ser declaradas usando regras/funções em Python	
	Permite escrever modelos concretos ou abstratos	
OR-Tools	Suporta Python, C++, DotNet, Java	Maior complexidade
	Interface para solvers internos e externos	
	Possui bibliotecas especializadas para resolver uma ampla gama de problemas de otimização	

Tabela 3.1: Avaliação de Cada Biblioteca

No entanto, o uso de bibliotecas prontas em Python também apresenta algumas desvantagens. Uma das principais limitações é a necessidade de entender as nuances e limitações específicas de cada biblioteca. Por exemplo, algumas bibliotecas podem não suportar certos tipos de restrições ou podem ter limitações na modelagem de problemas não lineares.

Além disso, enquanto as bibliotecas abstraem muitos detalhes algorítmicos, essa abstração pode ser uma desvantagem quando se necessita de controle fino sobre o processo de otimização ou quando se enfrenta problemas de convergência.

Dependendo do problema específico, pode ser necessário um conhecimento mais profundo dos algoritmos subjacentes para ajustar parâmetros ou resolver problemas complexos que as bibliotecas padrão não conseguem lidar diretamente. Finalmente, a dependência de bibliotecas externas pode introduzir problemas de compatibilidade e manutenção, especialmente em projetos de longo prazo ou em ambientes onde a estabilidade e a longevidade do software são críticas.

4

Formulação de Problemas de Programação Linear

4.1. Como Iniciar um Problema de Programação Linear

Formular problemas de programação linear (PL) envolve a transformação de uma descrição verbal de um problema em um modelo matemático composto por uma função objetivo e um conjunto de restrições. A seguir, apresento os passos gerais para formular problemas de programação linear.

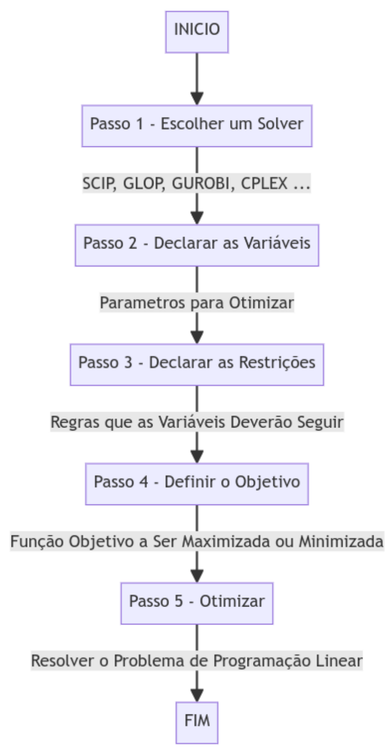

Figura 4.1: Etapas para Solução em Programação Linear

4.2. Esquema Geral do Funcionamento de Programação Linear

A programação linear é uma técnica de otimização matemática utilizada para encontrar a melhor solução possível, dada um conjunto de restrições lineares e uma função objetivo linear. As variáveis de decisão são os elementos chave que representam os valores que queremos determinar. Em um problema de PL, essas variáveis são ajustadas pelo solver para maximizar ou minimizar a função objetivo.

A função objetivo é uma expressão linear das variáveis, representando o objetivo do problema, como maximizar lucros ou minimizar custos. Essas variáveis são introduzidas no solver, que utiliza algoritmos eficientes para explorar o espaço de soluções possíveis.

As restrições em um problema de programação linear são condições lineares que as variáveis de decisão devem satisfazer. Essas restrições podem ser igualdades ou desigualdades que delimitam o conjunto de soluções viáveis. Por exemplo, em um problema de alocação de recursos, as restrições podem representar limites de capacidade, requisitos mínimos de produção, ou quaisquer outras condições que os valores das variáveis precisam respeitar.

As restrições são fundamentais para definir a região viável, que é o conjunto de todas as possíveis combinações de variáveis que atendem a todas as restrições simultaneamente.

O solver é a ferramenta computacional que processa as variáveis e as restrições para encontrar a solução ótima. Utilizando algoritmos como o método Simplex ou métodos de pontos interiores, o solver explora a região viável para identificar a melhor combinação de variáveis que maximiza ou minimiza a função objetivo.

A saída do solver é a solução ótima, que representa os melhores valores das variáveis de decisão, dadas as restrições e o objetivo do problema. Em um diagrama de fluxo, como o ilustrado acima, as variáveis e as restrições entram no solver, e o solver gera a solução ótima, que é a resposta ao problema de programação linear.

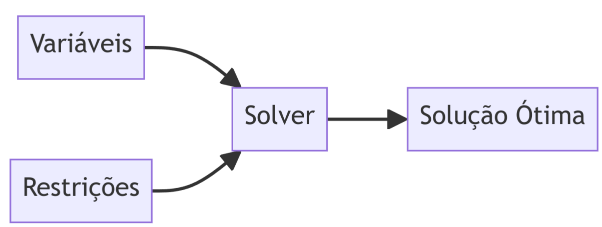

Figura 4.2: Esquema Geral do Funcionamento de Programação Linear

4.3. Selecionando um Solver

4.3.1. GLOP (Google Linear Optimization Package)

O OR-Tools tem um solver de programação linear interno chamado GLOP. Otimizado para precisão e rapidez, ele resolve problemas de programação linear de grande escala com sucesso. A solução ideal é encontrada por GLOP usando o método Simplex, onde os principais atributos são:

- Específicamente desenvolvido para resolver problemas de programação linear.
- Rápido e eficiente para problemas de grande escala.
- Utiliza a técnica Simplex.

4.3.2. SCIP (Programas de Resolução de Constraint Integer)

Um dos solvers mais rápidos e sofisticados de otimização combinatória é SCIP. Ele suporta não apenas programação inteira e inteira mista, mas também programação linear e por restrições, onde os principais atributos são:

- Pode resolver problemas de programação linear (LP), programação inteira (IP) e MIP.
- A programação por restrições é suportada.
- Adequado para tarefas complexas combinadas.

4.3.3. GUROBI

GUROBI é um solver comercial de otimização matemática que é amplamente utilizado por sua velocidade e eficiência. Ele suporta programação linear, programação inteira e inteira mista, entre outros tipos de problemas de otimização, onde os principais atributos são:

- Muito rápido e eficiente.
- Suporta uma ampla gama de problemas de otimização, incluindo LP, IP e MIP.
- Requer uma licença comercial.

4.3.4. CPLEX

CPLEX é outro solver de otimização comercial conhecido por resolver problemas de otimização de grande escala de forma eficaz. Além disso, ele fornece suporte a LP, IP e MIP, onde os principais atributos são:

- Extremamente eficaz quando se trata de problemas de grande escala.
- É compatível com programação linear, inteira e mista.
- É necessária uma licença comercial.

