Velton Cardoso Pires

Utilização de Ferramentas Computacionais para a Resolução de Problemas de Programação Linear

Vitória da Conquista

2018

Velton Cardoso Pires

Utilização de Ferramentas Computacionais para a Resolução de Problemas de Programação Linear

...

Universidade Estadual do Sudoeste da Bahia – UESB

Departamento de Ciências Exatas e Tecnológicas – DCET

Licenciatura em Matemática

Orientador: Gonçalo Renildo Lima Cerqueira

Vitrória da Conquista

2018

Velton Cardoso Pires

Utilização de Ferramentas Computacionais para a Resolução de Problemas de Programação Linear

...

Trabalho aprovado. Vitrória da Conquista, _____/ _____/ 2018

Gonçalo Renildo Lima Cerqueira
Orientador

Professor
Sérgio da Silva Aguiar

Professor
Marlos André Marques Simões de Oliveira

Vitrória da Conquista
2018

RESUMO

Apresentamos neste trabalho uma proposta para resolução de problemas de otimização utilizando os softwares GeoGebra e LINGO. O GeoGebra é um software de matemática dinâmica para todos os níveis de ensino que reúne Geometria, Álgebra, Gráficos, Planilha de Cálculo, Probabilidade, Estatística e Cálculos Simbólicos em um único programa. O LINGO foi desenvolvido para resolver problemas de Programação Linear. Embora o GeoGebra possa ser utilizado para outros fins, como citado, no nosso trabalho utilizamos exclusivamente para construção gráfica dos problemas de otimização linear que resolvemos, já o LINGO foi utilizado para resolver algebricamente tais problemas. A vantagem de utilizá-lo está na possibilidade de poder resolver problemas com um número grande de variáveis e da facilidade de se inserir os dados, além da gratuidade da sua utilização. Uma aplicação prática surge nos problemas de corte de estoque unidimensional.

Palavras-chave: Otimização Combinatória, Ferramentas computacionais, Problema de Corte de Estoque Unidimensional.

ABSTRACT

We present in this work a proposal for solving optimization problems using GeoGebra and LINGO softwares. GeoGebra is a dynamic math software for all levels of education that combines Geometry, Algebra, Graphs, Spreadsheet, Probability, Statistics and Symbolic Calculations in a single program. LINGO was developed to solve Linear Programming problems. Although GeoGebra can be used for other purposes, as mentioned, in our work we used exclusively for graphical construction of the problems of linear optimization that we solve, since LINGO was used to algebrically solve such problems. The advantage of using it is to be able to solve problems with a large number of variables and the ease of inserting the data, besides the gratuity of its use. A practical application arises in the problems of cutting one-dimensional stock.

Keywords: Combinatorial optimization, Computational tools, Integer Linear Optimization, Problems of Cutting One-Dimensional Stock.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PL	Programação Linear
PPL	Problema de Programação Linear
PLI	Problemas de Programação Inteira

LISTA DE SÍMBOLOS

α	Letra grega Alfa
β	Letra grega Beta
$\sum$	Somatório
$\in$	Pertence

SUMÁRIO

INTRODUÇÃO

Tem-se observado nos últimos anos com o avanço tecnológico e com o aumento da concorrência, que é crescente a busca de muitas empresas por consultorias técnicas ou métodos que envolvam processos de otimização, visando um melhor planejamento da linha de produção/distribuição, evitando perdas de material desnecessários, otimizando assim, os custos de produção. É comum a algumas destas empresas, utilizar os modelos de Programação Linear a fim de simular situações advindas da realidade que possam melhor auxiliar nas tomadas de decisão.

Uma característica importante que devemos estudar e que facilita muito processo de análise de decisão é a utilização de modelos, pois o modelo que adotamos, trabalha com a construção de um sistema real como meio de analisar e compreender o comportamento de situações multivariadas, com o objetivo de levá-lo a apresentar o desempenho que se deseja.

Para construirmos um Modelo Matemático que bem represente a realidade produtiva da empresa, é necessário observar com cuidado sua cadeia produtiva, a fim de identificar os custos fixos e os custos variáveis, identificando assim as variáveis de decisão do modelo.

Muitos modelos na prática podem ser formulados utilizando a programação linear e programação linear inteira. Iremos no decorrer do trabalho, apresentar estes modelos e os métodos de resolução, afim de se obter soluções ótimas.

1 UM POUCO DE HISTÓRIA

Desde o início de sua existência, o homem tem buscado melhor compreender o mundo onde vive. A primeira grande necessidade do homem foi o domínio do seu habitat. A segunda foi a busca da alimentação e estocagem, seguindo da segurança da sua sociedade em frente aos predadores e a organização de seus grupos sociais. À medida que as necessidades do homem foram se tornado mais complexas, também cresceu a necessidade de técnicas de compreensão do mundo.

Como tantos outros ramos da matemática, a otimização teve sua origem da necessidade do homem em se aprimorar. Durante vários anos, muitos estudiosos Propuseram modelos matemáticos para descrever e observar os acontecimentos do mundo.

No começo da década de 40, apareceram resultados significativos e relevantes para o desenvolvimento da programação linear. Este desenvolvimento se originou por conta da revolução industrial da época, assim, houve uma necessidade de aprimoramento das técnicas e dos modelos para resolução dos problemas pertinentes deste período.

Em 1947 (DANTZIG, 1963) propôs o método simplex, um método para resolução do problema de programação linear, baseando na teoria matricial e em técnicas de melhoramento da solução. O método simplex, possibilitou a resolução de problemas de otimização que envolve um grande número de variáveis, como os problemas de transporte, problemas de produção, de distribuição, entre outros.

2 PROGRAMAÇÃO LINEAR

Os modelo de Programação Linear se compõem em um tipo de modelo de otimização que utiliza a função função-objetivo, função que devemos maximizar ou minimizar, munido de restrições que deve ser respeitas para encontrar a solução ótima. Os modelos de Programação Linear(PL) contém uma característica especial de modelos de otimização. Para que um modelo possar ser representado por meio de uma PL, ele deve possuir algumas características. Vamos utilizar as características abordadas por (GRIFFIN, 2014), essas características são:

- **Suposição de Proporcionalidade:** Um problema pode ser expresso como um PPL, apenas se a quantidade de recursos para a função-objetivo e o lado esquerdo de cada restrição por cada variável de decisão $(x_1, \cdots, x_n)$ é proporcional ao valor da varável de decisão. Essa suposição nos informa que a quantidade de recurso consumido por uma atividade, deve ser proporcional ao nível de operação da atividade.

- **Suposição de Aditividade:** Um problema pode ser formulado como um problema de PPL, se a contribuição para a função-objetivo e o lado esquerdo de cada restrição por qualquer variável de decisão $x_i (i = 1, \cdots, n)$ é completamente independente de qualquer outra variável de decisão $x_j (j \neq i)$ e aditivo devendo ser sempre possível prosseguir com atividade em qualquer nível não negativo. Qualquer proporção de um dado recurso, deve sempre poder ser utilizado. A suposição informa que o custo total, seja ele de material ou tempo, deve ser a soma das parcelas associadas a cada atividade executada.

- **Suposição de Divisibilidade:** Um problema pode ser formulado como um PPL apenas se as quantidades representadas por cada variável de decisão forem infinitamente divisíveis, isto é, as respostas fracionadas fazem sentido. Essa suposição informa que pode-se identificar de forma separada, o custo especifico das operações realizadas por cada atividade.

- **Suposição de Certeza:** Um problema pode ser formulado como um PPL apenas se os coeficientes na função-objetivo e suas restrições forem conhecido com certeza. Essa suposição nos informa que sempre será possível desenvolver, dada atividade em qualquer nível não negativo e em qualquer proporção de um dado material ou recurso deve sempre poder ser utilizado.

As duas primeiras características simplesmente afirmam que tanto a função-objetivo, quanto as restrições são funções lineares de variáveis $x_1, \cdots, x_n$. A terceira característica afirma que uma resposta ótima válida, poderia conter valores fracionários para as variáveis de decisão. A ultima característica afirma que os coeficientes são conhecidos com certeza absoluta.

Um modelo de Programação Linear, é um modelo matemático de otimização em que todas as suas função são lineares da varável contínua x.

2.1 FORMULAÇÃO ALGÉBRICA GERAL

De acordo com (GARCIA; GUERREIRO; CORRAR, 1997), o modelo matemático de um PPL pode ser formulado como seguinte:

Maximizar ou Minimizar

$$z(x_1,\cdots,x_n) = c_1x_1 + c_1x_2 + \cdots + c_nx_n \tag{2.1}$$

Sujeito a:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots a_{mn}x_n &\leq b_m \end{aligned} \tag{2.2}$$

Onde $x_i \geq 0$ e $b_j \geq 0$, para $i = 1,2,\cdots,n$ e $j = 1,2,\cdots,m$

A equação 2.1 é a função matemática que exemplifica o objetivo do problema e é denominada função-objetivo.

A equação 2.2 inequações matemáticas que são as restrições do problemas.

z : é a função que queremos otimizar, maximizar ou minimizar, respeitando o conjunto de restrições;

x_i : são as variáveis de decisão que representam as quantidades ou recursos que temos disponíveis;

a_{ij} : é a quantidade de recursos que cada variável decisória consome;

c_i : são os coeficientes de ganho ou custo que cada variável é capaz de gerar;

b_j : é a quantidade disponível de cada recursos;

Os Problemas de Programação linear são geralmente expressos algebricamente. A resolução dos PPL, na maioria das vezes, é feita através do método Simplex, utilizando o software (LINGO,) para realizar os cálculos.

Definição 2.1. (Função Linear) A função $z : \mathbb{R}^n \to \mathbb{R}$ é linear se há constantes $c_1,\cdots,c_n \in \mathbb{R}$ de modo que:

$$z(x_1,\cdots,x_n) = c_1x_1 + c_1x_2 + \cdots + c_nx_n$$

Lema 2.1. (Função Linear) Se $z : \mathbb{R}^n \to \mathbb{R}$ é linear, então para todo $x_1, x_2 \in \mathbb{R}^n$ e para todo constante escalar $\alpha \in \mathbb{R}$ temos:

$z(x_1 + x_2) = z(x_1) + z(x_2)$
$z(\alpha x_1) = \alpha(x_1)$

Uma observação que devemos ficar atentos e que já foi citada anteriormente, um mesmo PLL composto pelo conjunto de equações apresentadas, pode operar de forma que, mudando o seus critério de otimização podemos transformar um problema de maximização para minimização e vice-versa. Essa mudança pode ser realizada através da seguinte propriedade:

- Maximizar $f(x)$ corresponde a Minimizar $(-f(x))$ e
- Minimizar $f(x)$ corresponde a Maximizar $(-f(x))$

2.1.0.1 Formulações Equivalentes

Podemos formular os nossos PLL de outras formas distintas, visando em uma forma mais simples de visualização:

Forma Padrão

Maximizar ou Minimizar

$$z = cx$$

sujeito a:

$$Ax = b$$

$$x \geq 0$$

Sendo que A é a matriz da quantidade de recursos disponíveis, x será a matriz que possui as variáveis de decisão e b será a matriz com a quantidade de recursos disponíveis.

Forma Canônica

Maximizar ou Minimizar

$$z = cx$$

sujeito a:

$$Ax \leq b$$

ou

$$Ax \geq b$$

$$x \geq 0$$

3 MODELAGEM DO PROBLEMA

3.1 O MÉTODO DE MODELAGEM

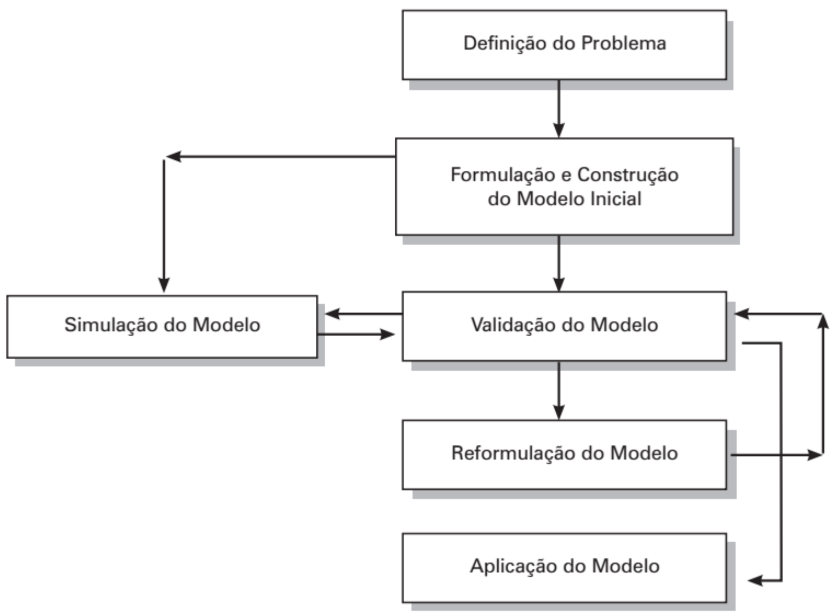
Iremos definir alguns mecanismos usados para obtermos o modelo. Esse processo é de total importância para conseguir os resultados mais precisos e fieis do modelo.

O processo de modelagem inicialmente pode ser desafiador, principalmente quando o problema a ser resolvido envolve uma grande quantidade de variáveis e restrições. A modelagem é uma etapa importante para a resolução de um problema de otimização. A descrição do problema deve ser clara e objetiva a fim de permitir que o modelo retrate suas características principais. Estas características são:

- Variáveis de decisão
- Níveis de detalhe
- Objetivos a serem alcançados

O processo de modelagem deve ser construído como no fluxograma 1:

Figura 1 – Fluxograma para Modelagem



Fonte – (GOLDBARG; LUNA, 2005)

Para construímos um ótimo modelo, devemos ter algumas aptidões construídas ao longo do processo de formação. Tais como, criatividade, intuição, dados fieis e principalmente experiência.

3.2 MODELAGEM MATEMÁTICA

Após criarmos um modelo de otimização de forma geral, podemos formular o modelo matemático, utilizando técnicas e algoritmos para obter os resultados ótimos. Vamos analisar alguns modelos matemáticos e quais modelos devemos usar para certos tipos de situações.

3.2.1 Tipos de Modelos

A modelagem possui varias ramificações na matemática. Muitas vezes, os modelos demandam técnicas e algoritmos diferentes para obtermos as soluções. Modelar e resolver corretamente os modelos de otimização, é essencial para conseguirmos os resultados mais precisos e rápidos.
Segundo (BASSANEZI, 2002) *O Modelo Matemático é um conjunto de símbolos e relações matemáticas que representam de alguma forma o objeto estudado*. Bassanezi formula alguns modelos de acordo com a natureza dos fenômenos ou situações analisadas e classificadas conforme o tipo de matemática utilizada, esses modelos são:

i. *Linear* ou *não-linear*, essa classificação depende de suas variáveis, função objetivo e restrições, se terão características lineares ou não.