4.3.5. CBC(Coin-or branch and cut)

CBC é um solver de programação inteira mista (MIP) de código aberto. Ele faz parte do projeto COIN-OR, onde os principais atributos são:

- A solução é de código aberto.
- Problemas com programação inteira mista surgiram.
- Excelente opção para solvers comerciais.

4.3.6. Como Selecionar um Solver

A etapa mais importante do processo de otimização é escolher o solver correto. Isso depende da natureza do problema que você está tentando resolver.

O GLOP (Google Linear Optimization Package) é uma excelente opção para problemas puramente lineares (LP), em que todas as relações e restrições são representadas por equações lineares. GLOP é o solver de programação linear interno do OR-Tools, que é conhecido por sua capacidade de se integrar rapidamente com outras ferramentas do ecossistema do Google. Muitos profissionais o preferem por sua capacidade de resolver problemas complexos com rapidez e precisão.

A escolha do solver pode diferir de acordo com o tipo de problema, seja programação inteira (IP) ou programação inteira mista (MIP). Os solvers SCIP, GUROBI e CPLEX são os melhores para esses tipos de problemas. Vários problemas de otimização combinatória e programação linear podem ser resolvidos por SCIP, que é reconhecido por sua versatilidade.

Por outro lado, GUROBI e CPLEX são solvers comerciais conhecidos por sua rapidez e desempenho superiores na resolução de problemas complexos de grande escala. Ao escolher entre esses solvers, você deve considerar a disponibilidade de uma licença comercial e os requisitos de desempenho específicos, pois ambos oferecem recursos avançados e suporte técnico sólido.

O CBC (Coin-or branch and cut) é uma excelente alternativa para aqueles que procuram uma solução de código aberto para problemas de programação inteira mista (MIP). O CBC, um solver de otimização combinatória que faz parte do projeto COIN-OR, oferece uma boa performance sem os custos associados a solvers comerciais.

Para aqueles que precisam de uma solução eficaz sem comprometer o orçamento, o CBC é uma opção barata e robusta. No entanto, não é tão rápido quanto GUROBI ou CPLEX em todos os casos. A natureza do problema, os recursos disponíveis e as necessidades únicas de cada aplicação devem ser consideradas ao escolher entre esses solvers.

4.4. Identificação das Variáveis de Decisão

As variáveis de decisão representam as quantidades a serem determinadas para resolver o problema. Elas são geralmente denotadas por $x_1, x_2, \ldots, x_n$.

Exemplo: Em um problema de alocação de recursos, as variáveis de decisão podem representar a quantidade de cada recurso a ser alocado.

4.5. Definição da Função Objetivo

A função objetivo é a função linear que deve ser maximizada ou minimizada. Ela é uma combinação linear das variáveis de decisão.

$$\text{Maximizar (ou Minimizar)} \quad Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

Exemplo: Se queremos maximizar o lucro total, a função objetivo pode ser a soma dos lucros unitários multiplicados pelas quantidades de cada produto.

4.6. Formulação das Restrições

As restrições são as condições que as soluções devem satisfazer. Elas são expressas como equações ou inequações lineares envolvendo as variáveis de decisão.

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\leq b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\leq b_m
\end{aligned}$$

Exemplo: Em um problema de produção, as restrições podem representar limites de capacidade, como a quantidade máxima de recursos disponíveis.

4.7. Inclusão das Restrições de Não Negatividade

As variáveis de decisão geralmente devem ser não negativas, o que significa que não podemos ter quantidades negativas de recursos ou produtos.

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \ldots, \quad x_n \geq 0$$

5

Como Minimizar os Custos de Produtos

5.1. Ambiente da Indústria

A indústria de alimentos para gatos é um segmento crescente e dinâmico do mercado pet, impulsionado por um aumento na adoção de gatos como animais de estimação e uma crescente conscientização sobre a saúde e o bem-estar dos felinos.

Esta indústria oferece uma ampla gama de produtos, desde alimentos secos e úmidos até opções premium e especializadas que atendem às necessidades dietéticas específicas de gatos de diferentes idades, raças e condições de saúde.

As empresas investem significativamente em pesquisa e desenvolvimento para formular alimentos balanceados e nutritivos, incorporando ingredientes de alta qualidade e frequentemente promovendo benefícios como a saúde do trato urinário, controle de peso e melhora na pelagem.

Além disso, a demanda por alimentos naturais e orgânicos também está em ascensão, refletindo as tendências gerais de consumo humano. A competição é intensa, com grandes marcas estabelecidas e novas empresas inovadoras disputando a lealdade dos consumidores, muitas vezes através de estratégias de marketing digital e programas de fidelidade.

5.2. Contexto do Problema

No ambiente competitivo para alimentação felina a Miau Alimentos, deseja elaborar uma nova receita de ração e assim aumentar sua participação de mercado, porém esta nova receita deve ter um custo mínimo de produção e um sabor diferenciado para os felinos.

Para alcançar este objetivo, o departamento de pesquisa e desenvolvimento da Miau Alimentos deve misturar dois tipos de alimentos, que chamaremos Alimento Tipo 1 e Tipo 2, assim a cada 100 gramas do Alimento 1 deve ser adicionado 10 gramas de um nutriente especialmente criado e como estamos falando de uma fórmula secreta, a mesma será referida como nutriente X. Adicionalmente ao nutriente X deve ser inserido na receita 50 gramas do nutriente Y além de 40 gramas do nutriente Z.

O Alimento 2, deve conter para cada 100 gramas, 20 gramas do nutriente X, 60 gramas do nutriente Y e 20 gramas do nutriente Z. Porém deve ser considerado os custos de produção, ou seja, a cada 100 gramas do Alimento 1, a Miau Alimentos gasta 0,60 dólares e a cada 100 gramas do Alimento 2, 0,80 dólares.

A fórmula final do novo alimento deve conter no mínimo 2 gramas do nutriente X, 64 gramas do nutriente Y e por fim 34 gramas do nutriente Z. Com isto posto a Miau Alimentos, deseja saber qual o custo mínimo de fabricação.

5.3. Estruturando o Problema da Miau Alimentos

Após ler o problema e entender as demandas do mesmo, torna-se necessário estruturar as informações, para tanto a Tabela-5.1 oferece de maneira organizada, todas as informações necessárias a solução do problema.

Requerimentos	**Composição para Cada 100g**		**Composição dos Nutrientes**
	Alimento 1	**Alimento 2**	
Nutriente X	10	20	2
Nutriente Y	40	60	64
Nutriente Z	50	20	34
Custo por 100 g	0,60	0,80	

Tabela 5.1: Estrutura do Problema da Miau Alimentos

5.4. Definindo a Função Objetivo

O problema proposto claramente busca uma minimização, ou seja, deverá ser usado x_1 gramas do Alimento 1, e x_2 gramas do Alimento 2, que deverão ter o custo mínimo, porém o problema se refere a 100 gramas e precisamos ter a unidade de 0.006 e 0.008.

Com estes pontos entendidos, poderemos escrever a função objetivo que obviamente deve ser minimizada, nas condições de não negatividade, ou seja: $x_1 \geq 0$ e $x_2 \geq 0$, teremos $0.006 \cdot x_1 + 0.008 \cdot x_2$.

5.5. Definindo as Restrições

Recorrendo a Tabela-5.1, podemos escrever as restrições por nutriente, assim obteremos o resultado apresentado na Tabela-5.2.

Nutriente	**Equação**
Nutriente X	$0.1 \cdot x_1 + 0.2 \cdot x_2 \geq 2$
Nutriente Y	$0.4 \cdot x_1 + 0.6 \cdot x_2 \geq 64$
Nutriente Z	$0.5 \cdot x_1 + 0.2 \cdot x_2 \geq 34$

Tabela 5.2: Estrutura das Restrições

5.6. Implementando a Solução do Problema da Miau Alimentos

Nesta seção, será apresentado o código escrito na linguagem Python, onde cada bloco, será precedido de comentários que visam facilitar o entendimento.

Importando as Biliotecas Necessárias

```
from ortools.linear_solver import pywraplp
from scipy.optimize import linprog
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
```

Instância o Google Linear Optimization Package como solver

```
solver = pywraplp.Solver.CreateSolver('GLOP')
```

Criando as variáveis prontas para assumir qualquer valor não negativo

```
x1 = solver.NumVar(0, solver.infinity(), 'x1')
x2 = solver.NumVar(0, solver.infinity(), 'x2')
x3 = solver.NumVar(0, solver.infinity(), 'x3')
```

Definindo as Restrições

```
solver.Add(0.1 * x1 + 0.2 * x2 >= 2)
solver.Add(0.4 * x1 + 0.6 * x2 >= 64)
solver.Add(0.5 * x1 + 0.2 * x2 >= 34)
```

Definindo a Função Objetivo

```
solver.Minimize(0.006 * x1 + 0.008 * x2)
```

Executa a Solução e Mostra um Resumo dos Resultados

```python
status = solver.Solve()
if status == pywraplp.Solver.OPTIMAL or status == pywraplp.Solver.FEASIBLE:
    print ( "#"*72)
    print ( "# Resultado Obtido para as Variáveis do Problema ")
    print ( "#"*72)
    print ( "# Objetivo .: ", solver.Objective().Value())
    print ( "# x1 .......: ", x1.solution_value())
    print ( "# x2 .......: ", x2.solution_value())
    print ( "#"*72)

    print ()
    print ( "#"*72)
    print ( "# Resumo Operacional ")
    print ( "#"*72)

    print (f"# Versão do Solver .......: {solver.SolverVersion()}")
    print (f"# Número de Variáveis ....: {solver.NumVariables()}")
    print (f"# Número de Restrições ...: {solver.NumConstraints()}")
    print ( "# Gasto de Tempo .........:", solver.wall_time(), "Milissegundo ")
    print ( "# Intereções Necessárias .:", solver.iterations(), "Intereções")
    print ( "#"*72)
else:
    print ( "#"*72)
    print('O problema não tem uma solução ótima.')
    print ( "#"*72)
```

Executa os Gráficos de Resultado

```
Presentes e Detalhados no Programa Fonte no GitHub
```

5.7. Resultado Obtido Pelo Programa

Abaixo, é apresentado a saída com os resultados obtidos pelo programa que implementa a programação linear.

Saída do Programa Anteriormente Apresentado

```
########################################################################
# Resultado Obtido para as Variáveis do Problema
########################################################################
# Objetivo .:  0.8763636363636363
# x1 .......:  34.54545454545455
# x2 .......:  83.63636363636363
########################################################################

########################################################################
# Resumo Operacional
########################################################################
# Versão do Solver .......: Glop solver v9.10.4067
# Número de Variáveis ....: 3
# Número de Restrições ...: 3
# Gasto de Tempo .........: 2669 Milissegundo
# Intereções Necessárias .: 2 Intereções
########################################################################
```

5.8. Interpretando os Resultados Obtidos

Em resumo, os gráficos fornecem uma visão detalhada das restrições e da região viável do problema de otimização, permitindo identificar a solução ótima visualmente e entender como cada restrição contribui para a definição dessa solução.

O primeiro gráfico (Figura-5.1) detalha as restrições $R1$, $R2$ e $R3$ com suas respectivas equações. A interseção dessas linhas determina a região viável, destacada em azul claro. O ponto (35, 84) é a solução ótima, onde $R2$ e $R3$ se encontram, indicando que ambas as restrições são ativas na solução. Os pontos de interseção das restrições com os eixos ajudam a definir os limites da região viável.

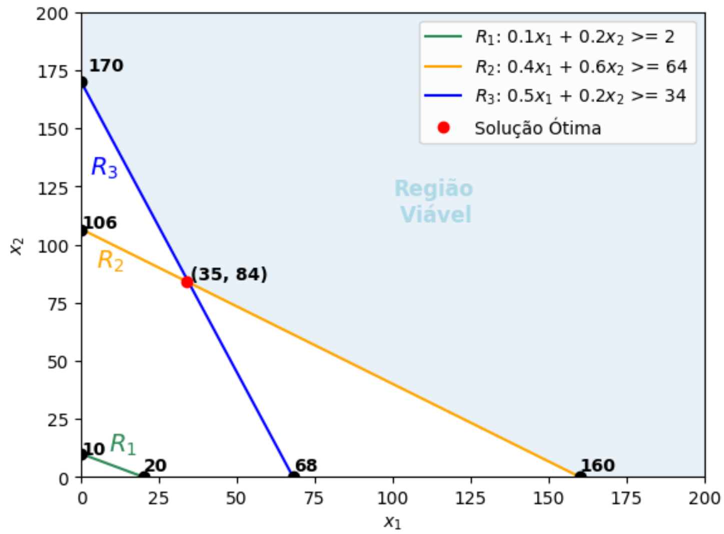

Figura 5.1: Identificação da Solução Ótima

O segundo gráfico (Figura-5.2) foca na restrição $R3$ $(0.5x_1 + 0.2x_2 \geq 34)$. Esta linha, representada em azul, cruza os pontos (68, 0) e (0, 170), delimitando a região viável para essa restrição específica. A área azul claro indica onde a restrição $R3$ é satisfeita.