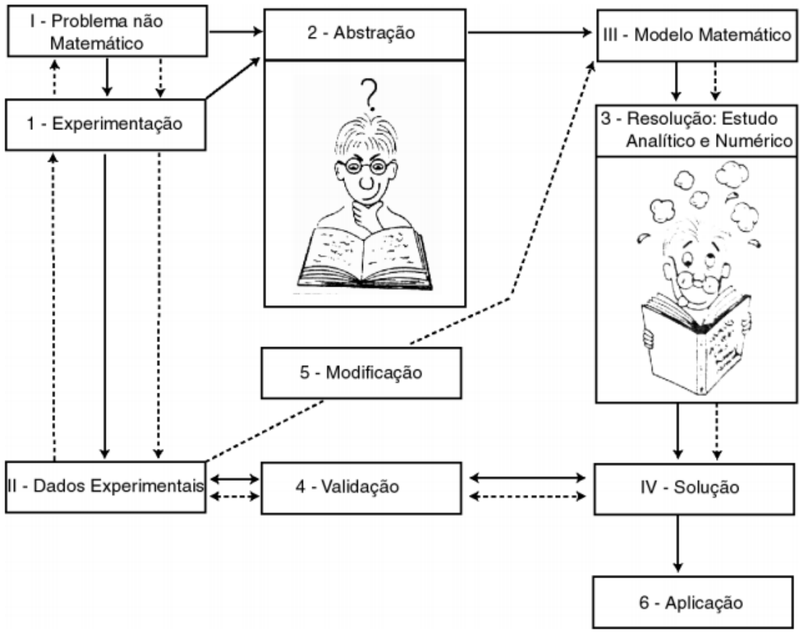
ii. *Estático*, quando representa a forma do objeto, por exemplo, um crescimento proporcional de uma colmeia, ou o crescimento logaritmo de uma Vitória Regia.

iii. *Educacional*, quando sua fonte é baseada em um numero pequenos ou simples de hipóteses, quase sempre encontrando soluções analíticas

(BASSANEZI, 2002) destaca os procedimentos envolvidos na modelagem matemática de um problema ou sistema:

(1) *Experimentação: É a atividade que faz a coleta de informações e dados.* Observamos na figura 2 um esquema de uma modelagem.

Figura 2 – Esquema de uma modelagem: as setas contínuas indicam a primeira aproximação. A busca de um modelo matemático que melhor descreva o problema estudado torna o processo dinâmico, indicado pelas setas pontilhadas.



Fonte – (BASSANEZI, 2002)

(2) *Abstração:* É a atividade que deve levar à formulação dos modelos matemáticos. Nesta fase deve-se observar os seguintes passos:

i Seleção das variáveis;

ii Problematização dos problemas teóricos numa linguagem própria da área em que se está trabalhando;

iii Formulação de hipóteses;

iiii Simplificação- *"Se você não consegue resolver problema a que se propôs, então tente simplificá-lo. A condição única é esta: você não deve simplificá-lo demasiadamente a ponto de perder as informações essenciais"(Mark Kac - matemático polonês - 1914-1983)*;

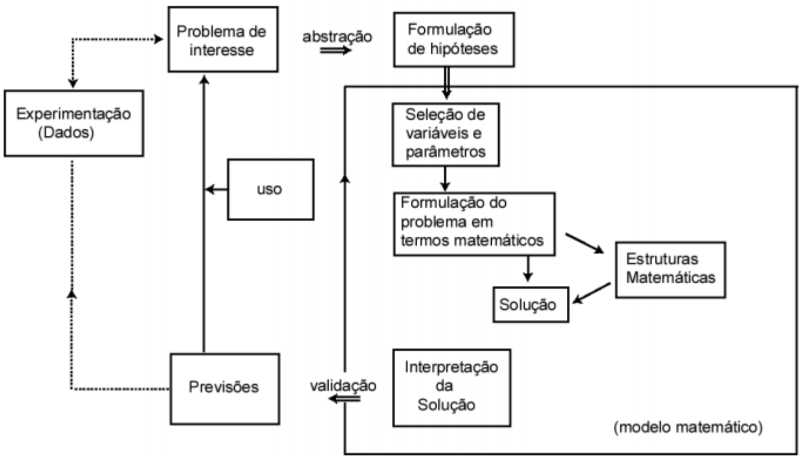
(3) *Resolução:* Nesta etapa usam os resultados e técnicas para resolver o modelo;

(4) *Validação:* É o processo de aceitação ou não do modelo proposto. Nesta etapa devemos comparar se os resultados e previsões com os valores obtidos servem para o problema ou sistema real;

(5) *Modificação:* Caso o nosso modelo não seja valido, devemos fazer alterações nas etapas anteriores. Nenhum modelo deve ser considerado definitivo, podendo sempre ser melhorado e modificado.

No esquema da figura 3, podemos ter uma ideia desta divisão de atividades intelectuais:

Figura 3 – Divisão de atividades intelectuais



Fonte – (BASSANEZI, 2002)

As ferramentas computacionais possui grande relevância para resolver os problemas de otimização que iremos abordar, pois, utilizando softwares específicos, podemos resolver diferentes tipos de problemas otimização combinatória. Os problemas de otimização combinatória estão divididos em subáreas que são: Otimização Linear, Otimização Discreta e Otimização Não linear. Para o presente trabalho será abordado os dois primeiros tipos.

3.2.2 Otimização Combinatória

Nos modelos de otimização linear as suas variáveis são continuas, ou seja, sua função-objetivo e suas restrições (equações e/ou inequações) são lineares e suas variáveis reais não-negativas. Iremos mais adiante estudar alguns problemas clássicos de Otimização Linear, como o problema da ração, problema da dieta, problema do sítio e também vamos abordar como se resolve esses problemas de forma geométrica e computacional.

Para (ARENALES et al., 2011), oque caracteriza a otimização discreta ou combinatória, é que algumas das variáveis pertencem a um conjunto discreto, tipicamente um subconjunto dos números inteiros. Otimização discreta também é conhecida como programação inteira. Alguns problemas envolvendo tais variáveis são os problemas de corte de estoque, problema da mochila, problemas de roteamento de veículos, entre outros.

Os problemas de Otimização não linear, segundo (GOLDBARG; LUNA, 2005), são bem tratáveis nos casos de convexidade, situação em que preservam propriedades importantes tanto sob ótica da Programação Matemática como da teoria econômica. Esses modelos de otimização não serão abordados neste trabalho, pois requer um estudo mais aprofundado e ferramentas mais complexas.

Vamos estudar o seguinte problema proposto por (NETO, 2009).

João pretende vender latinhas de refrigerantes, água e cerveja nos semáforos de São Paulo. Ele possui uma caixa de isopor que suporta $15kg$. Supondo que ele venda tudo que leve na caixa, com base na tabela 1, quantas garrafas de água, latas de cerveja e de refrigerantes ele deve levar para obter o lucro máximo?

Tabela 1 – Problema do Semáforo

	Cerveja	Refrigerante	Água
Peso (gramas)	400	390	550
Lucro (R$)	0,50	0,40	0,60

Fonte – Própria

Esse exemplo é um modelo clássico de problema da mochila que iremos estudar, mas então como devemos resolvê-lo? Ou melhor, como devemos modelá-lo, em uma linguagem mais teórica, na linguagem matemática?

Nesta fase, devemos observar e identificar os dados, as suas restrições e o objetivo, neste exemplo, o objetivo é obter o lucro máximo. Para alcançá-lo, devemos em particular, apresentar uma quantidade de garrafas de água, latas de cerveja e de refrigerante que João deverá levar para o semáforo, com o **objetivo** de obter o lucro máximo. Para modelamos esses objetivos, fica inviável a todo momento escrever o número de latas de cerveja, de garrafas de água ou de refrigerante, para sermos mais eficientes vamos nomear de forma de um forma simplificada, x_1 a quantidade de latas de cerveja, x_2 a quantidade de latas de refrigerante e x_3 a quantidade de garrafas de água.

Temos um objetivo, João obter o lucro máximo. Esse lucro depende da quantidade de latas de cerveja (x_1), latas de refrigerante (x_2) e da quantidade de garrafas de água (x_3) que ele vai levar. Vamos supor que esteja um dia muito favorável e que ele venda tudo. Observando atentamente, não fica difícil perceber que o lucro é uma **função** de x_1, x_2, x_3. Falamos da função, mas qual seria essa função? Vamos supor que João venda duas garrafas de água, então ele terá o lucro de $2 \times 0,60 = R\$1,20$; se ele vende mais três latas de cerveja obtém o lucro de $2 \times 0,60 + 3 \times 0,50 = R\$1,20 + R\$1,50 = R\$2,70$; se ele vende mais 5 latas de refrigerante consegue o lucro de $2 \times 0,60 + 3 \times 0,50 + 5 \times 0,40 = R\$1,20 + R\$1,50 + R\$2,0 = R\$4,70$. Então podemos escrever a função lucro L, da seguinte forma:

$$L(x_1, x_2, x_3) = 0,50 \times x_1 + 0,40 \times x_2 + 0,60 \times x_3$$

simplificando ficará $L = 0,50 \times x_1 + 0,40 \times x_2 + 0,60 \times x_3$

Chamaremos essa função de função-objetivo, pois ajudará a obter os resultados. Agora que já sabemos a função-objetivo e o nosso objetivos como devemos prosseguir? Lendo mais atentamente observamos que o problema tem restrições. Observe que a caixa de isopor é limitada no tamanho e na quantidade de peso que comporta, ou seja, a caixa tem a capacidade de armazenar $15kg$.

Precisamos traduzir essa informação para o nosso modelo. Para isso vamos repetir o mesmo raciocínio usado anteriormente na construção da função-objetivo. Se João só colocar garrafas de água, até a capacidade máxima do isopor, então ele levará 27 garrafas e se ele vender todas as garrafas de água, ele obterá um lucro de $27 \times 0,60 = R\$16,20$. Entretanto, se João levar somente latas de cerveja em vez de água, João levará 37 latas de cerveja e supondo que ele venda todas, ele obterá o lucro de $37 \times 0,50 = R\$18,50$. Observe que o lucro foi maior se comparamos com a quantidade de água.

Agora se joão colocar 10 garrafas de água, estará levando $10 \times 550g = 5,50kg$. Se colocar mais 6 latas de cerveja levará $10 \times 550g + 6 \times 400g = 5,50 + 2,4 = 7,9kg$. Desta forma, temos que o peso levado em quilogramas é, $0,40 \times x_1 + 0,39 \times x_2 + 0,55 \times x_3$. Essa soma deve ser menor ou igual $15Kg$. Portanto, temos a seguinte desigualdade:

$$0,40 \times x_1 + 0,39 \times x_2 + 0,55 \times x_3 \leq 15$$

Vale ressaltar que tanto a função-objetivo quanto a desigualdade acima são lineares. Observe que não podemos levar meia lata de cerveja, ou de água, deste modo, falta definimos o conjunto de valores que x_1, x_2, x_3 podem assumir. De fato, não faria sentido temos $x_2 = -3$, por exemplo, ou $\sqrt{2}$ de refrigerante. Logo, precisamos que nossas variáveis sejam inteiras e não-negativas. $x_i \geq 0, x_i \in \mathbb{Z}, i = 1,2,3$.

Voltando ao início do problema, queremos maximizar o lucro, ou seja, obter o maior lucro possível, assim o modelo matemático para o Problema do Semáforo é o seguinte:

$$\begin{cases} Maximizar & L = 0,50 \times x_1 + 0,40 \times x_2 + 0,60 \times x_3 \\ sujeito\ a: & 0,40 \times x_1 + 0,39 \times x_2 + 0,55 \times x_3 \leq 15 \\ & x_i \geq 0, x_i \in \mathbb{Z}, i = 1,2,3 \end{cases} \tag{3.1}$$

Temos aqui um Problema da Mochila limitada ou restrita, sendo um problema com aplicações práticas interessantes. Mas adiante vamos apresentar o método de resolução para este problema.

3.3 MÉTODO SIMPLEX

Problemas que envolvem duas ou três variáveis, podem ser resolvidos de forma gráfica. Para problemas maiores, o método gráfico se torna impraticável por conter muitas restrições, ficando praticamente impossível encontrar a solução ótima. Neste caso, precisamos de uma técnica mais eficiente para resolvê-los, portanto, utilizamos o *método simplex*.

Este método nada mais é do que um algoritmo de busca. Este método é considerado como uma das grandes contribuições para a Otimização Matemática deste século. Desenvolvido por (DANTZIG, 1963), e após sua descoberta, houve um crescimento muito grande na Otimização Linear. Antes de 1947 a Otimização Linear não era muito aplicada.

A seguir apresentamos um exemplo onde se aplica o método Simplex.

3.3.1 Um exemplo numérico

Vamos resolver o seguinte problema de programação linear passo a passo, proposto por (SILVA, 2006):

$$\begin{cases} Maximizar & z = 6x_1 + 8x_2 \\ sujeito\ a: & \\ & 30x_1 + 20x_2 \leq 300 \\ & 5x_1 + 10x_2 \leq 110 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases} \tag{3.2}$$

1. Transformar o modelo para o formato padrão, ou seja, passar todos os elementos da função objetiva para o outro lado da equação, igualando a zero, transformar as desigualdades em igualdades e acrescentar as variáveis de folga.

$$\begin{cases} Maximizar & z - 6x_1 - 8x_2 + 0F_1 + 0F_2 \quad = 0 \\ sujeito\ a: & \\ & 30x_1 + 20x_2 + 0F_1 \quad = 300 \\ & 5x_1 + 10x_2 \quad + 0F_2 = 110 \\ & x_1 \geq 0, x_2 \geq 0, F_1 \geq 0, F_2 \geq 0 \end{cases} \tag{3.3}$$

2.Preencher a tabela SIMPLEX, com as informações obtidas no modelo.

Tabela 2 – Tabela Inicial Simplex

VB	x_1	x_2	F_1	F_2	VTI
z	-6	-8	0	0	0
F_1	30	20	1	0	300
F_2	5	10	0	1	110

Fonte – Própria

- VB= Variáveis Básicas

- VTI= Valor dos termos independentes
- z= Função objetiva
- F_i=Variáveis de folga

Observe que na linha z, colocamos os valores respectivos da função objetivo e nas outras respectivamente, colocamos os valores das restrições e suas variáveis de folga.

3. Determinação da variável que sai da base (determinação do pivô).

Para a maximização da função, devemos observar algumas etapas que serão descritas a seguir, a partir dessas etapas, conseguiremos determinar a variável VB, a nossa base e o critério de parada para determinar a solução ótima.

Algoritmo:

1. Se na equação da função (quadro Simplex) há variáveis com coeficientes negativo, então a solução corrente não maximiza z, ou seja, não é ótima.
2. A variável que na linha de z tem coeficiente negativo com maior valor absoluto, deve ser selecionada para entrada na nova base. Se houver empate decidi-se arbitrariamente.
3. Se na equação da função(quadro Simplex) todas as variáveis da forma padrão têm coeficientes não negativos, então a solução corrente é ótima.

Observamos o quadro inicial, notamos que a solução não é a ideal, pois, na equação da função, existem coeficientes negativos. Percebemos que a coluna x_2 possui o maior valor absoluto, portanto, essa coluna será nossa coluna pivô.

Tabela 3 – Tabela Simplex Parte 1

VB	x_1	x_2	F_1	F_2	VTI
z	-6	-8	0	0	0
F_1	30	20	1	0	300
F_2	5	10	0	1	110

Fonte – Própria

4. Escolha da VB que deve sair da base

Tabela 4 – Tabela Simplex Parte 2

VB	x_1	x_2	F_1	F_2	VTI
z	-6	**-8**	0	0	0
F_1	30	20	1	0	300
F_2	5	10	0	1	110

Fonte – Própria

Já definimos que x_2 é a nova VB. Nestas circunstâncias a variável x_1, não está na base. Vamos analisar a seguinte situação para definimos o termo pivô da coluna pivô. Observe no sistema abaixo:

$$\begin{cases} 20x_2 + 1F_1 & = 300 \\ 10x_2 \qquad + 1F_2 & = 110 \end{cases} \tag{3.4}$$

Qual deverá ser o valor de x_2 para fazemos a mudança de base, escolhendo F_1 ou F_2 para sair da base?