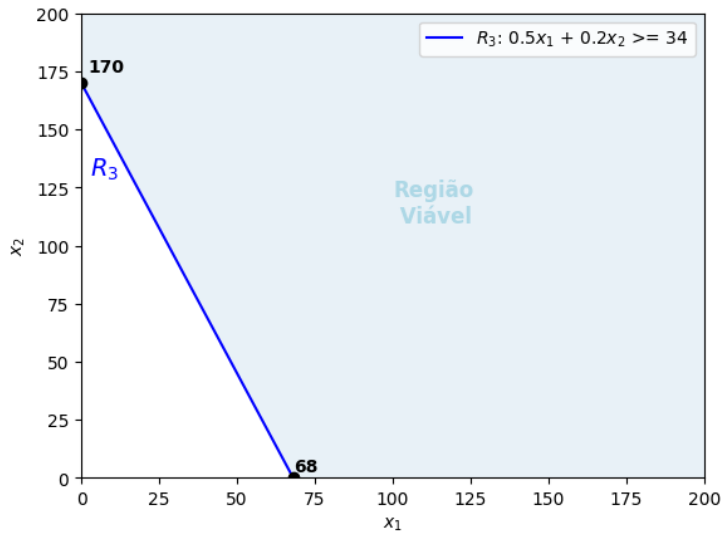

Figura 5.2: Identificação da Restrição $R3$

O terceiro gráfico (Figura-5.3) isola a restrição $R2$ $(0.4x_1 + 0.6x_2 \geq 64)$, mostrada em laranja. Esta linha passa pelos pontos (160, 0) e (0, 106). A região viável, onde esta restrição é satisfeita, é mostrada em azul claro.

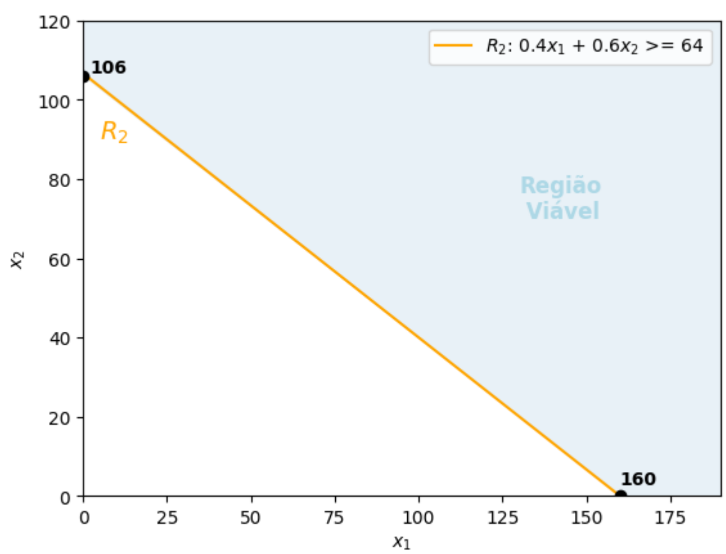

Figura 5.3: Identificação da Restrição $R2$

O quarto gráfico (Figura-5.4) apresenta a restrição $R1$ ($0.1x_1 + 0.2x_2 \geq 2$), destacada em verde. A linha passa pelos pontos (20, 0) e (0, 10). A região viável para essa restrição está novamente em azul claro, mostrando onde $R1$ é satisfeita.

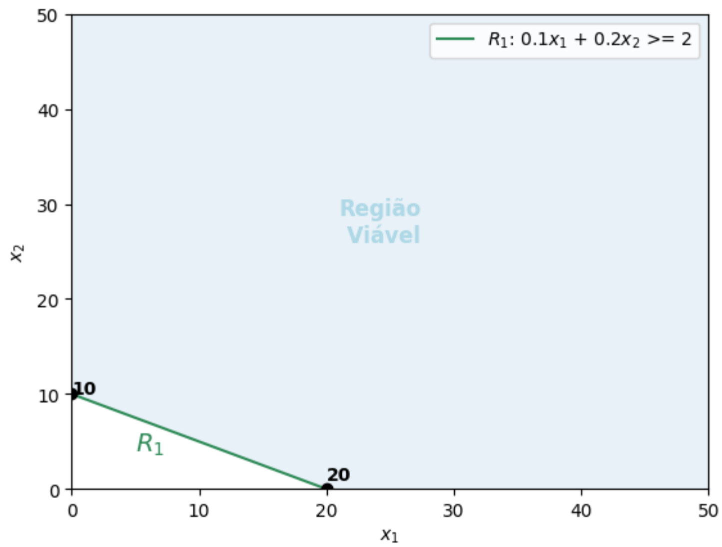

Figura 5.4: Identificação da Restrição $R1$

5.9. Como Calculamos as Retas Suporte

A partir do gráfico (Figura-5.1), podemos identificar três pontos que chamaremos de A, B e C como apresentado na Figura-5.5.

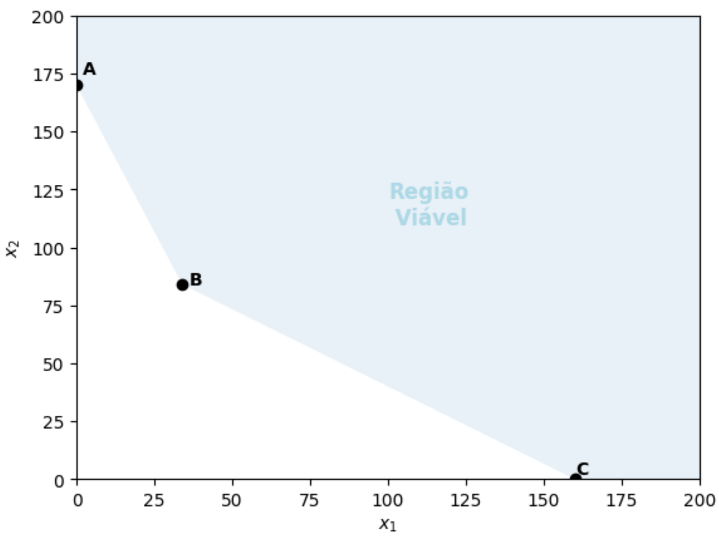

Figura 5.5: Pontos da Região Viável

Para auxiliar no cálculo das retas suporte, foi utilizado a Tabela-5.3, como o problema descrito é uma minimização, deveremos escolher o menor valor obtido pela função objetivo, neste caso o ponto B.