Na primeira equação, para anular F_1 é necessário que $x_2 = \frac{300}{20} = 15$.

Na segunda equação, para anular F_2 é necessário que $x_2 = \frac{110}{10} = 11$.

Se escolhermos $x_2 = 15$, sai F_1 da base; se escolhemos $x_2 = 11$, sai F_2 da base.
Pare decidir devemos olhar para a admissibilidade da nova solução, ou seja, todos os valores do sistema com valores não negativos. Se substituímos $x_2 = 15$ no sistema de equações, fica com $F_1 = 0$ mas com F_2 com valor negativo que não é admissível. Note que se resolvemos o sistema com $x_2 = 15$ teremos $F_1 = 0$ e $F_2 = -40$. Agora se escolhemos $x_2 = 11$ teremos que, $F_1 = 80$ e $F_2 = 0$, que é admissível.
Analisando estes quocientes, podemos determinar qual será o elemento pivô na nossa base, nesse caso, será o elemento 10. Para determinar o elemento pivô devemos analisar o mínimo de

$$\delta = mim \left\{ \frac{VTI_{F_1}}{F_1}, \frac{VTI_{F_2}}{F_2} \right\} = \left\{ \frac{300}{20}, \frac{110}{10} \right\} = 11 = x_2$$

Como o mínimo é 11 e esse valor corresponde a linha F_2, nosso elemento pivô na tabela Simplex, será **10**. Para selecionar a variável que deve sair da base, temos que utilizar o seguinte Algoritmo:

1. Dividem-se os termos independentes da tabela Simplex VTI, pelas componentes não negativas do vetor da variável selecionada para VB.

2. Sai da base a VB associada à equação onde se obtém o mínimo, entre a divisão efetuada, que deve ser seguramente não negativo, pois é o valor da nova VB.

3. Em caso de empate, a decisão pe arbitrária.

Tabela 5 – Tabela Simplex Parte 3

VB	x_1	x_2	F_1	F_2	VTI
z	-6	**-8**	0	0	0
F_1	30	20	1	0	300
F_2	5	10	0	1	110

Fonte – Própria

Agora determinamos o elemento pivô e sua linha pivô. Prosseguimos fazendo o seguinte passo.

5. Dividimos a linha pivô pelo elemento pivô

Tabela 6 – Tabela Simplex Parte 4

VB	x_1	x_2	F_1	F_2	VTI
z	-6	-8	0	0	0
F_1	30	20	1	0	300
x_2	5/10=1/2	$10/10 = 1$	$0/10 = 0$	$1/10 = 1/10$	$110/10 = 11$

Fonte – Própria

Nossa nova tabela ficará assim:

Tabela 7 – Tabela Simplex Parte 5

VB	x_1	x_2	F_1	F_2	VTI
z	-6	-8	0	0	0
F_1	30	20	1	0	300
x_2	$1/2$	1	0	$1/10$	11

Fonte – Própria

Assim, determinamos a nossa nova tabela e iremos prosseguir para o próximo passo.

6. Substituímos as outras linhas fazendo o seguinte procedimento: Linha antiga - (Coeficiente da Coluna pivô) x (Nova linha pivô).

Tabela 8 – Tabela Simplex Parte 6.1

VB	x_1	x_2	F_1	F_2	VTI
z	$-(-8)(1/2)-6$ $=-2$	$-(-8)(1)-8$ $=0$	$-(-8)(0)+0$ $=0$	$-(-8)(1/10)+0$ $=4/5$	$-(-8)(11)+0$ $=88$

Fonte – Própria

Tabela com a função objetiva já calculada.

Tabela 9 – Tabela Simplex Parte 6.2

VB	x_1	x_2	F_1	F_2	VTI
z	-2	0	0	4/5	88
F_1	$-(20)(1/2)+30$ $=20$	$-(20)(1)+20$ $=0$	$-(20)(0)+1$ $=1$	$-(20)(1/10)+0$ $=-2$	$-(20)(11)+300$ $=80$

Fonte – Própria

Nova tabela Simplex com os novos coeficientes calculados.

Tabela 10 – Tabela Simplex Parte 6.3

VB	x_1	x_2	F_1	F_2	VTI
z	-2	0	0	4/5	88
F_1	20	0	1	-2	80
x_2	1/2	1	0	1/10	11

Fonte – Própria

Observando a nova tabela Simplex, tem-se uma nova solução, $x_2 = 11; F_1 = 80; z = 88$.

Esta solução não é ótima, pois na equação da função o coeficiente de x_1 é negativo, portanto, ainda não podemos utilizar o critério de parada. Como x_1 é o único número negativo, conclui-se que ele irá entrar para a base, pois entre todos, ele tem o maior valor absoluto e voltamos para o passo 3. Agora que voltamos para o passo 3, devemos decidir qual das atuais VB, $F_1 = 80$ e $x_2 = 11$ deve sair da base.

Vamos calcular novamente o mínimo entre a divisão de VTI, com as variáveis VB.

$$\delta = mim\left\{\frac{VTI_{F_1}}{F_1}, \frac{VTI_{x_2}}{x_2}\right\} = \left\{\frac{80}{20}, \frac{11}{1/2}\right\} = 4 = F_1$$

Temos que o menor valor não negativo é 4, portanto, F_1 sai da base. Na nova base ficarão as variáveis de x_1 e x_2 que irão permanecer na base.

Trocando as VB, teremos:

Tabela 11 – Tabela Simplex Parte 7.1

VB	x_1	x_2	F_1	F_2	VTI
z	-2	0	0	4/5	88
x_1	$20/20=1$	$0/20=0$	$1/20=1/20$	$-2/20=-1/10$	$80/20=4$
x_2	1/2	1	0	1/10	11

Fonte – Própria

Essa será a tabela com a nova VB.

Tabela 12 – Tabela Simplex Parte 7.2

VB	x_1	x_2	F_1	F_2	VTI
z	-2	0	0	4/5	88
x_1	1	0	$1/20$	$-1/10$	4
x_2	1/2	1	0	1/10	11

Fonte – Própria

Fazendo os cálculos repetidamente nas outras linhas teremos:

Etapa 1.

Tabela 13 – Tabela Simplex Parte 8.1

VB	x_1	x_2	F_1	F_2	VTI
z	$2(1)-2$ $=0$	$2(0)+0$ $=0$	$2(1/20)+0$ $=1/10$	$2(-1/10)+4/5$ $=3/5$	$2(4)+88$ $=96$
x_1	1	0	$1/20$	$-1/10$	4

Fonte – Própria

Etapa 2.

Tabela 14 – Tabela Simplex Parte 8.2

VB	x_1	x_2	F_1	F_2	VTI
z	0	0	$1/10$	$3/5$	96
x_1	1	0	$1/20$	$-1/10$	4
x_2	$-1/2(1)+1/2$ =0	$-1/2(0)+1$ =1	$-1/2(1/20)+0$ =-1/40	$-1/2(-1/10)+1/10$ =3/20	$-1/2(4)+11$ =9

Fonte – Própria

Essa será nossa nova tabela Simplex

Tabela 15 – Tabela Simplex Parte 8.3

VB	x_1	x_2	F_1	F_2	VTI
z	0	0	$1/10$	$3/5$	96
x_1	1	0	$1/20$	$-1/10$	4
x_2	0	1	-1/40	3/10	9

Fonte – Própria

Analisando a nova tabela Simplex, notamos que essa solução é ótima, pois se aplicamos o critério de parada notamos que na equação da função objetiva não temos variáveis negativas, portanto, encontramos a solução ideal.

Temos que a solução ideal descrita a seguir:

$$S = \begin{bmatrix} x_1 \\ x_2 \\ F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 9 \\ 0 \\ 0 \end{bmatrix} ; Max\ z = 96$$

Desse modo, finalizamos o nosso exemplo numérico, que serve como um auxilio para entendemos o algoritmo. Alguns outros exemplos de programação linear serão resolvidos utilizando o Software "LINGO".

3.3.2 O Algoritmo simplificado do Método Simplex

Primeiramente devemos observar se o problema é Maximização ou de Minimização. Se for um problema de Maximização, então o algoritmo que usamos para solucionar o problema numérico acima está valido, caso a função seja um problema de Minimização, podemos rescrevê-lo como um problema de Maximização multiplicando por (-1), desse modo, podemos calcular normalmente utilizando o algoritmo. Por se tratar de um problema em que as variáveis de decisão são reais, uma solução para o problema pode ser mais rapidamente determinada. O algoritmo resumido do método Simplex proposto por (SILVA, 2006), se comporta desta maneira:

Algoritmo:

1. **Solução Inicial**

   i. Escolher para VB as variáveis auxiliares cujos vetores constituem uma matriz identidade.

2. **Escolher a Nova VB**

   i. Identificar as varáveis não básicas que na equação da função têm coeficiente negativo.

   ii. Destas variáveis, selecionar para entrada na base aquela cujo coeficiente tem o maior valor absoluto.

   iii. Se houver mais do que uma variável nas mesmas condições, escolha arbitrariamente.

   iv. Se todas as variáveis têm coeficiente não negativo, então a solução corrente é ótima

3. **Escolher a Nova Variável não Básica VNB**

   i. Nas equações técnicas, calcular as divisões entre os valores da solução corrente (Coluna VTI) e os coeficientes não negativos da nova VB(Ignorar os resultados com denominador nulo, ou negativo).

   ii. A VB corrente em cuja linha se verifica o menor destes resultados, é aquela que deve sair da base.

iii. Se este mínimo se verifica em mais do que umas das VB correntes, escolher arbitrariamente umas delas para nova VNB.

4. **Calcular o nono quadro Simplex**

   i. Efetuar transformações lineares para que a linha dos coeficientes da nova VB juntamente com os das VB que permanecem na base constituída uma matriz identidade.

   ii. Todas as VB devem ter coeficientes nulo na equação da função.

5. **Voltar ao passo 2**

A partir deste algoritmo, podemos resolver problemas de otimização linear com muito mais praticidade e velocidade. Desta maneira, gastamos menos tempo na resolução dos problemas de otimização e possibilita resultados mais precisos e confiáveis. Um algoritmo mais refinado pode ser encontrado em (BAZARRA; JARVIS; SHERALI, 2009).

3.3.2.1 Aspectos práticos na construção do quadro Simplex

Alguns aspectos propostos por (SILVA, 2006), nos auxilia no momento de realizamos os cálculos da tabela Simplex, esses aspectos podem ser vistos como uma consequência dos cálculos que efetuamos, mas que não devem ser ignorados, pois, facilita na hora de fazemos diversas contas. Esses aspectos são:

- No cálculo de um novo quadro, os valores nas equações técnicas da nova VB e das VB que permanecem na base, devem constituir uma matriz identidade, como já vimos no exemplo numérico. Deste modo, não há necessidade de calculamos esses valores.

- No cálculo de um novo quadro, os coeficientes na equação da função da nova VB e das VB que continuam na base, devem ser nulos, podendo ser registrados sem a necessidade de cálculo.

- Se a nova VB tem coeficiente nulo numa equação técnica, esta equação não sofre alteração, podendo ser registrada no novo quadro sem necessidade de cálculo.

3.3.3 Definições e conceitos

A **Região viável** de um Problema de Programação Linear, é o conjunto de todos os pontos $\alpha_i, \beta_i, z_i \in \mathbb{R}$ que satisfazem todas as restrições. Cada restrição do tipo $\alpha_1 x + \beta_1 y = z_1$ define uma reta no plano cartesiano $xy \in \mathbb{R}$. Toda restrição da forma $\alpha_2 x + \beta_2 \leq z_2$ ou $\alpha_2 x + \beta_2 \geq z_2$, define um semiplano que inclui a reta de limite $\alpha_2 x + \beta_2 = z_2$. Logo, a região viável é sempre uma intersecção de um número finito de retas e semiplanos.

Em um problema de maximização uma **solução ótima** de um PPL, é um ponto extremo na região viável, que possui o maior valor na função-objetivo. Em problemas de minimização, é um ponto extremo na região viável, em que, possua o menor valor possível na função-objetivo.

Uma região viável de um problema de otimização, defini um conjunto convexo. Um subconjunto $X \in \mathbb{R}^n$ é dito *convexo* se e somente se, para todos os vetores $\overrightarrow{a}$ e $\overrightarrow{b}$ de X, qualquer combinação linear $X = \alpha\overrightarrow{a} + (1-\alpha)B\overrightarrow{b}$ é um elemento de X, onde $0 \leq \alpha \leq 1$. Se a região é vazia o PPL não possui solução.

Os pontos de extremos de uma região viável, são interseções de dois segmentos de retas de fronteira.

Podemos agora enunciar alguns teoremas.

Teorema 3.1. O conjunto de todas as soluções viáveis de um modelo de PPL é um conjunto convexo.

Teorema 3.2. Toda solução viável básica de um PPL é um ponto extremo(vértice) do conjunto de soluções viáveis, isto é, do conjunto convexo X de soluções viáveis.

Teorema 3.3. Se a função objetivo possui um máximo ou mínimo finito, então pelo menos uma solução ótima é um ponto extremo do conjunto convexo X de soluções viáveis do PPL.

A demostração destes teoremas podem ser encontradas em (GOLDBARG; LUNA, 2005).

Em uma PPL acontece uma das seguintes situações:

(1) O PPL tem uma única solução ótima;

(2) O PPL tem múltiplas soluções ótimas;

(3) O PPL é inviável, ou seja, a região viável é vazia e não existe solução;

(4) O PPL é ilimitado, ou seja, existem pontos na região viável, cujo o valor da função-objetivo e praticamente ilimitado superiormente no cado do problema de maximização e ilimitado inferiormente nos problemas de minimização.

Um exemplo na primeira situação, é o problema do sitio que vimos anteriormente. As outras situações podem ser exemplificadas em (NETO, 2009).

4 PROGRAMAÇÃO LINEAR INTEIRA

4.1 CARACTERÍSTICAS DOS MODELOS LINEARES DE PROGRAMAÇÃO INTEIRA

Retirado de (GOLDBARG; LUNA, 2005). Uma confeitaria pode produzir dois tipos de sorvete em lata: chocolate e creme. Cada lata do sorvete de chocolate é vendido com um lucro de \$3 e as latas de creme com um lucro de \$1. Contratos com várias lojas impõem que sejam produzidas no mínimo 10 latas de sorvete de chocolate por dia e que o total de latas fabricadas por dia nunca seja menor que 20. O mercado só é capaz de consumir até 40 latas de sorvete de creme e 60 de chocolate. As máquinas de preparação do sorvete disponibilizam 180 horas de operação por dia, sendo que cada lata de sorvete de chocolate consome 2 horas de trabalho e cada lata de creme, 3 horas. Determine o esquema de produção que maximiza os lucros com a venda de latas de sorvete.

Modelando o PPI temos:

(1) **Variáveis de Decisão:** temos como variáveis de decisão as latas de sorvete, então denominamos x_1 para o sorvete de chocolate e x_2 para o sorvete de creme. $x_i; i = 1, 2$.

(2) **Função-objetivo:** Nosso objetivo é maximizar o lucro da confeitaria, portanto temos que maximizar: Max $l = 3x_1 + 1x_2$.