Ponto	**Valor x_1**	**Valor x_2**	**Função Objetivo** $\mathbf{0.006 \cdot x_1 + 0.008 \cdot x_2}$
A	0	170	1.36
B	34.5	83.6	0.88
C	160	0	0.96

Tabela 5.3: Estrutura de Cálculo das Retas Suporte

Nesta etapa, basta construir as equações com os valores anteriormente tabulados, que obteremos facilmente as equações abaixo:

$$\mathbf{0.006} \cdot \boldsymbol{x_1} + \mathbf{0.008} \cdot \boldsymbol{x_2} = 1.36$$
$$\mathbf{0.006} \cdot \boldsymbol{x_1} + \mathbf{0.008} \cdot \boldsymbol{x_2} = 0.88$$
$$\mathbf{0.006} \cdot \boldsymbol{x_1} + \mathbf{0.008} \cdot \boldsymbol{x_2} = 0.96$$

Como base nas equações anteriormente apresentadas, basta calcular as retas suporte utilizando as equações abaixo:

$$\boldsymbol{x_2} = \frac{1.36 - 0.006 \cdot x_1}{0.008}$$

$$\boldsymbol{x_2} = \frac{0.88 - 0.006 \cdot x_1}{0.008}$$

$$\boldsymbol{x_2} = \frac{0.96 - 0.006 \cdot x_1}{0.008}$$

5.10. Apresentação das Retas Suporte

O gráfico apresentado pela Figura-5.6 mostra a região viável delineada pelas restrições $Z1$, $Z2$ e $Z3$. A área azul claro representa todas as combinações de x_1 e x_2 que satisfazem todas as restrições simultaneamente. O ponto vermelho indica a solução ótima, onde a função objetivo é maximizada. As linhas de nível $Z1$, $Z2$ e $Z3$ representam diferentes valores da função objetivo, com a linha $Z3$ passando pela solução ótima.

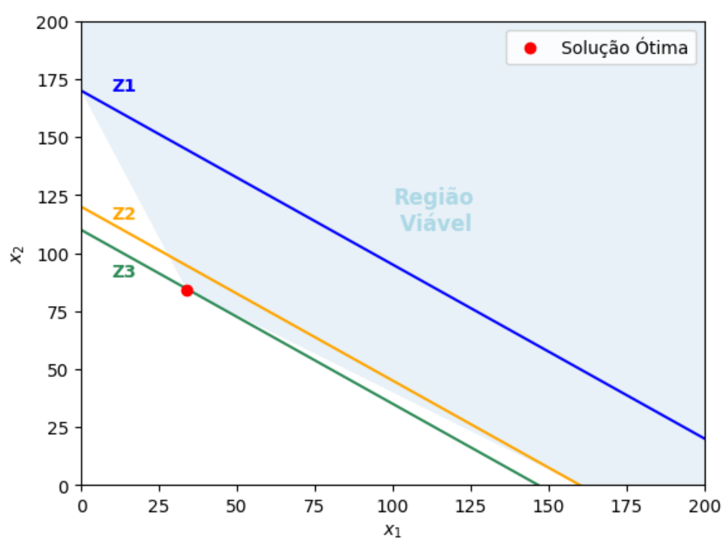

Figura 5.6: Identificação das Retas de Suporte

5.11. Conclusão

Os gráficos também destacam a solução ótima, que é o ponto onde a função objetivo atinge seu valor máximo dentro da região viável. No segundo gráfico, a solução ótima é indicada pelo ponto (35, 84), onde as restrições R_2 e R_3 se encontram, mostrando que essas restrições são ativas na solução ótima.

Cada gráfico adicional detalha individualmente como uma restrição específica contribui para a definição da região viável, demonstrando a interseção das restrições com os eixos e a área onde cada restrição é satisfeita. Esta abordagem visual facilita a compreensão de como as restrições interagem e definem o espaço de soluções possíveis.

Em conclusão, a combinação dessas representações gráficas proporciona uma visão abrangente e intuitiva do problema de otimização linear, permitindo uma análise clara da solução ótima e das contribuições de cada restrição. Essa visualização é uma ferramenta poderosa para a tomada de decisões informadas e para o desenvolvimento de estratégias de otimização eficazes em diversos campos, como logística, produção e finanças.

6

Como Maximizar a Receita de Produtos

6.1. Ambiente da Indústria

No ambiente altamente competitivo da indústria, o lançamento de novos produtos é uma das estratégias mais cruciais para sustentar o crescimento e a inovação. Empresas de diversos setores, desde tecnologia até bens de consumo, enfrentam o desafio constante de desenvolver produtos que não apenas atendam às necessidades dos consumidores, mas também se destaquem no mercado.

O sucesso de um lançamento de produto pode determinar a posição de uma empresa em seu segmento, influenciando diretamente sua receita e participação de mercado. No entanto, a pressão para inovar rapidamente e a necessidade de diferenciar-se da concorrência tornam este processo extremamente complexo e arriscado.

Maximizar a receita no lançamento de produtos envolve uma série de desafios estratégicos e operacionais. Primeiramente, é essencial realizar uma análise de mercado abrangente para identificar tendências, comportamentos do consumidor e lacunas no mercado que o novo produto pode preencher.

Este conhecimento permite que a empresa posicione seu produto de maneira eficaz e desenvolva estratégias de marketing direcionadas. Além disso, é fundamental estabelecer um preço competitivo que balanceie a percepção de valor do consumidor com a necessidade de recuperar os custos de desenvolvimento e gerar lucro.

Outro aspecto crítico é a gestão da cadeia de suprimentos e a capacidade de produção. Para maximizar a receita, as empresas precisam garantir que possam atender à demanda prevista sem enfrentar problemas de estoque ou atrasos na entrega.

A implementação de tecnologias avançadas, como análises preditivas e sistemas de gestão de inventário, pode ajudar a antecipar necessidades e otimizar a alocação de recursos. Além disso, a capacidade de ajustar rapidamente as estratégias com base no feedback do mercado durante o lançamento pode significar a diferença entre o sucesso e o fracasso.

Em suma, maximizar a receita no lançamento de novos produtos exige uma abordagem integrada que alinha insights de mercado, estratégias de preço, distribuição eficiente e operações robustas.

6.2. Contexto do Problema

A fábrica Parafuso Brilhante, enfrenta o desafio de produzir dois novos produtos, cada um dos quais requer tanto material quanto mão-de-obra. A produção de cada produto consome recursos que são limitados: a fábrica dispõe de apenas 30 unidades de material e 20 unidades de mão-de-obra. Além disso, a venda de cada produto gera receita, e o objetivo da fábrica é maximizar essa receita total.

Para alcançar esse objetivo, é essencial construir um plano de produção que utilize os recursos disponíveis de forma eficiente, evitando desperdícios e garantindo que os recursos sejam alocados de maneira a gerar o maior retorno possível.

Para cada unidade do Produto 1, são necessárias 2 unidades de material e 4 unidades de mão-de-obra; para cada unidade do Produto 2, são necessárias 5 unidades de material e 2 unidades de mão-de-obra.