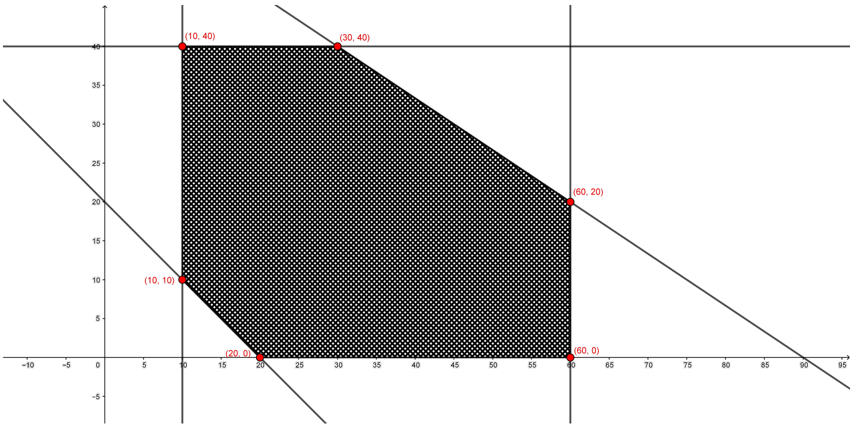
(3) **Restrições:** Notamos que no problema, o mercado não absorve mais do que 60 latas de sorvete de chocolate por dia, então nossa primeira restrição será: $x_1 \leq 60$. Seguindo o mesmo raciocínio para o sorvete de creme temos a nossa segunda restrição, que será: $x_2 \leq 40$. Nossas outras restrições se originam da produção mínima diária. Várias lojas impõem uma produção mínima diária de 10 latas de sorvete de chocolate e 20 latas de sorvete de creme, portanto nossas restrições serão: $x_1 \geq 10; x_1 + x_2 \geq 20$. A maquinas apenas disponibilizam 180 horas de operação, sendo que uma lata de sorvete de chocolate consome 2horas, a lata de creme 3horas, logo nossas restrições serão: $2x_1 + 3x_2 \leq 180$. A nossa ultima restrição será de não-negatividade e como as variáveis serão inteiras temos que: $x_1, x_2 \in \mathbb{Z}^+$

Nosso modelo será :

$$\begin{cases} Maximizar & l = 3x_1 + 1x_2 \\ sujeito\ a: & \\ & x_1 \leq 60 \\ & x_2 \leq 40 \\ & x_1 \geq 10 \\ & x_1 + x_2 \geq 20 \\ & 2x_1 + 3x_2 \leq 180 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{cases} \tag{4.1}$$

Sabemos pelo teorema 3.3 que a solução ótima de PPL, se existir, estará em um vértice da nossa região ótima.

Figura 4 – Região ótima PLI Sorvete



Fonte – Própria

Analisando a nossa tabela 20, podemos analisar o valores obtidos com seus respectivos vértices e suas soluções:

Tabela 16 – Possíveis valores ótimos

Coordenadas (x_1, x_2)	**Valor da função** $l = 3x_1 + x_2$
(10,10)	40
(10,40)	70
(30,40)	130
(60,20)	200
(60,0)	180
(20,0)	60

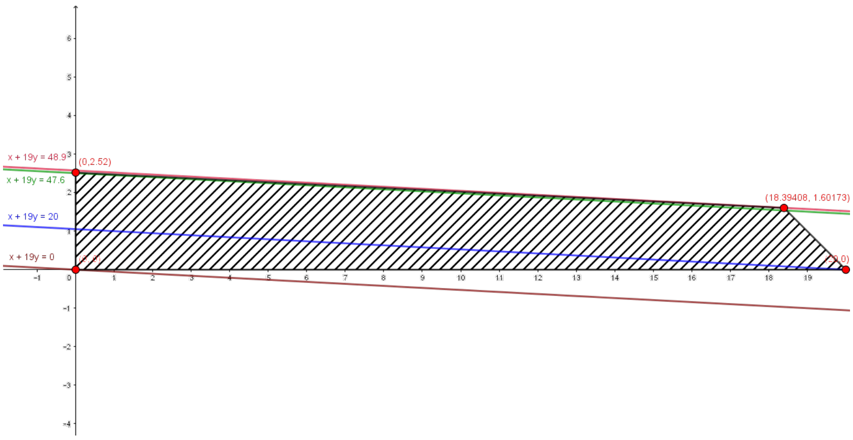
Fonte – Própria

Por sorte, temos que os vértices da nossa região ótima, se posiciona em coordenadas inteiras, por conta disso, encontramos a solução ótima, $(60,20)$, com o valor ótimo dado por $l = 200$, isto é, no esquema ótimo de produção, devem ser produzidas diariamente 60 latas de sorvete de chocolate e 20 de creme, com um retorno de \$200.

No exemplo dado, a solução ótima ocorreu em um ponto de coordenadas inteiras. No entanto, nem sempre isso acontece como mostra o exemplo a seguir:

$$\begin{cases} Maximizar \quad l = x_1 + 19x_2 \\ sujeito\ a: \\ \qquad\qquad x_1 + 20x_2 \leq 50 \\ \qquad\qquad x_1 + x_2 \leq 20 \\ \qquad\qquad x_1, x_2 \in \mathbb{Z}^+ \end{cases} \tag{4.2}$$

Figura 5 – Região ótima



Fonte – Própria

Analisando o problema e desprezando a restrição das variáveis serem inteiras e assumindo que são não-negativas, a solução ótima desse PPI é:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 18,39408 \\ 1,60173 \end{bmatrix}; Max\ l = 48,9$$

Suponhamos que seja uma boa estratégia determinar os inteiros mais próximos do entorno dessa solução contínua. Vamos analisar os valores da função-objetivo para as possíveis combinações:

Utilizando essa estrategia, concluiríamos de forma errada que a solução ótima seria: $x_1 = 19$ e $x_2 = 1$ com valor ótimo $l = 38$. Na realidade, a solução ótima é: $x_1 = 10$ e $x_2 = 2$, com $l = 48$, ou seja, o valor ótimo difere em 21% do anterior. Notamos que não pode ser uma boa

Tabela 17 – Possíveis valores para o PPL

x1	x2	lucro
19	2	inviável
18	1	37
19	1	38
18	2	inviável

Fonte – Própria

estratégia arrendondar os valores do entorno de uma solução ótima contínua como uma forma de determinar a solução ótima do PPI.

4.2 PROBLEMA DA MOCHILA

Imagine que irá realizar uma viagem com sua família, obviamente necessitamos levar uma mala com itens dentro. Você poderá levar vários itens como: roupas, livros, aparelhos eletrônicos, produtos de higiene, etc. No entanto, uma mala não possui uma capacidade infinita e cada item possui um peso e tamanhos diferentes. Logo, algo poderá ser deixado para trás. Essa ideia logo nos permeia uma questão muito simples, mas com uma resposta não tão simples: quais itens eu devo levar?

Estamos no seguinte problema: você possui apenas uma mala para a sua viagem e vários itens para levar, mas não poderá levar todos, pois, não caberão em sua mala. Uma maneira tentar resolver isso, é atribuir valor de importância para cada item e adicionar apenas os itens mais valiosos na sua mala, colocando o item mais importante entre todos, em seguida, o segundo item mais importante que caiba no espaço restante e assim sucessivamente até completar a capacidade da mala.

Deste modo, definimos o problema da mochila, temos um número limitado de itens para colocar em uma mochila levando em consideração o peso de cada e a capacidade máxima da mochila e queremos maximizar o valor total de utilidade ou lucro.

O problema da mochila é um PLI e é classificado como NP-hard, em que, NP significa "Non-Deterministic Polynomial Time". Esse problema tem complexidade de ordem crescente de 2^n, onde n é o número de itens. Supondo um computador que analisa 1 bilhão de vetores por segundo e retorna a melhor combinação, iríamos precisar de aproximadamente 30 anos para analisar um cado com 60 itens, conforme (MARTELLO; TOTH, 1990) apresentaram. Porém, existe alguns algoritmos que precisam de poucos segundos para resolver problemas com 10000 itens em um computador simples.

4.2.1 Mochila 0-1

O problema da mochila 0-1, ou mochila binária, se denomina assim, pois suas variáveis só podem assumir valores 0 ou 1. Consiste em dada uma mochila de capacidade C e n itens de peso p_i e valor de utilidade v_i, queremos determinar um conjunto de itens a serem colocados na mochila, de modo que o lucro seja máximo. Considerando as variáveis de decisão $x_i, i = \{1, 2, \cdots, n\}$ de maneira que se o item x_i é adicionado na mochila, então $x_i = 1$, caso contrario, $x_i = 0$. Deste modo, usamos a definição do modelo matemático descrito por (MARTELLO; TOTH, 1990):

$$Maximiza\ z = \sum_{i=1}^{n} v_i x_i \tag{4.3}$$

$$Sujeito\ a\ \sum_{i=1}^{n} p_i x_i \leq C, \tag{4.4}$$

$$x_i = \{0, 1\}, i \in \mathbb{Z}^+ = \{1, \cdots, n\} \tag{4.5}$$

Essa variação do problema da mochila estar entre as mais estudadas, pois, pode ser visto como um simples problema de programação inteira. O problema da mochila 0-1 aparece como subproblemas em muitos outros problemas mais complexos, é o caso do problema de Corte de Estoque Unidimensional. Ver (CERQUEIRA, 2009).

Para ilustrar, considere o seguinte problema: um alpinista planeja fazer uma viagem ao pico Agulha do Diablo, localizado no Parque Nacional de Serra dos Órgãos-RJ. Há 5 itens que ele deseja levar consigo, mas estes itens juntos excedem o limite de 12 quilos que ele supõe ser capaz de carregar. Para ajudar a si próprio no processo de seleção dos itens, ele atribuiu valores, por ordem crescente de importância a cada um dos itens conforme a tabela 18:

Tabela 18 – Peso e Valor

Item	1	2	3	4	5
Peso(Kg)	6	5	4	3	2
Valor	8	6	3	3	1

Fonte – Própria

Supondo a existência de uma unidade de cada item, vamos fazer um modelo de programação inteira que maximize o valor total sem exceder as restrições de peso.

(1) **Variáveis de Decisão:** como o problema é um PLI binário, então teremos: $x_i; i = 1, \cdots 6 = \begin{cases} 1, & se\ o\ item\ j\ for\ colocado\ na\ mochila; \\ 0, & caso\ contrrio \end{cases}$

(2) **Função-Objetivo:** o objetivo é colocar os itens na mochila que retorne o maior valor possível. A função-objetivo é dada por: $Max\ v = 8x_1 + 6x_2 + 3x_3 + 3x_4 + 1x_5$.

(3) **Restrições:** teremos duas restrições nesse problema, a primeira restrição consiste na limitação da capacidade da mochila, deste modo, a restrição será: $6x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 \leq 40$. A segunda restrição, será de integralidade, pois a solução deverá ser de forma binária, então: $x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}$

O modelo matemático é o seguinte:

$$\begin{cases} Maximizar & v = 8x_1 + 6x_2 + 3x_3 + 3x_4 + 1x_5 \\ sujeito\ a: & \\ & 6x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 \leq 15 \\ & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\} \end{cases} \tag{4.6}$$

Nesse exemplo simples, a solução ótima é dada pelo vetor $x = (1, 1, 0, 0, 0)$, os métodos de solução serão discutidos mais adiante.

4.2.2 Tipos de mochila

A *mochila inteira*, é uma variação da mochila 0-1. Anteriormente tínhamos as variáveis binárias, mas agora existe uma quantidade maior que uma unidade do mesmo item para ser inserida na mochila. O modelo matemático neste caso é o seguinte:

$$Maximizar\ z = \sum_{i=1}^{n} v_i x_i \tag{4.7}$$

$$Sujeito\ a\ \sum_{i=1}^{n} p_i x_i \leq c, \tag{4.8}$$

$$x_i \in \mathbb{Z}^+; i = 1, \cdots n \tag{4.9}$$

em que,

- $x_i \in \mathbb{Z}^+$: quantidade do item i colocado na mochila;
- v_i: valor de utilidade do item i;
- p_i: peso de cada item;
- C: capacidade da mochila;
- n: a quantidade de itens.

Notamos que a diferença entre a mochila 0-1 e a mochila inteira, está na variável de decisão x_i, que não esta mais sujeita ao conjunto binário, $\{0, 1\}$, assim o x_i pode assumir valores naturais, incluindo o 0 caso o item não seja escolhido.

A aplicação do problema da mochila inteira pode ser vista no exemplo 1, que devemos maximizar a função-objetivo que chamamos de lucro, para obter a maior renda possível. Nosso modelo será:

$$\begin{cases} Maximizar \quad L = 0,50 \times x_1 + 0,40 \times x_2 + 0,60 \times x_3 \\ sujeito\ a: \quad 0,40 \times x_1 + 0,39 \times x_2 + 0,55 \times x_3 \leq 15 \\ \qquad\qquad x_i \geq 0, x_i \in \mathbb{Z}, i = 1,2,3 \end{cases} \tag{4.10}$$

Vamos utilizar uma abordagem diferente para tentar achar uma solução para o problema. Primeiro queremos saber quais são os itens mais valiosos de acordo com o seu peso e lucro, que devemos colocar na mochila. Primeiramente vamos dividir o valor ou lucro do produto pelo seu peso em Kg, logo teremos: V_i = valor de importância de cada produto. Portanto, teremos: $V_1 = \frac{0,50}{0,40} = 1,25, V_2 = \frac{0,40}{0,39} \cong 1,0256, v_3 \cong 1,0909$. Note que o item 1 possui um valor de importância maior, logo vamos preencher a mochila somente com esse produto, respeitando sua capacidade total. Executando esse procedimento, observe que a mochila suporta 37 unidades do item 1 e o valor da função-objetivo será: $L = 1850$. Executando o mesmo procedimento para os itens 2 e 3, notaremos que o valor da função-objetivo será menor que para o item 1. Assim, definimos a nossa solução inicial: $\overline{G} = (37,0,0); L = 18,50$.

Observe que a capacidade da mochila ficou em 14,8kg e portanto, obedeceu a restrição, mas observe que sobraram 200g, que poderia ser preenchidas por outro produto. Se adicionarmos outro produto acabaria por ultrapassar a capacidade da mochila, assim, não obedecendo a restrição. Seguindo o procedimento, podemos pensar agora em retirar uma unidade do item 1 da mochila, assim, a mochila ficará: $(36,0,0)$ e o seu peso ficará, $14,4Kg$. Note que agora restou um espaço de $600g$, desse modo, podemos inserir um item 2 ou 3, vamos colocar o item 3, pois seu lucro será maior em relação ao 2. Agora nossa mochila ficará: $(36,0,1); L = 36 \times 0,50 + 1 \times 0,60 = 18,60$, observe que a função-objetivo será maior, logo a nova solução será: $\overline{G} = (36,0,1); L = 18,60$.

Podemos repetir esse processo até fazer todas as combinações possíveis e encontrarmos a melhor solução, mas observe que esse processo além de ser muito demorado, não é um processo muito eficiente para um número muito grande de variáveis, portanto, não é um método muito viável. Por sorte, a nossa solução ótima para esse problema será o vetor $(36,0,1)$. Adiante, vamos conhecer outros métodos de solução mais eficiente, para resolver esses tipos de PLI.