Ao resolver este modelo, a fábrica pode determinar as quantidades ótimas de cada produto que devem ser produzidas para maximizar a receita, respeitando os limites de material e mão-de-obra. Ao utilizar essa abordagem, a fábrica pode garantir que está aproveitando ao máximo seus recursos limitados, otimizando a produção e, consequentemente, aumentando sua receita.

6.3. Estruturando o Problema da Parafuso Brilhante

Após ler o problema e entender as demandas do mesmo, torna-se necessário estruturar as informações, para tanto a Tabela-6.1 oferece de maneira organizada, todas as informações necessárias a solução do problema.

Requerimentos	**Produtos a Fabricar**		**Estoques**
	Produto 1	**Produto 2**	
Material	2	5	30
Mão de Obra	4	2	20
Lucro	3	4	

Tabela 6.1: Estrutura do Problema da Parafuso Brilhante

6.4. Definindo a Função Objetivo

O problema proposto claramente busca uma maximização, ou seja, deverá ser usado x_1 representando o produto 1, e x_2 para o produto 2, que deverão ter a receita máxima.

Com estes pontos entendidos, poderemos escrever a função objetivo que obviamente deve ser maximizada, nas condições de não negatividade, ou seja: $x_1 \geq 0$ e $x_2 \geq 0$, teremos $3 \cdot x_1 + 4 \cdot x_2$.

6.5. Definindo as Restrições

Recorrendo a Tabela-5.1, podemos escrever as restrições por nutriente, assim obteremos o resultado apresentado na Tabela-5.2.

Produto	**Equação**
Produto 1	$2 \cdot x_1 + 5 \cdot x_2 \leq 30$
Produto 2	$4 \cdot x_1 + 2 \cdot x_2 \leq 20$

Tabela 6.2: Estrutura das Restrições da Parafuso Brilhante

6.6. Implementando a Solução do Problema de Receita

Nesta seção, será apresentado o código escrito na linguagem Python, onde cada bloco, será precedido de comentários que visam facilitar o entendimento.

Neste bloco estamos incluindo todas as bibliotecas necessárias a execução do programa

```
import numpy as np
from ortools.linear_solver import pywraplp
from scipy.optimize import linprog
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
```

Estamos iniciando o solver com a instância do GLOP(Google Linear Optimization Package).

```
solver = pywraplp.Solver.CreateSolver('GLOP')
```

Criando as duas variáveis do problema e assumindo qualquer valor não negativo as mesmas

```
x1 = solver.NumVar(0, solver.infinity(), 'x1')
x2 = solver.NumVar(0, solver.infinity(), 'x2')
```

Definindo as restrições do problema ao solver instânciado

```
solver.Add(2 * x1 + 5 * x2 <= 30.0)
solver.Add(4 * x1 + 2 * x2 <= 20.0)
```

Definindo a função objetivo do problema e inicia-se a execução

```
solver.Maximize(3 * x1 + 4 * x2)
```

Após terminado a execução é verificado se existe solução

```
status = solver.Solve()

if status == pywraplp.Solver.OPTIMAL:
    print('Valor do Objetivo =', solver.Objective().Value())
    print(f'(x1, x2): ({x1.solution_value():.2}, {x2.solution_value():.2})')
else:
    print('O problema não tem uma solução ótima.')
```

Visão Gráfica da região viável e ponto ótimo

```
Código Detalhado no Arquivo Fonte Disponível no Repositório GitHub
```

6.7. Resultado Obtido pelo Programa

Abaixo, é apresentado a saída com os resultados obtidos pelo programa que implementa a programação linear.

Saída do Programa Anteriormente Apresentado

```
######################################################################
# Resultado Obtido para as Variáveis do Problema
######################################################################
# Objetivo .:  27.5
# x1 .......:  2.4999999999999996
# x2 .......:  5.0
######################################################################

######################################################################
# Resumo Operacional
######################################################################
# Versão do Solver .......: Glop solver v9.10.4067
# Número de Variáveis ....: 2
# Número de Restrições ...: 2
# Gasto de Tempo .........: 2069 Milissegundo
# Intereções Necessárias .: 2 Intereções
######################################################################
```

6.8. Interpretando os Resultados Obtidos

Os gráficos nesta seção, mostram como as restrições de recursos limitam a produção e como a programação linear pode ser usada para encontrar a combinação de produção que maximiza a receita, respeitando essas limitações. A solução ótima é visualmente identificada na interseção das restrições, destacando a eficiência e utilidade da programação linear na tomada de decisões em ambientes de produção.

A Figura-6.1 mostra a restrição R_1, representada pela equação $2x_1 + 5x_2 \leq 30$. A linha verde no gráfico delimita a região onde esta restrição é válida. Os pontos onde a linha encontra os eixos x_1 e x_2 são importantes para entender os limites da restrição: quando $x_1 = 0$, x_2 pode ser até 6 unidades, e quando $x_2 = 0$, x_1 pode ser até 15 unidades. Esses pontos formam a fronteira da região viável definida por R_1.

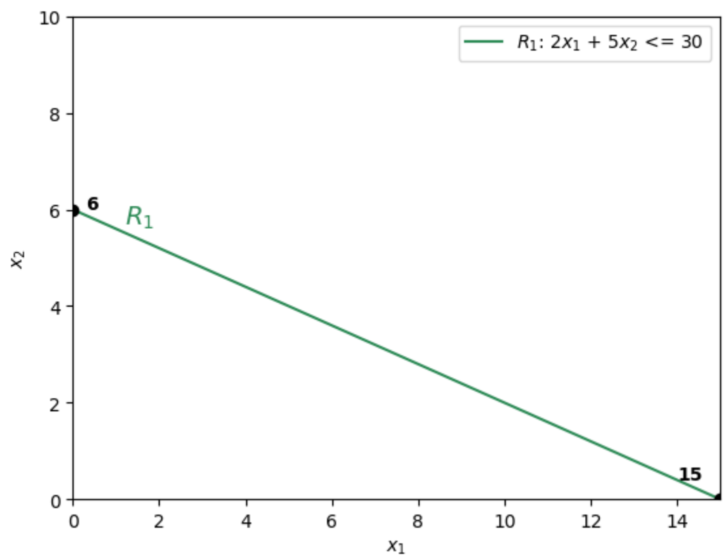

Figura 6.1: Identificação da restrição R1

A Figura-6.2 apresenta a restrição R_2, descrita pela equação $4x_1 + 2x_2 \leq 20$. A linha laranja marca a região onde esta restrição é atendida. Similarmente ao gráfico anterior, os pontos de interseção com os eixos são essenciais: quando $x_1 = 0$, x_2 pode ser até 10 unidades, e quando $x_2 = 0$, x_1 pode ser até 5 unidades. Esses pontos ajudam a delinear a área onde a produção é possível considerando a restrição de R_2.

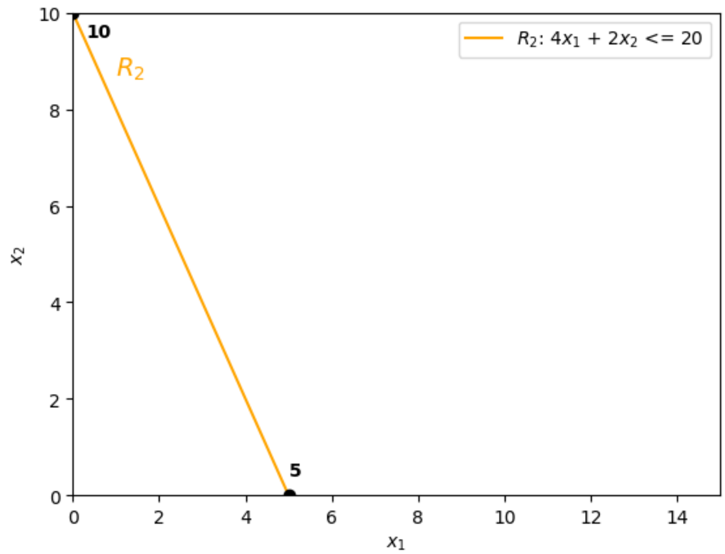

Figura 6.2: Identificação da restrição R2

A Figura-6.3 combina as duas restrições R_1 e R_2 para ilustrar a região viável (em azul claro), onde ambas as restrições são satisfeitas simultaneamente. A interseção das duas linhas define os limites da região viável, e o ponto vermelho representa a solução ótima. Este ponto é a melhor combinação de x_1 e x_2 que maximiza a receita total, considerando as limitações impostas pelo material e mão-de-obra disponíveis. A solução ótima é encontrada na interseção das restrições R_1 e R_2, indicando que ambas as restrições são ativas nesse ponto.

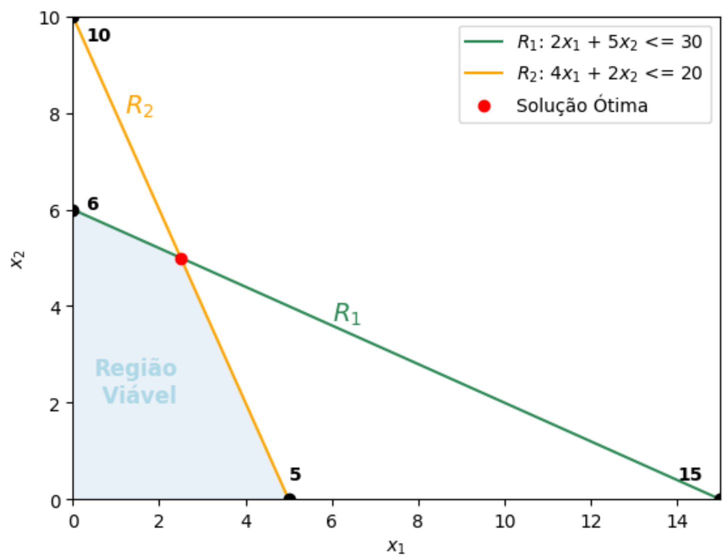

Figura 6.3: Identificação da Região Ótima

6.9. Como Calculamos as Retas Suporte

A partir da Figura-6.3), podemos identificar três pontos que chamaremos de A, B e C como apresentado na Figura-6.4.

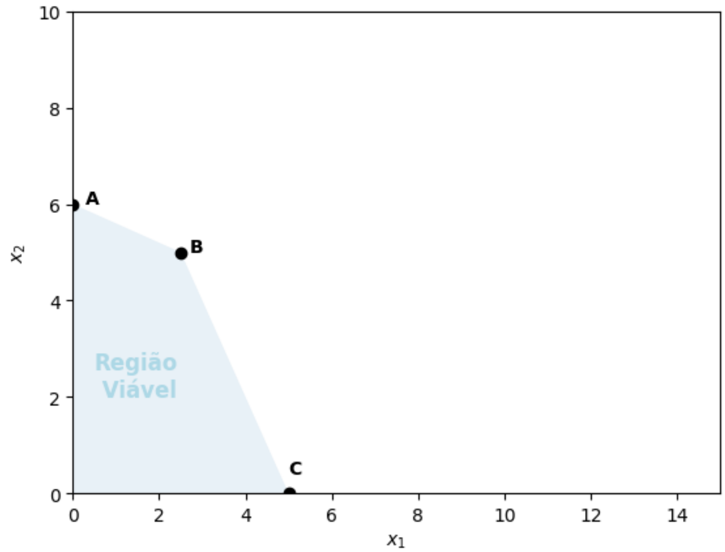

Figura 6.4: Pontos da Região Viável

Para auxiliar no cálculo das retas suporte, foi utilizado a Tabela-5.3, como o problema descrito é uma minimização, deveremos escolher o menor valor obtido pela função objetivo, neste caso o ponto B.

Ponto	Valor x_1	Valor x_2	Função Objetivo $3 \cdot x_1 + 4 \cdot x_2$
A	0	6	24
B	2.5	5	27.5
C	5	0	15

Tabela 6.3: Estrutura de Cálculo das Retas Suporte

Nesta etapa, basta construir as equações com os valores anteriormente tabulados, que obteremos facilmente as equações abaixo:

$$\mathbf{3} \cdot \boldsymbol{x_1} + \mathbf{4} \cdot \boldsymbol{x_2} = 24$$
$$\mathbf{3} \cdot \boldsymbol{x_1} + \mathbf{4} \cdot \boldsymbol{x_2} = 27.5$$
$$\mathbf{3} \cdot \boldsymbol{x_1} + \mathbf{4} \cdot \boldsymbol{x_2} = 15$$

Como base nas equações anteriormente apresentadas, basta calcular as retas suporte utilizando as equações abaixo:

$$\boldsymbol{x_2} = \frac{24 - 3 \cdot x_1}{4}$$

$$\boldsymbol{x_2} = \frac{27.5 - 3 \cdot x_1}{4}$$

$$\boldsymbol{x_2} = \frac{15 - 3 \cdot x_1}{4}$$

6.10. Apresentação das Retas Suporte

A Figura-6.5 mostra a região viável delineada pelas restrições $Z1$, $Z2$ e $Z3$. A área azul claro representa todas as combinações de x_1 e x_2 que satisfazem todas as restrições simultaneamente. O ponto vermelho indica a solução ótima, onde a função objetivo é maximizada. As linhas de nível $Z1$, $Z2$ e $Z3$ representam diferentes valores da função objetivo, com a linha $Z3$ passando pela solução ótima.