Em muitos problema reais nem sempre temos apenas um item de cada tipo no problema e também um número infinito de itens a nossa disposição. Geralmente, dispomos de um intervalo fechado de itens $[0,d_i]$ para cada variável $x_i \in \mathbb{Z}^+$, onde $d_i \in \mathbb{Z}^+$ é sua quantidade máxima de itens de peso p_i que podem ser colocados na mochila. Neste caso, teremos uma *mochila inteira*, no qual sua formulação será a seguinte:

$$Maximizar\ z = \sum_{i=1}^{n} v_i x_i \tag{4.11}$$

$$Sujeito\ a\ \sum_{i=1}^{n} p_i x_i \leq c \tag{4.12}$$

$$0 \leq x_i \leq d_i, i = \{1, \cdots, n\}; x_i \in \mathbb{Z}^+ \tag{4.13}$$

4.2.3 Métodos de solução

Vamos apresentar dois métodos de solução para o problema da mochila. Vamos apresentar o modelo de programação dinâmica para o problema da mochila 0-1. Posteriormente, apresentamos o método de enumeração implícita, para o problema da mochila inteira.

4.2.3.1 Programação Dinâmica

Nesse método, cada solução so problema é resolvida a partir da solução de subproblemas menores. Uma característica distintiva da programação dinâmica é a tabela que armazena as soluções dos seus vários subproblemas. Esse método resolve o problema de várias mochilas menores dentro da mochila original, até atingir a capacidade máxima da mochila. Esse método trabalha recursivamente, que normalmente é ineficiente porque terá que refazer os cálculos, mas como criamos uma tabela com os resultados dos subproblemas, evita o recálculo.

Para melhor entender o método, considere uma mochila de tamanho L e os itens x_i com os seus respectivos valores e pesos, a programação dinâmica resolve recursivamente o problema da mochila para o tamanho 0, ou seja, aloca os itens na mochila até atingir a sua capacidade e analisa o valor obtido, se esse valor for menor que o posterior, então esse item é desconsiderado. Posteriormente para a mochila de tamanho 1, depois para a mochila de tamanho 2 e assim sucessivamente até obter uma solução para a mochila de tamanho L.

Considere o seguinte problema da mochila 0-1.

$$\begin{cases} f(x) = Max \quad v_1x_1 + v_2x_2 + \cdots + v_nx_n \\ sujeito\ a: \quad p_1x_1 + p_2x_2 + \cdots + p_nx_n \leq L \\ \qquad\qquad x_i \in \{0,1\}, i = 1, \cdots, n \end{cases} \tag{4.14}$$

O método de programação dinâmica para resolução do problema da mochila, consiste numa forma recursiva de determinar todas as soluções. Ver (YANASSE; SOMA, 1987), (PISINGER, 1995) e (MARTELLO; TOTH, 1990). Neste método a solução de um nó representando um espaço livre X é determinada assim que os nós sucessores $X-L1; X-L2; \cdots; X-Ln$ estiverem resolvidos e denotaremos a função-objetivo por $F(i,X)$, que é o valor máximo obtido ao combinar os valores dos pesos $p_1, p_2, p_3, \cdots, p_i, 1 \leq i \leq n$ ao longo do espaço $X = 0, 1, \cdots L$. Ou seja, teremos:

$$F(i,X) = Max \quad \sum_{k=1}^{i} v_k x_k \tag{4.15}$$

$$s.a \quad \sum_{k=1}^{i} p_k x_k \leq X \tag{4.16}$$

$$x_k \in \{0,1\}, k = 1, \cdots, i \tag{4.17}$$

Os valores $X = 0, 1, \cdots, L$ são os estado das variáveis, que representa diferentes configurações do problema durante o processo de decisão. A fórmula de que define a solução de cada estado é dada por: $F(i,X) = \begin{cases} F(i-1,X) & se\ p_i > X \\ Max\ \{F(i-1,X), F(i-1,X-p_i)+v_i\} & se\ p_i \leq X \end{cases}$ em que , $1 \leq i \leq n, 0 \leq X \leq L$.

Temos que o $F(i,X)$ é igual a $F(i-1,X)$ caso este novo item seja maior do que a capacidade atual da mochila, caso o contrario ele será igual ao máximo entre o estado atual sem o item $F(i-1,X)$ e $F(i-1,X-p_i)+v_i$ onde é colocado o item e sua capacidade é reduzida conforme o peso do item que é alocado e então, é inserido o valor do item na função.

$F(n,X)$, é o valor máximo obtido para o estado X e n é a quantidade máxima de itens. Este valor é o resultado na sequencia de operações tomadas até o estágio X. $F(n,L)$ corresponde à solução ótima do problema da mochila. Agora apresentaremos um pseudocódigo proposto por (CARVALHO, 2015) deste método:

```
%%Procedimento Programação Dinamica

Início procedimento

  n = Insirá a quantidade de itens;
  v = Insirá os valores de v_i;
  p = Insirá os pesos de p_i;
  L = Insirá a capacidade da mochila;

  Para j = 0 até L faça
    F(0,j) = 0;
  fim para

  Para i = 1 até n faça
    Para j = 0 até L faça
      se (( p(i) <= j ) e ( v(i) + F(i - 1,j - p(i) ) > F(i - 1,j) )) então
         F(i,j) = v(i) + F(i - 1, j - p(i));
```

```
          armazena(i,j) = 1;
      senão
         F(i,j) = G(i - 1,j);
          armazena(i,j) = 0;
      fim
    fim
  fim

  Para i = n até 1 faça
    se (armazena(i,j) == 1)
      exiba i;
       j = j - p(i);
    fim
  fim
Retorne F(n,L);

Fim procedimento
```

O vetor `armazena(i,j)` armazena quais itens foram colocados na mochila e no final do algoritmo, ele é percorrido para determinar a solução ótima, gerando uma tabela de valores.

4.2.3.2 Enumeração Implícita

O método de enumeração implícita para resolução do problema da mochila apresentado a seguir, é sugerido por (GILMORE; GOMORY, 1961). Este método realiza uma busca em profundidade primeiro e consiste em enumerar implicitamente todas as solução do problema da mochila, permitindo através do uso de limitantes, que as piores soluções sejam descartada, sem perda da solução ótima.

Neste método, iremos resolver problemas em que a quantidade de itens a serem escolhidos é limitada dentro de um intervalo do tipo $[0, d_i]$, e $d_i \in \mathbb{Z}^+$ é a quantidade máxima de itens disponíveis para alocação na mochila.

O algoritmo de enumeração implícita para resolução do problema da mochila é o seguinte:

Inicio do Procedimento Enumeração Implícita

Dados os vetores $v = valor, p = peso, d = numero\ de\ itens$ e os inteiros, n e L,

Passo 1: Ordene os itens mais valiosos por unidade de comprimento

Faça $\theta_i = \dfrac{v_i}{p_i}$, com $i = 1, \cdots, n$, e reordene as variáveis em ordem decrescente e sem perda de generalidade suponha que: $\theta_i \geq \theta_2 \geq \theta_3 \geq \cdots \geq \theta_n$

Faça $v_{n+1} = 0, p_{n+1} = 1, \underline{G} = 0$ e $g(\sigma) = 0$

Passo 2; determine a solução inicial, utilizando a busca em profundidade primeiro

Determine $\sigma = (\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_n)$ de modo

$$\alpha_1 = mim\left\{\left\lfloor \frac{L}{p_1} \right\rfloor, d_1\right\} \Rightarrow \text{resta } L - p_1\alpha_1$$

$$\alpha_2 = mim\left\{\left\lfloor \frac{L - p_1\alpha_1}{p_2} \right\rfloor, d_2\right\} \Rightarrow \text{resta } L - p_1\alpha_1 - p_2\alpha_2$$

$$\vdots$$

$$\alpha_n = min\left\{\left\lfloor \frac{L - \sum_{j=1}^{n-1} p_j\alpha_j}{p_n} \right\rfloor, d_n\right\}$$

Passo 3: Avalie a solução corrente e armazene a mais valiosa

Faça $g(\sigma) = \sum_{i=1}^{n} v_i\alpha_i$

Se $\underline{G} < g(\sigma)$ então $\underline{G} = g(\sigma)$(inicialmente $\underline{G} = 0$) e $\sigma^* = \sigma$

Passo 4: Teste a otimalidade e calcule o limitante superior

Determine o maior índice k tal que $\alpha_k \neq 0$

Se $\sigma = 0$ então PARE, σ^* é a melhor solução.

Caso contrário, calcule

$$\overline{G}(\sigma) = v_1\alpha_1 + v_2\alpha_2 + \cdots + v_k(\alpha_k - 1) + \frac{v_{k+1}}{p_{k+1}}(L - p_1\alpha_1 - p_2\alpha_2 - \cdots - p_k(\alpha_k - 1))$$

Passo 5: Backtracking

Se $\overline{G}(\sigma) \leq \underline{G}$ então faça $\alpha_k = 0$ e volte ao *Passo 4.*

Caso contrário, faça $\alpha_k = \alpha_k - 1$ e defina σ com

$$\alpha_j = min\left\{\left\lfloor \frac{L - \sum_{i=1}^{j-1} p_i\alpha_i}{p_j} \right\rfloor, d_j\right\}, j = k+1, \cdots, n \text{ e volte ao } \textit{Passo 3}.$$

Fim do Procedimento

Ao final do método, $g(\sigma)$ guarda o valor ótimo encontrado e o vetor σ é a solução do problema.

5 FERRAMENTAS COMPUTACIONAIS

5.1 REVISÃO BIBLIOGRÁFICA

O uso da programação linear é um instrumento importante de otimização para problemas de Economia, problemas de transportes, gestão de empresas, obtenção de misturas ótimas, roteamento, entre outros. A resolução de forma analítica destes problemas fica impraticável dependendo da quantidade de variáveis que serão utilizadas, neste caso, o uso das ferramentas computacionais nos fornecem uma solução mais rápida e confiável.

Conforme (ZACHI, 2016) o (GEOGEBRA,) funciona como facilitador na construção gráfica e interpretação dos sistemas de equações e inequações, assim como a visualização das regiões planas, facilitando o entendimento do cálculo do ponto de intersecção de retas. Deste modo, o uso do Geogebra facilita na visualização da região ótima e nos fornece uma solução mas rápida.

O GeoGebra em muitos problemas reais, acaba sendo uma ferramenta inviável, pois, como é um software principalmente gráfico, acaba ficando restrito a somente problemas que trabalhem com duas à 3 variáveis de decisão. Partindo dessa limitação, podemos utilizar ferramentas mais poderosas como o LINGO (Linear, Interactive, and Discrete Optimizer), para obter as soluções ótimas dos problemas com mais de 3 variáveis de decisão.

Segundo (JÚNIOR; SOUZA, 2004) o LINGO é uma poderosa ferramenta para resolver problemas de otimização, minimizando ou maximizando a função objetivo de exercícios de programação linear.

O uso das ferramentas computacionais, traz uma segurança de confiabilidade dos resultados obtidos, deste modo, verifica-se que a teoria só pode ser utilizada com total amplitude utilizando as ferramentas computacionais para a resolução do PPL. Para (WEBER, 2013) com a pesquisa sobre otimização combinatória, verificou-se a necessidade de uma ferramenta de apoio à resolução de problemas desta natureza. Assim, as ferramentas computacionais estão inteiramente ligadas com os processos de obtenção das soluções ótimas dos modelos.

5.2 O USO DO GEOGEBRA NA RESOLUÇÃO DE UM PROBLEMA DE PROGRAMAÇÃO LINEAR

Nesta seção, mostraremos como resolver um problema de programação linear utilizando o software Geogebra. Este método só permite resolver problemas de Programação Linear de pequena dimensão, com duas ou três variáveis. Vamos utilizar um exemplo de Programação

Linear, o Problema do Sítio com duas variáveis de decisão.

5.2.1 Resolvendo o PPL

Problema do Sítio. Manoel após alguns anos de trabalho, resolve comprar uma fazenda de 90 hectares para plantar café e cacau. Cada hectare de café gera um lucro de $R\$400,00$ e cada hectare de cacau, retorna $R\$600,00$ de lucro. O número de empregados e fertilizantes necessários para cada hectare são descritos na tabela 19. Considerando que Manoel pode contar com 200 empregados e 240 toneladas de fertilizantes, como ele pode maximizar seu lucro?

Tabela 19 – Recursos Humanos e Materiais Disponíveis

	Café	Cacau
Empregados	6	4
Fertilizantes	4 toneladas	8 toneladas

Fonte – Própria

Vamos construir o modelo matemático deste problema

(1) **Seleção das Variáveis de Decisão:** Manoel precisa decidir a quantidade de hectares que vai plantar de café e cacau. Assim, sejam x_1 a quantidade de hectares onde será plantado café e x_2 a quantidade de hectares onde será plantado o cacau.

(2) **Função-Objetivo:** Manoel deseja obter o lucro máximo, ou seja, maximizar a função objetivo. Cada hectare cultivado com café gera um lucro, após a colheita de $R\$400,00$, e cada hectare de cacau, após a colheita gera o lucro de $R\$600,00$. Portanto a função objetivo é $Lucro = 400x_1 + 600x_2$.

(3) **Restrições:** Além das restrições de não-negatividade, temos as restrições definidas no problema de área total do sitio, quantidade de empregados e de fertilizantes disponíveis. Assim, temos três restrições:

   i. Área Total: $x_1 + x_2 \leq 90$;

   ii. Número de empregados: $6x_1 + 4x_2 \leq 200$;

   iii. Quantidade de Fertilizantes: $4x_1 + 8x_2 \leq 240$.

Deste modo, podemos escrever o seguinte PPL:

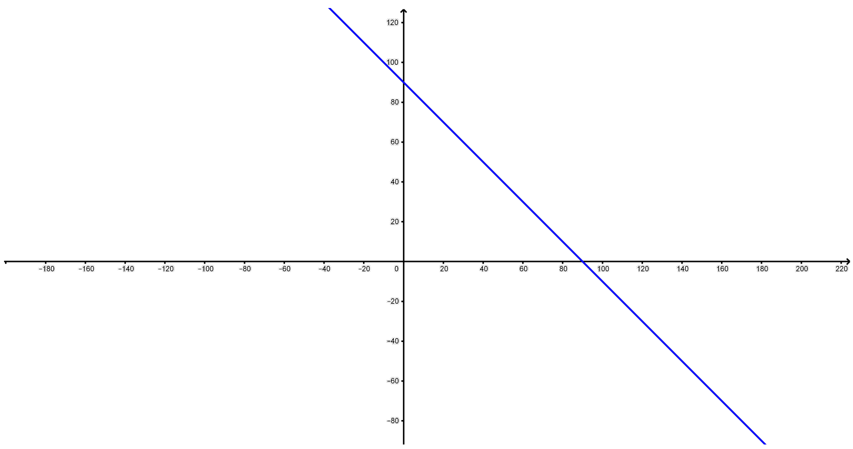
$$\begin{cases} Maximizar \quad L = 400x_1 + 600x_2 \\ sujeito\ a: \\ \qquad\qquad x_1 + x_2 \leq 90 \\ \qquad\qquad 6x_1 + 4x_2 \leq 200 \\ \qquad\qquad 4x_1 + 8x_2 \leq 240 \\ \qquad\qquad x_1 \geq 0, x_2 \geq 0 \end{cases} \tag{5.1}$$

Para resolver um PPL, devemos primeiramente seguir alguns passos.. Esses passos serão mostrados a seguir, juntamente com a resolução do problema acima.

1° Passo: Determinar o conjunto dos pontos que satisfazem as restrições do PPL. Neste passo, precisamos determinar a intersecção das regiões definidas por cada uma das inequações.