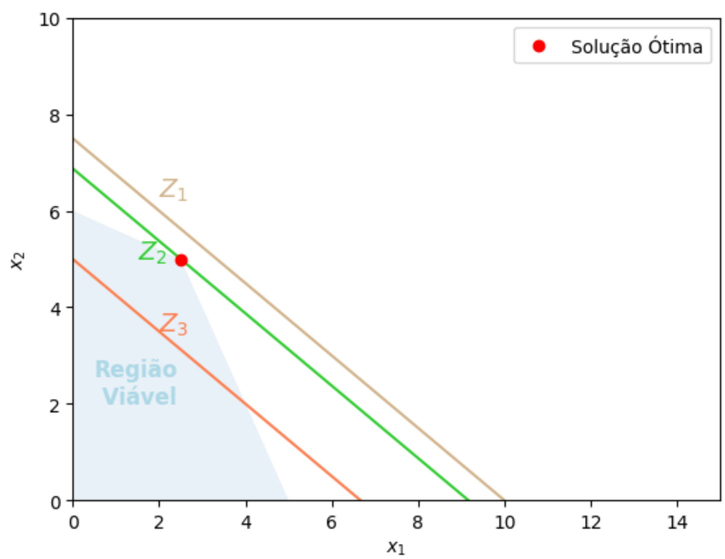

Figura 6.5: Identificação das Retas de Suporte

6.11. Conclusão

As imagens fornecem uma representação visual clara do problema de otimização linear, destacando as restrições e a região viável para a solução ótima. O primeiro gráfico ilustra a restrição R_1 ($2x_1+5x_2 \leq 30$), com a linha verde delimitando a área onde essa restrição é satisfeita.

As interseções da linha verde com os eixos x_1 e x_2 mostram os limites da quantidade máxima de recursos que podem ser alocados para x_1 e x_2, respectivamente. Este gráfico estabelece uma base para entender como os recursos disponíveis impõem limites à produção.

O segundo gráfico mostra a restrição R_2 ($4x_1 + 2x_2 \leq 20$), com a linha laranja indicando onde essa restrição é atendida. Novamente, as interseções com os eixos ajudam a visualizar os limites máximos para x_1 e x_2 dentro desta restrição.

A interseção das duas restrições, mostrada no terceiro gráfico, destaca a região viável (em azul claro) onde ambas as restrições são simultaneamente satisfeitas. A solução ótima, indicada pelo ponto vermelho, é encontrada na interseção das restrições R_1 e R_2, mostrando a melhor combinação de x_1 e x_2 que maximiza a receita, dadas as limitações de recursos.

O último gráfico mostra as linhas de nível da função objetivo, representadas como Z_1, Z_2 e Z_3. A linha verde Z_2 toca a solução ótima, destacando o valor máximo da função objetivo dentro da região viável. Esse gráfico adicional ajuda a visualizar como a função objetivo varia dentro da região viável e reforça a importância da interseção das restrições para encontrar a solução ótima.

A conclusão geral dessas visualizações é que a otimização linear, utilizando gráficos para representar restrições e a região viável, é uma ferramenta poderosa para maximizar a receita em cenários de produção limitados por recursos.

7

Desafios e Tendências Futuras em Otimização

7.1. Entendimento do Futuro

Na prática da otimização, os profissionais frequentemente enfrentam desafios significativos, especialmente relacionados à incerteza e variabilidade dos dados. Muitas vezes, os modelos de otimização assumem que os parâmetros são conhecidos e fixos, mas no mundo real, esses parâmetros podem variar e serem incertos.

Esta incerteza pode surgir de várias fontes, como mudanças nas condições de mercado, flutuações na disponibilidade de recursos ou dados imprecisos. Além disso, a complexidade dos problemas de otimização em grande escala também representa um desafio. Problemas com um grande número de variáveis e restrições podem ser computacionalmente intensivos, exigindo algoritmos mais eficientes e técnicas de decomposição para serem resolvidos em tempo hábil.

Para lidar com a incerteza, a otimização robusta e a otimização estocástica estão ganhando destaque como áreas de pesquisa promissoras. A otimização robusta busca encontrar soluções que sejam viáveis sob uma variedade de condições possíveis, garantindo que a solução permaneça eficaz mesmo quando os parâmetros do modelo variam dentro de determinados limites.

Por outro lado, a otimização estocástica incorpora a incerteza diretamente no modelo, utilizando distribuições de probabilidade para representar parâmetros incertos. Isso permite a formulação de soluções que otimizam o desempenho esperado, levando em consideração a variabilidade dos dados. Ambas as abordagens oferecem maneiras poderosas de melhorar a resiliência e a eficácia das soluções de otimização em cenários reais.

Outra tendência emergente na otimização é a integração com técnicas de machine learning. O machine learning pode ser utilizado para aprimorar os modelos de otimização de várias maneiras. Por exemplo, algoritmos de aprendizado podem ser aplicados para estimar parâmetros desconhecidos, prever a demanda futura ou identificar padrões em dados históricos que podem informar decisões de otimização.

Além disso, técnicas de aprendizado por reforço podem ser usadas para desenvolver políticas de controle adaptativo, que ajustam dinamicamente as decisões de otimização com base em feedback contínuo. Essa sinergia entre machine learning e otimização abre novas possibilidades para resolver problemas complexos de maneira mais eficiente e eficaz.

O futuro da otimização será moldado por avanços tecnológicos contínuos e novas áreas de pesquisa. O aumento do poder computacional, juntamente com o desenvolvimento de algoritmos mais avançados, permitirá resolver problemas de otimização que antes eram considerados intratáveis. Tecnologias emergentes, como computação quântica, também podem revolucionar o campo, oferecendo novas maneiras de abordar problemas de otimização em grande escala.

Devido a crescente disponibilidade de grandes volumes de dados (big data) e a evolução das técnicas de análise de dados permitirão modelos de otimização mais precisos e informados. A combinação dessas tendências promete ampliar significativamente as capacidades e aplicações da otimização em diversos setores, desde a logística e manufatura até a saúde e finanças.

Referências

[1] *Fundamental algorithms for scientific computing in Python*. Acessado em 12.07.2024. 2024. URL: https://scipy.org.

[2] *linear optimization*. Acessado em 12.07.2024. 2024. URL: https://developers.google.com/optimization/lp.

[3] *PuLP is an LP modeler written in python*. Acessado em 12.07.2024. 2024. URL: https://pypi.org/project/PuLP/.

[4] *Pyomo is a Python-based open-source software package*. Acessado em 12.07.2024. 2024. URL: https://www.pyomo.org.