Primeiramente devemos determinar a região do plano x_1, x_2 que satisfaz a restrição $x_1 + x_2 \leq 90$. Traçamos a reta $x_1 + x_2 = 90$ e posteriormente identificamos quais dos dois semi-planos determinados pela reta, compreende o conjunto dos pontos (x_1, x_2) que satisfaz a restrição. Uma maneira simples de determinamos quais desses dois semi-planos é o correto, é pegamos um ponto P qualquer e verificar se este satisfaz a desigualdade. Se satisfazer a desigualdade, significa que todos os pontos deste semi-plano satisfazem a inequação, caso contrario, o outro semi-plano que não contém o ponto P, satisfaz a inequação analisada.

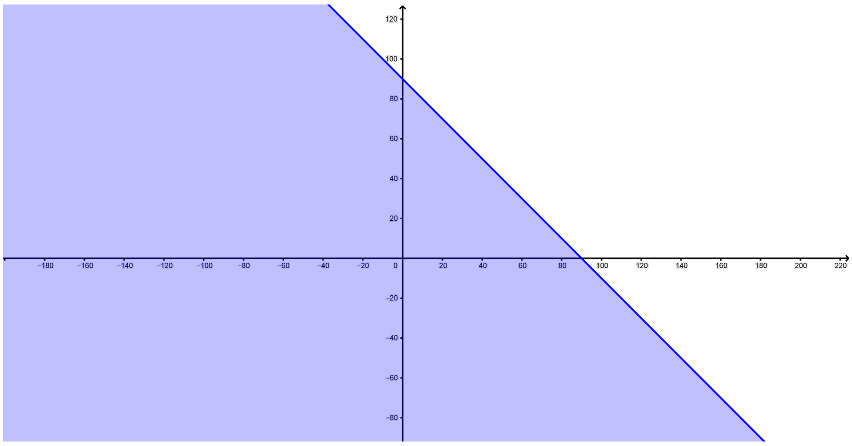
Figura 6 – Reta $x_1 + x_2 = 90$



Fonte – Própria

A figura 6 traz a reta $x_1 + x_2 = 90$. Agora vamos tomar o ponto $P = (0,0)$. Temos que P satisfaz a inequação $x_1 + x_2 \leq 90$, isso significa que o semi-plano indicado no figura 7, satisfaz a inequação, ou seja, a região compreendida embaixo da reta $x_1 + x_2 = 90$ satisfaz a inequação.

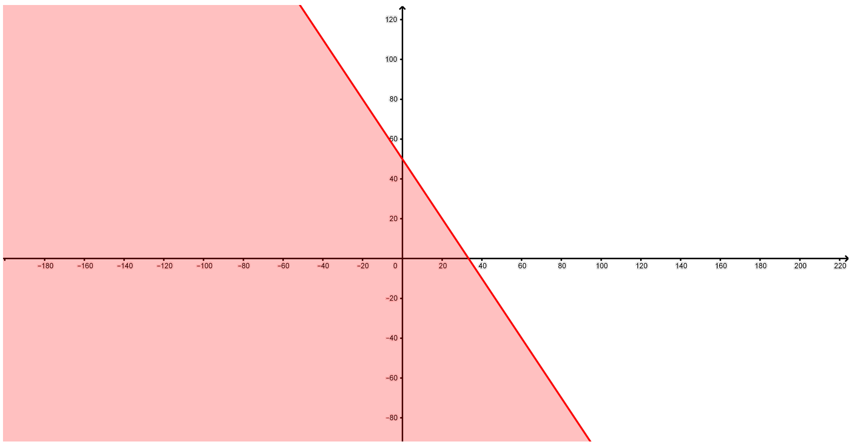
Figura 7 – Semi-plano que satisfaz $x_1 + x_2 \leq 90$



Fonte – Própria

Continuamos traçando a inequação $6x_1 + 4x_2 \leq 200$, para determinar o semi-plano que compreende os pontos que satisfazem a segunda restrição, conforme mostra na figura 8.

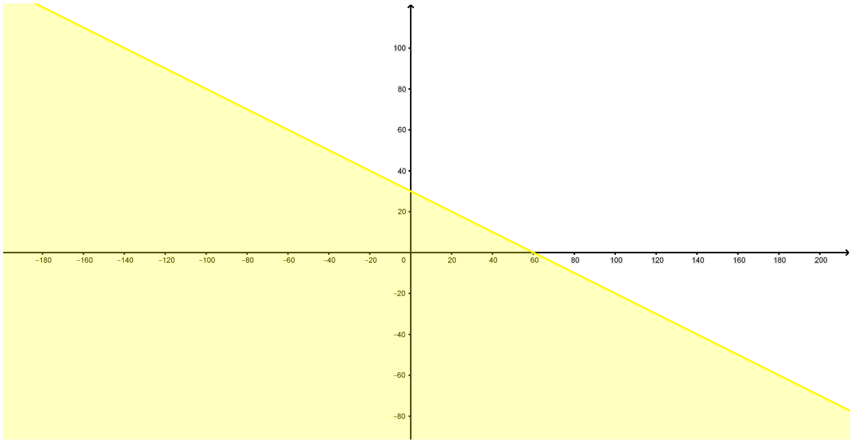
Figura 8 – Semi-plano que satisfaz $6x_1 + 4x_2 \leq 200$



Fonte – Própria

A figura 9, traz o semi-plano da ultima restrição, $4x_1 + 8x_2 \leq 240$.

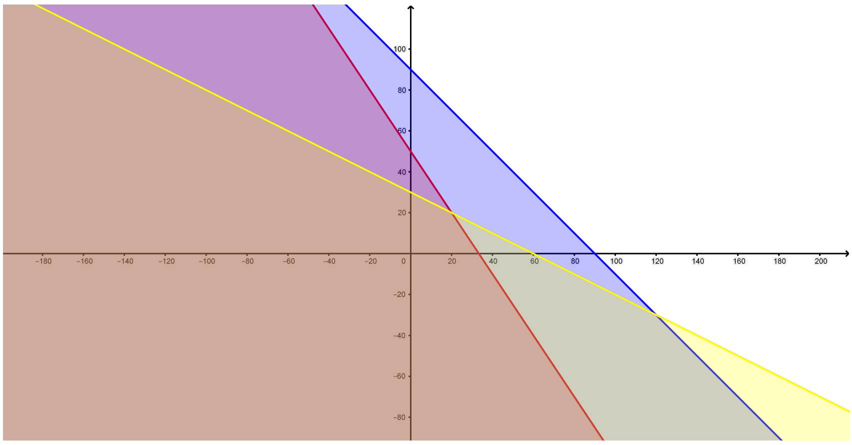
Figura 9 – Semi-plano que satisfaz $4x_1 + 8x_2 \leq 240$



Fonte – Própria

A figura 10, mostra a interseção dos três semi-planos $x_1 + x_2 \leq 90$, $6x_1 + 4x_2 \leq 200$, $4x_1 + 8x_2 \leq 240$.

Figura 10 – Semi-planos $x_1 + x_2 \leq 90$, $6x_1 + 4x_2 \leq 200$ e $4x_1 + 8x_2 \leq 240$



Fonte – Própria

Inserindo a última restrição, $x_1 \leq 0$ e $x_2 \leq 0$, obtemos a região mais escura mostrada na figura 11. Chamaremos esta região de viável, ou seja, qualquer ponto pertencente a esta região, satisfaz todas as restrições do problema. Agora basta determinamos o ponto que maximiza a função-objetivo lucro.

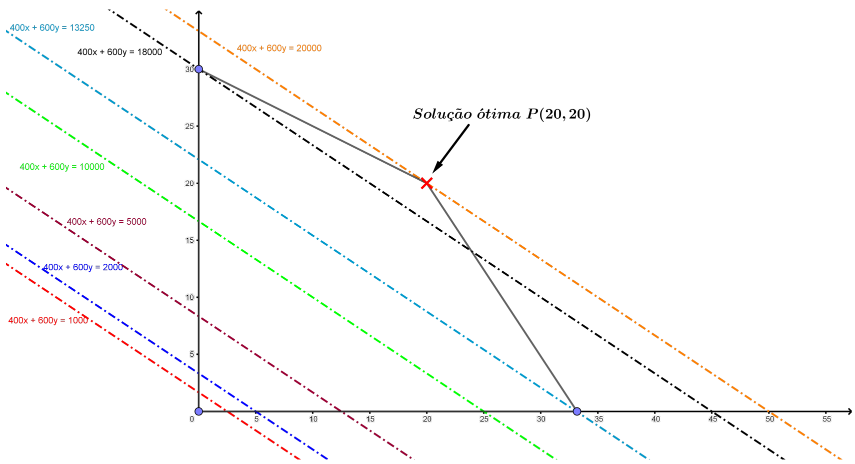
Figura 11 – Região viável do Problema do Sítio

Fonte – Própria

Passo 2: Encontrar a solução da função-objetivo na região viável.

Após encontrada a região viável, podemos tentar atribuir valores a função-objetivo $Lucro = z_0$ e verificar se a reta intercepta a região viável. Se $z_0 \geq 0, z_0 \in \mathbb{R}$ qualquer valor da reta $400x_1 + 600x_2 = z_0$ que intercepta a região viável, satisfaz as restrições e tem o $Lucro = z_0$. Observe que na figura 12, traz as retas tracejadas para o $Lucro = 1000; 2000; 5000; 10000; 13250; 18000; 20000$. Observe que a reta $400x_1 + 600x_2 = 10000$ intercepta o ponto $(20, 20)$ da região viável e caso aumentamos os valores de z_0 as respectivas restas não interceptarão mais a região viável. Concluímos que o ponto onde o lucro é máximo é justamente no ponto $P(20, 20)$, que chamamos de *solução ótima*, como mostrado na figura 12. Note também que outra solução viável é no ponto $(0, 30)$, observe que aparentemente as regiões onde se encontra a solução ótima, sempre recaem no vértice da nossa região viável.

Figura 12 – Retas da Função-objetivo



Fonte – Própria

Portanto, Manoel obterá um lucro máximo se ele plantar 20 hectares de café e 20 hectares de cacau.

Observe que toda vez que queremos encontrar um valor ótimo para a função-objetivo, teremos que proceder através do método de tentativa e erro. Notamos que isso implica em um método não muito eficaz, pois dependendo da função, teremos que fazer várias tentativas para encontrar a solução ótima, isso é claro, se não tivemos usando um software de gráficos, como o GeoGebra, usado para plotar esses gráficos. Para agilizar nossos cálculos de obter uma solução ótima de um Problema de Programação Linear, notemos que a solução como dito anteriormente, está localizada num dos vértices da região poligonal que forma a região viável. Podemos reformular o passo 2 visto anteriormente, se existir solução ótima finita, então ela estará em um dos vértices da região viável. Assim, podemos reformular o passo 2:

Passo 2: Calculemos os valor da função-objetivo nos pontos de vértice na região viável. O ponto em que resultar no maior valor da função-objetivo, será a solução ótima, conforme na tabela 20.

Tabela 20 – Função-objetivo nos pontos de vértice

Ponto Extremo	Lucro (R$)
(0,0)	0
(33.18,0)	13250
(20,20)	20000
(0,30)	18000

Fonte – Própria

5.3 UTILIZANDO O LINGO PARA RESOLUÇÃO DE PPL

Vamos agora trabalhar alguns exemplos de PPL, visando uma melhor compreensão. Iremos modelar e resolver utilizando o Software (LINGO,). O Software LINGO criado em 1988, tornou-se o primeiro produto da LINDO a incluir uma linguagem de modelagem completa. Os usuários puderam utilizar a linguagem de modelagem para expressar modelos de forma concisa utilizando resumos e variáveis subscritas. Em 1993, a LINGO adicionou um solver não-linear de grande escala. Era único em que o usuário não precisava especificar qual o Solver deveria usar.

Com a adição do solver não linear, o LINGO substituiu essencialmente o GINO como produto de estreia da LINDO Systems para otimização não linear. GINO estreou em 1984 e foi o primeiro solucionador não-linear disponível na plataforma PC. Em 1994, a LINGO tornou-se o primeiro software de linguagem de modelagem a ser incluído em um texto de ciência de gestão popular. Em 1995, ocorreu o primeiro lançamento do Windows LINGO. Hoje, a LINDO Systems continua a desenvolver versões mais rápidas e poderosas.

5.3.1 Problema da Dieta

Suponha por motivos nutricionais, que Marcelo tenha entrado na Academia para emagrecer e adquirir uma musculação mais definida, ao consultar o nutricionalista, ele recebeu uma dieta alimentar para redução de peso e ganho de massa magra. Marcelo pretende ingerir os nutrientes necessários para manter a saúde minimizando o número de calorias. Considere a tabela abaixo, como deverá ser a dieta de Marcelo?

Tabela 21 – Alimentos e Nutrientes

Porção de 100g	Proteína (g)	Carboidratos(g)	Lipídeos(g)	Calorias(Kcal)
Arroz Integral Cozido	2,6	25,8	1	123,5
Ovo de Galinha Inteiro Cozido	13,3	0,6	9,5	145,7
Peito de Frango sem Pele Grelhado	32,0	-	2,5	159,2
Carne Bovina Patinho Sem Gordura	35,9	-	7,3	219,3
Mandioca Cozida	0,60	30,1	0,20	119,0
Feijão Carioca Cozido	4,8	13,6	0,5	76,4

Fonte – Própria

Assim teremos:

(1) **Variáveis de Decisão:** Marcelo precisa saber quantas porções de $100g$ de cada alimento irá consumir. Deste modo, podemos definir as seguintes variáveis de decisão: x_i a quantidade de porções do alimento i consumidos, com $i = 1, \cdots 6$, onde atribuímos 1 à porção de arroz integral, 2 ao ovo de galinha, 3 ao peito de frango, 4 carne bovina, 5 à mandioca, 6 ao feijão.

(2) **Função-objetivo:** Marcelo deseja minimizar o número de calorias ingeridas. Sua função-objetivo é a quantidade de calorias ingeridas, assim minimizando, podemos reduzir o ganho de gordura em Marcelo: $z = 123,5x_1 + 145,7x_2 + 159,2x_3 + 219,3x_4 + 119x_5 + 76,4x_6$.

(3) **Restrições:** Além da restrição de não-negatividade, temos os nutrientes mínimos diários:

(i) *Proteína:* $2,6x_1 + 13,3x_2 + 32x_3 + 35,9x_4 + 0,60x_5 + 4,8x_6 \geq 100$

(ii) *Carboidratos:* $25,8x_1 + 0,6x_2 + 30,1x_5 + 13,6x_6 \geq 320$

(iii) *Lipídeos:* $x_1 + 9,5x_2 + 2,5x_3 + 7,3x_4 + 0,20x_5 + 0,5x_6 \geq 11$

Podemos escrever o seguinte PPL:

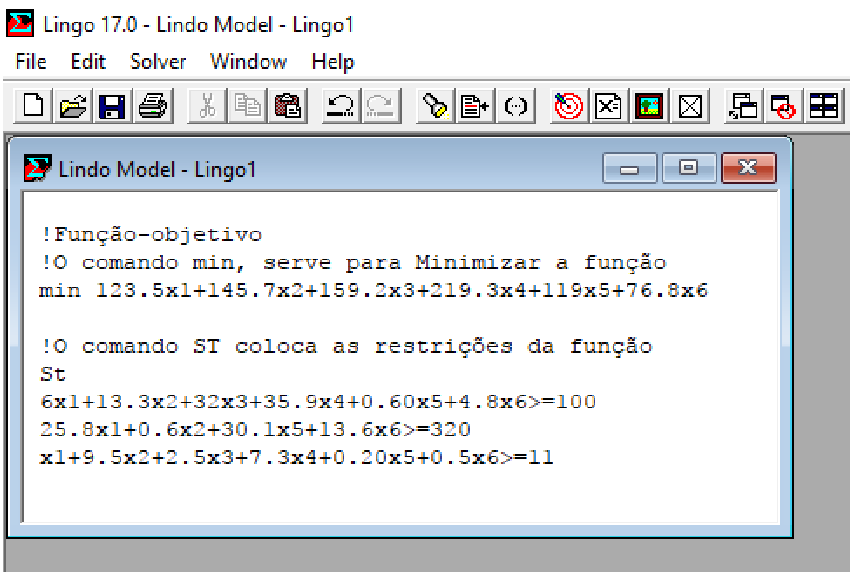
$$
\begin{cases}
Minimizar \quad z = 123,5x_1 + 145,7x_2 + 159,2x_3 + 219,3x_4 + 119x_5 + 76,4x_6 \\
sujeito\ a: \\
\qquad 2,6x_1 + 13,3x_2 + 32x_3 + 35,9x_4 + 0,60x_5 + 4,8x_6 \geq 100 \\
\qquad 25,8x_1 + 0,6x_2 + 30,1x_5 + 13,6x_6 \geq 320 \\
\qquad x_1 + 9,5x_2 + 2,5x_3 + 7,3x_4 + 0,20x_5 + 0,5x_6 \geq 11 \\
\qquad x_i \geq 0, i = 1, \cdots, 6.
\end{cases}
\tag{5.2}
$$

Este PPL não pode ser resolvido graficamente, vamos resolver este PPL utilizando o Software LINGO.

No LINGO devemos colocar o seguintes comandos:

```
min 123.5x1+145.7x2+159.2x3+219.3x4+119x5+76.8x6
St
6x1+13.3x2+32x3+35.9x4+0.60x5+4.8x6>=100
25.8x1+0.6x2+30.1x5+13.6x6>=320
x1+9.5x2+2.5x3+7.3x4+0.20x5+0.5x6>=11
```

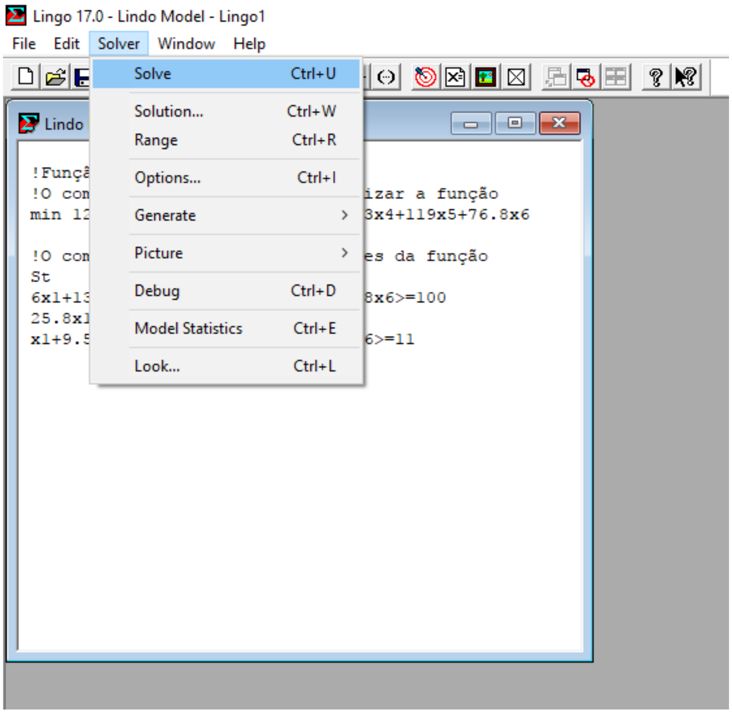
Figura 13 – Colocando o Modelo no LINGO



Fonte – Própria

O comando *Solve* 14 resolve o modelo, função-objetivo e suas respectivas restrições, utilizando o método Simplex.

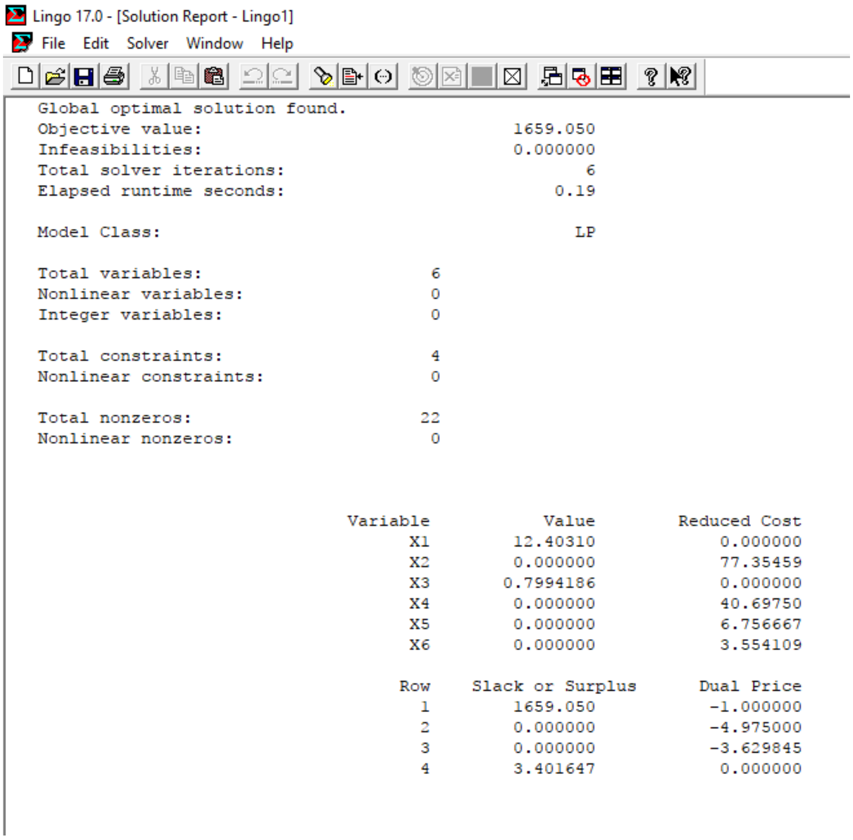
Figura 14 – Executando o Solve



Fonte – Própria

O nosso PPL resolvido 15, contém informações sobre a função-objetivo e suas restrições.

Figura 15 – Informações do Modelo Resolvido


Lingo 17.0 - [Solution Report - Lingo1]
File Edit Solver Window Help

```
Global optimal solution found.
Objective value:                              1659.050
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.19

Model Class:                                        LP

Total variables:                      6
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    4
Nonlinear constraints:                0

Total nonzeros:                      22
Nonlinear nonzeros:                   0


                              Variable           Value        Reduced Cost
                                    X1        12.40310            0.000000
                                    X2        0.000000            77.35459
                                    X3       0.7994186            0.000000
                                    X4        0.000000            40.69750
                                    X5        0.000000            6.756667
                                    X6        0.000000            3.554109

                                   Row    Slack or Surplus      Dual Price
                                     1        1659.050           -1.000000
                                     2        0.000000           -4.975000
                                     3        0.000000           -3.629845
                                     4        3.401647            0.000000
```

Fonte – Própria

Analisando a figura 15, observamos que existe uma solução mínima e ótima. Essa solução é:

$$z = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 12,40310 \\ 0 \\ 0,7994186 \\ 0 \\ 0 \\ 0 \end{bmatrix}; Min\ z = 1659,050$$

5.3.2 Problema da Mistura

Vamos estudar outro Problema de Programação Linear contínuo proposto por (SANTOS, 2000).

Uma companhia deseja obter uma nova liga metálica com 30% de chumbo, 20% de zinco e 50% de estanho a partir de alguns minérios tendo as seguintes propriedades:

Tabela 22 – Tabela das propriedades do Minério

Propriedades	*Minério 1*	*Minério 2*	*Minério 3*	*Minério 4*	*Minério 5*
%-Chumbo	30	10	50	10	50
%-Zinco	60	20	20	10	10
%-Estanho	10	70	30	80	40
Custo($Kg)	8,5	6	8,9	5,7	8,8

Fonte – (SANTOS, 2000)

O objetivo é determinar as proporções destes minérios que deveriam ser misturados para produzir a nova liga com o menor custo possível.

Modelando o PPL, temos:

(1) **Variáveis de Decisão:** Temos na tabela 22, 5 tipos de mineiro diferente, então vamos denominar, x_1 para o mineiro 1, x_2 para o mineiro 2 e seguindo assim sucessivamente. Deste modo, vamos ter: x_i= fração de 1 quilo do minério i usada na produção de 1 quilo da nova liga, $x_i; i = 1, \cdots 5$.

(2) **Função-Objetivo:** Nosso objetivo é buscar a redução de custo, ou seja, minimizar o custo, portanto, nossa função-objetivo será: $l = 8,5x_1 + 6x_2 + 8,9x_3 + 5,7x_4 + 8,8x_5$.

(3) **Restrições:** Temos nossas restrições da quantidade de elementos que temos por quilo de minério, logo, nossas restrições serão: $0,3x_1 + 0,1x_2 + 0,5x_3 + 0,1x_4 + 0,5x_5 = 0,3$; $0,6x_1 + 0,2x_2 + 0,2x_3 + 0,1x_4 + 0,1x_5 = 0,2$; $0,1x_1 + 0,7x_2 + 0,3x_3 + 0,8x_4 + 0,4x_5 = 0,5$. Também não podemos esquecer da restrição de não-negatividade e da restrição de quantidade de minério, que não pode passar de 1k. Portanto, as duas ultimas restrições serão: $x_1 + x_2 + x_3 + x_4 + x_5 = 1$ e $x_i \geq 0$.

Nosso modelo ficará da forma:

$$\begin{cases} Minimizar \quad l = 8,5x_1 + 6x_2 + 8,9x_3 + 5,7x_4 + 8,8x_5 \\ sujeito\ a: \\ \qquad x_1 + x_2 + x_3 + x_4 + x_5 = 1 \\ \qquad 0,3x_1 + 0,1x_2 + 0,5x_3 + 0,1x_4 + 0,5x_5 = 0,3 \\ \qquad 0,6x_1 + 0,2x_2 + 0,2x_3 + 0,1x_4 + 0,1x_5 = 0,2 \\ \qquad 0,1x_1 + 0,7x_2 + 0,3x_3 + 0,8x_4 + 0,4x_5 = 0,5 \\ \qquad x_i \geq 0, i = 1, \cdots, 3. \end{cases} \tag{5.3}$$

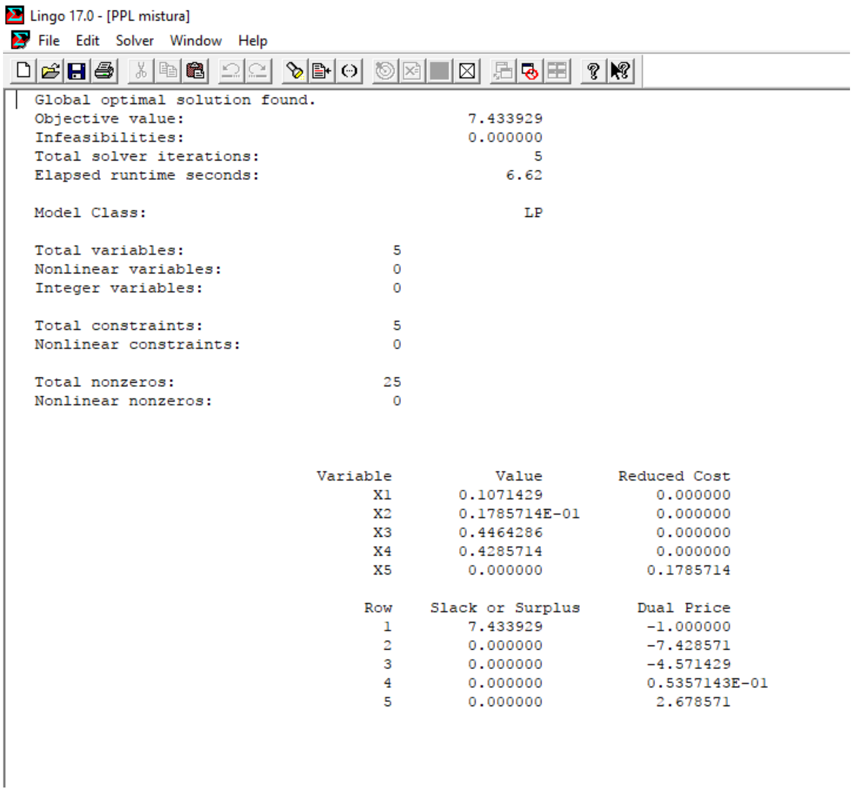
Vamos inserir o modelo no LINGO:

```
min  8.5x1+6x2+8.9x3+5.7x4+8.8x5
```

```
St
x1+x2+x3+x4+x5=1
0.3x1+0.1x2+0.5x3+0.1x4+0.5x5=0.3
0.6x1+0.2x2+0.2x3+0.1x4+0.1x5=0.2
0.1x.1+0.7x2+0.3x3+0.8x4+0.4x5=0.5
```

Colocando para resolver, vamos obter o seguinte resultado como segue na figura 16.

Figura 16 – Resultado PPL da Mistura




```
Global optimal solution found.
Objective value:                              7.433929
Infeasibilities:                              0.000000
Total solver iterations:                             5
Elapsed runtime seconds:                          6.62

Model Class:                                        LP

Total variables:                      5
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    5
Nonlinear constraints:                0

Total nonzeros:                      25
Nonlinear nonzeros:                   0


                              Variable           Value        Reduced Cost
                                    X1       0.1071429            0.000000
                                    X2       0.1785714E-01        0.000000
                                    X3       0.4464286            0.000000
                                    X4       0.4285714            0.000000
                                    X5        0.000000           0.1785714

                                   Row    Slack or Surplus      Dual Price
                                     1        7.433929           -1.000000
                                     2        0.000000           -7.428571
                                     3        0.000000           -4.571429
                                     4        0.000000           0.5357143E-01
                                     5        0.000000            2.678571
```

Fonte – Própria

Analisando a figura 16 obtemos a solução mínima ótima:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0,1071429 \\ 0,1785714 \\ 0,4464286 \\ 0.4285714 \\ 0 \end{bmatrix} ; Min\ l = 7,433929$$

Desse modo, concluímos o nosso estudo dos problemas de programação linear contínua. Existe vários PPL de diferentes modelos neste trabalho, procuramos abordar alguns problemas por seu fácil entendimento. Diversos autores trabalham problemas de tipos e especificidades diferentes, como (BAZARRA; JARVIS; SHERALI, 2009), (GOLDBARG; LUNA, 2005) (GRIF-

FIN, 2014) (SANTOS, 2000), entre outros. Iremos ver agora como resolver os Problemas de Programação Linear Inteira-PPI.

5.3.3 Problema da Fábrica

Uma empresa tem 3 fábricas com ociosidade na produção. Todas as 3 fábricas tem capacidade de produzir um certo produto e a gerência decidiu usar uma parte da ociosidade na produção deste produto. O produto pode ser feito em 3 tamanhos: grande, médio e pequeno, que dão um lucro líquido de \$12, \$10 e \$9 respectivamente. As fábricas 1, 2 e 3 tem capacidade de fabricar 500, 600 e 300 unidades do produto respectivamente, independentemente do tamanho a ser produzido. Há, no entanto, limitação do espaço para estocagem.

As fábricas 1, 2 e 3 tem 9000, 8000 e 3500 m^2 de área para estocagem respectivamente. Cada unidade de tamanho grande, médio e pequeno necessita de 20, 15 e 12 m^2 respectivamente. O Departamento de vendas indicou que 600, 800 e 500 unidades dos tamanhos grande, médio e pequeno, respectivamente, podem ser vendidas por dia. De maneira a manter uma certa uniformidade, a gerencia decidiu que a percentagem do uso das capacidades ociosas das 3 fábricas devem ser iguais. A gerência deseja saber quanto de cada tamanho deve ser produzido em cada fábrica de maneira que o lucro seja máximo.

Temos que modelar o problema.

(1) **Variáveis de Decisão:** Nosso PPL possui algumas informações importantes para ser extraídas. A empresa possui 3 fábricas, que vamos denominar de $x_i; i = 1, \cdots 3$. Todas as 3 fábricas produz o produto em 3 tamanhos diferentes, ou seja, grande, médio e pequeno, que vamos denominar de 1, 2 e 3 respectivamente, $x_j; j = 1, \cdots 3$. Temos então que: $x_{ij} =$ quantidade a ser produzida do tamanho j com $j = 1, \cdots 3$ na fábrica i com $i = 1, \cdots 3$. Se produzimos o produto na fábrica 2 de tamanho pequeno será a variável: x_{23}, se produzirmos um produto grande na fábrica 1 será a variável: x_{11}, e assim segue o raciocínio.

(2) **Função-Objetivo:** nosso modelo busca maximizar a função Lucro, obtendo a quantidade de produtos de cada tamanho deve ser produzido em cada fábrica de uma maneira que o lucro seja máximo. Portando a nossa função-objetivo será: $Max\ z = 12x_{11} + 10x_{12} + 9x_{13} + 12x_{21} + 10x_{22} + 9x_{23} + 12x_{31} + 10x_{32} + 9x_{33}$.

(3) **Restrições:** neste momento vamos analisar as restrições impostas pelo problema. As nossas primeiras restrições, se origina da capacidade da fábrica em produzir o produto, ou seja, nossas três primeiras restrições são: $x_{11} + x12 + x_{13} \leq 500, x_{21} + x22 + x_{23} \leq 600, x_{31} + x32 + x_{33} \leq 300$. A nossas outras 3 restrições vem da capacidade de cada fábrica de estocagem dos produtos: $20x_{11} + 15x_{12} + 12x_{13} \leq 9000, 20x_{21} + 15x_{22} + 12x_{23} \leq 8000, 20x_{31} + 15x_{32} + 12x_{33} \leq 3500$. Nossas outras três restrições se origina da quantidades de vendas por dia: $x_{11} + x12 + x_{13} \leq 600, x_{21} + x22 + x_{23} \leq 800, x_{31} + x32 + x_{33} \leq 500$.

Nossas ultimas restrições vem dá não-negatividade das variáveis e da percentagem do uso das capacidades das 3 fábricas: $x_{ij} \geq 0$ e $\frac{x_{11}+x_{12}+x_{13}}{500} = \frac{x_{21}+x_{22}+x_{23}}{600} = \frac{x_{31}+x_{32}+x_{33}}{300}$.

Nosso modelo ficará:

$$\begin{cases} Maximizar & z = 12x_{11} + 10x_{12} + 9x_{13} + 12x_{21} + 10x_{22} + 9x_{23} + 12x_{31} + 10x_{32} + 9x_{33} \\ sujeito\ a: & \\ & x_{11} + x12 + x_{13} \leq 500 \\ & x_{21} + x22 + x_{23} \leq 600 \\ & x_{31} + x32 + x_{33} \leq 300 \\ & 20x_{11} + 15x_{12} + 12x_{13} \leq 9000 \\ & 20x_{21} + 15x_{22} + 12x_{23} \leq 8000 \\ & 20x_{31} + 15x_{32} + 12x_{33} \leq 3500 \\ & x_{11} + x12 + x_{13} \leq 600 \\ & x_{21} + x22 + x_{23} \leq 800 \\ & x_{31} + x32 + x_{33} \leq 500 \\ & \frac{x_{11}+x_{12}+x_{13}}{500} = \frac{x_{21}+x_{22}+x_{23}}{600} = \frac{x_{31}+x_{32}+x_{33}}{300} \\ & x_{ij} \geq 0, i = 1, \cdots, 3; j = 1, \cdots, 3. \end{cases} \tag{5.4}$$

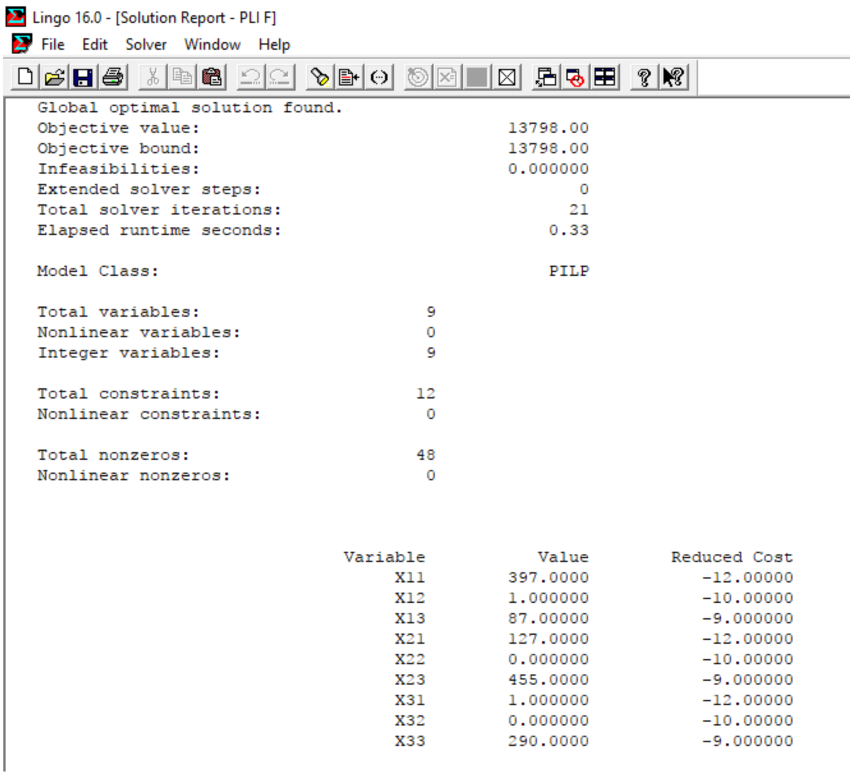
No LINGO:

```
MAX=12*x11+10*x12+9*x13+12*x21+10*x22+9*x23+12*x31+10*x32+9*x33;

x11+x12+x13<=500;
x21+x22+x23<=600;
x31+x32+x33<=300;
20*x11+15*x12+12*x13<=9000;
20*x21+15*x22+12*x23<=8000;
20*x31+15*x32+12*x33<=3500;
x11+x12+x13<=600;
x21+x22+x23<=800;
x31+x32+x33<=500;
6*x11+6*x12+6*x13=5*x21+5*x22+5*x23;
6*x11+6*x12+6*x13=10*x31+10*x32+10*x33;

@GIN(x11);
```

```
@GIN(x12);
@GIN(x13);
@GIN(x21);
@GIN(x22);
@GIN(x23);
@GIN(x31);
@GIN(x32);
@GIN(x33);
end
```

No LINGO o comando *@GIN()*, declara que a variável terá que ser inteira, deste modo, obteremos os valores respectivos a função-objetivo e os valores inteiros das variáveis:

Figura 17 – Solução da Função-Objetivo



Fonte – Própria

Nossa solução ótima será :

$$\begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{bmatrix} = \begin{bmatrix} 397 \\ 1 \\ 87 \\ 127 \\ 0 \\ 455 \\ 1 \\ 0 \\ 290 \end{bmatrix} = Max\ z = 13798$$

5.4 IMPLEMENTAÇÃO COMPUTACIONAL DO PROBLEMA DA MOCHILA

Neste trabalho foi implementado o algoritmo da mochila utilizando a Programação Dinâmica na linguagem C++ e executado em um notebook com 8Gb de memória RAM ddr3, processador *Intel(R) Core(TM)*i7-2620M 2.70GHz e S.O. Windows 10 64bits.

5.4.0.1 Programação Dinâmica

Considere uma mochila com capacidade $L = 10$ e 5 itens com seus pesos e valores dados pela tabela a seguir:

Tabela 23 – Peso e Valor

Item	1	2	3	4	5
Valor	110	60	70	20	18
Peso	8	5	6	3	2

Fonte – Própria

Através do algoritmo apresentado na seção 4.2.3.1, obtemos recursivamente a solução ótima para todas as capacidades de mochila, de 0 até $L = 10$. O valor ótimo encontrado é de $F(n,L) = F(5,10) = 128$ cujo vetor solução é $\alpha = (1,0,0,0,1)$. Todas as soluções obtidas pelo programa podem ser vistas na tabela a seguir.

Tabela 24 – Valores Obtidos na Solução

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1(8)	0	0	0	0	0	0	0	0	110	110	110
2(5)	0	0	0	0	0	60	60	60	110	110	110
3(6)	0	0	0	0	0	60	70	70	110	110	110
4(3)	0	0	0	20	20	60	70	70	110	110	110
5(2)	0	0	18	20	20	60	70	78	110	110	128

Fonte – Própria

Desse modo, encerramos o estudo dos problemas PPI.

6 CONCLUSÃO

Neste trabalho foi estudado os problemas de otimização combinatória, com enfoque no uso das ferramentas computacionais, tais como GeoGebra, LINGO e a linguagem de programação C++. Percebemos que o uso dessas ferramentas auxiliam em um melhor aproveitamento dos dados e das soluções.

Utilizamos modelos matemáticos propostos por (MARTELLO; TOTH, 1990) e nossos modelos de resolução dos problemas de otimização linear, foram propostos por (SILVA, 2006), (GOLDBARG; LUNA, 2005), (GILMORE; GOMORY, 1961),(CARVALHO, 2015). Nesses métodos, buscamos formalizar e simplificar as estratégias utilizadas para resolver os PPL. Buscamos associar o uso da teoria com a prática e a utilização das ferramentas como objeto auxiliadores para obtermos as melhores soluções.

Os métodos de solução foram testados e avaliados, de modo que se pode afirmar sobre a otimalidade ou quase-otimalidade de um número de exemplares resolvidos. A partir desses problemas pudemos comprovar a eficiência dos métodos propostos, seja ele com relação da qualidade das respostas obtidas, como também com o tempo computacional utilizado. A implementação destes métodos, envolveu aspectos computacionais, como o uso de softwares acessíveis para resolução destes problemas.

Notamos que a utilização dos softwares, é de total importância para a eficiência dos métodos de solução. Deste modo, buscamos utilizar ferramentas acessíveis e de fácil manuseio, para que o leitor possa se familiarizar com as ferramentas e utilizar em futuros trabalhos, sendo um exemplo de aplicação o uso do GeoGebra em sala de aula, para trabalhar soluções dos problemas de programação linear.

Em resumo, a contribuição deste trabalho foi orientada para o caso unidimensional, motivado pelos problemas de otimização combinatória linear, inteira e continua, abrindo futuramente, possibilidades de identificar novos problemas que apresentem estruturas semelhares e possam se beneficiar das ideias aqui desenvolvidas. Portanto, mostrando a importância e eficacia de se utilizar ferramentas computacionais para a resoluções dos problemas de programação linear.

REFERÊNCIAS

ARENALES, M. et al. *Pesquisa Operacional*. Rio de Janeiro: Elsevier, 2011. Citado na página 19.

BASSANEZI, R. C. *Ensino-aprendizagem com Modelagem Matemática*. São Paulo: Contexto, 2002. Citado 3 vezes nas páginas 17, 18 e 19.

BAZARRA, M. S.; JARVIS, J. J.; SHERALI, H. D. *Linear Programming and Network Flows*. 4. ed. New York: John Wiley & Sons, 2009. Citado 2 vezes nas páginas 29 e 54.

CARVALHO, R. *Problema da Mochila*. Campinas, 2015. Citado 2 vezes nas páginas 39 e 60.

CERQUEIRA, G. R. de L. *Heurísticas baseadas em geração Sequencial de Padrões para o Problema de Corte de Estoque Unidimensional*. Tese (Doutorado) — INPE, São José dos Campos, 2009. Citado na página 35.

DANTZIG, G. B. *Linear Programming and Extensions*. Princeton university press. Princeton: Elsevier, 1963. Citado 2 vezes nas páginas 12 e 22.

GARCIA, S.; GUERREIRO, R.; CORRAR, J. L. In: *Teoria das restrições e programação linear*: V congresso internacional de custos-costos, productividad y rentabilidad. [S.l.: s.n.], 1997. Citado na página 14.

GEOGEBRA. <https://www.geogebra.org/?lang=pt_BR>. Accessed: 2017-12-20. Citado na página 42.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. *Operations research*, INFORMS, v. 9, n. 6, p. 849–859, 1961. Citado 2 vezes nas páginas 40 e 60.

GOLDBARG, M. C.; LUNA, H. P. L. *Otimização Combinatória e Programação Linear*: Modelos e algoritmos. 2. ed.. ed. Rio de Janeiro: Elsevier, 2005. Citado 6 vezes nas páginas 16, 19, 30, 31, 54 e 60.

GRIFFIN, C. *Linear Programming*: Penn state math-lecture notes. Pennsylvania: [s.n.], 2014. Citado 2 vezes nas páginas 13 e 55.

JÚNIOR, A. de C. G.; SOUZA, M. J. F. *LINDO: Manual de Referência*. [S.l.], 2004. Citado na página 42.

LINGO. <https://www.lindo.com/index.php/ls-downloads>. Accessed: 2017-12-20. Citado 2 vezes nas páginas 14 e 48.

MARTELLO, S.; TOTH, P. *Knapsack problems*: algorithms and computer implementations. Chichester: John Wiley & Sons, 1990. Citado 4 vezes nas páginas 34, 35, 38 e 60.

NETO, L. L. de S. Tópicos de pesquisa operacional para o ensino médio. In: AEDB. *Simpósio de Excelência em Gestão Tecnológica*. Resende-RJ, 2009. Citado 2 vezes nas páginas 20 e 30.

PISINGER, D. *Algorithms for Knapsack Problems*. Tese (Doutorado) — DIKU, University of Copenhagen, Denmark, 1995. Technical Report 95-1. Citado na página 38.

SANTOS, M. P. dos. Programação linear. 2000. Citado 3 vezes nas páginas 52, 53 e 55.

SILVA, A. C. M. sa. *Investigação Operacional*: Programação linear. Rio de Janeiro: [s.n.], 2006. Citado 4 vezes nas páginas 22, 28, 29 e 60.

WEBER, R. W. *FERRAMENTA PARA EXECUÇÃO DE SCRIPTS OPL PARA OTIMIZAÇÃO COMBINATÓRIA EM GPU*. 2013. Monografia (Ciência da Computação), FURB (Universidade Regional de Blumenau), BLUMENAU, Brazil. Citado na página 42.

YANASSE, H. H.; SOMA, N. Y. A new enumeration scheme for the knapsack problem. *Discrete applied mathematics*, Elsevier, v. 18, n. 2, p. 235–245, 1987. Citado na página 38.

ZACHI, J. M. *Problemas de Programação Linear: uma proposta de resolução geométrica para o Ensino Médio com o uso do GeoGebra*. Rio Claro-SP: [s.n.], 2016. Citado na página 42.