UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

CAMILA FABRICIO KERKHOFF, CATALINA JARAMILLO VILLALBA, SERGIO KARDEC SOARES BATISTA

Relatório do Projeto Computacional

Campinas

2021

Camila Fabricio Kerkhoff, Catalina Jaramillo Villalba, Sergio Kardec Soares Batista

Relatório do Projeto Computacional

Relatório apresentado ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos parciais da Disciplina MT703 Programação Inteira da Professora Kelly Cristina Poldi.

:

Campinas

2021

Sumário

1 Introdução

Problemas de Otimização Linear são frequentemente utilizados em diferentes aplicações, abrangem diversas áreas, como, engenharias. São situações reais modeladas matematicamente, com o objetivo de minimizar ou maximizar os valores de uma função. Essa função, chamada de função objetivo, normalmente atende a algumas restrições, que são impostas ao problema para cada modelagem.

A Pesquisa Operacional é uma área do conhecimento que está associada a solução de problemas de tomada de decisão sendo possível desenvolver modelos matemáticos, que buscam soluções para diversos tipos de problemas. Neste contexto, a Pesquisa Operacional pode ser utilizada para a solução do problema de transporte.

Esse problema dentro da Pesquisa Operacional é dito como problema de transporte, tem como objetivo transportar um objeto de uma origem a um destino final, no caso, transportar mercadorias em caixas com um tamanho definido, de uma empresa do setor de confecção da cidade de Catalão, GO. Para enviar essas caixas, é necessário escolher dentre cinco transportadoras (A, B, $C1$, $C2$ e D), e cada transportadora tem um custo diferente para realizar o envio das caixas. Para realizar a modelagem desse problema é necessário empregar um método de solução para obter a solução ótima, neste caso, o problema é modelado com Programação Linear Inteira.

Para Problemas de Programação Linear Inteira, existem diversos métodos de resolução, tais como, *branch-and-bound*, algoritmo de plano de cortes e o *branch-and-cut*.

Para resolver o modelo matemático deste problema, foi utilizado os *software Python*, com a interface *Gurobipy* e o *Solver*, que é um suplemento do *Microsoft Excel*.

O presente documento tem como objetivo apresentar os resultados computacionais obtidos durante a implementação de um problema da vida real aplicando Programação Linear Inteira.

A primeira parte do documento faz referência a definição da Pesquisa Operacional, dividindo em Programação Linear e Programação Linear Inteira. A segunda parte define o Modelo Matemático do problema e na seguinte parte, tem a solução implementada no *Python* e *Excel*, incluindo uma análise de sensibilidade. E finalmente, temos as considerações finais e o modelo implementado nos anexos.

2 Pesquisa Operacional

2.1 Programação linear

Um problema de Programação Linear (PL) utiliza variáveis contínuas e apresenta comportamento linear, tanto em relação às restrições como à função objetivo (GOLDBARG; LUNA, 2005).

Um mesmo modelo de PL pode ser reescrito (LUENBERGER; YE, 2015), sem perda de suas propriedades matemáticas, utilizando a mudança no critério de otimização:

$$\text{Maximizar } \ (f\,(x)) \ \text{ corresponde a Minimizar } \ (-f\,(x)) \ \text{ e}$$
$$\text{Minimizar}\,(f\,(x)) \ \text{ corresponde a Maximizar } \ (-f\,(x))\,.$$

Problemas de PL podem ser descritos da seguinte forma:

$$\begin{gathered}
\text{Otimizar: } Z = \sum_{j=1}^{n} c_j^t x_j \\
\text{sujeito a} \\
\sum_{j=1}^{n} a_{ij} x_j \geqslant b_i \quad i = 1, 2, \ldots, p \\
\sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = p+1, p+2, \ldots, m \\
x_j \geqslant 0 \quad j = 1, 2, \ldots, n,
\end{gathered}$$

onde:

Z é a função objetivo;

$x = (x_j) \in R^n$ são as variáveis do problema;

$A \in R^{m \times n}$ é a matriz de restrições do problema;

$c^t = (c_j)^t \in R^n$ é o vetor de custos;

$b = (b_j) \in R^n$.

Para aplicação dos métodos de solução se faz necessário deixar o problema de Programação Linear modelado em uma forma denominada forma padrão. Para isso serão introduzidas as variáveis de folga e excesso:

Se o problema original apresenta uma restrição de desigualdade do tipo

$$a_j^t x \geqslant b_j,$$

ou do tipo

$$a_j^t x \leqslant b_j,$$

onde a_j^t é uma linha da matriz A, a restrição de desigualdade pode ser substituída por uma restrição de igualdade introduzindo-se uma variável adicional x_{n+1}, não-negativa, conhecida como variável de excesso quando a desigualdade é do tipo $\geqslant$, e variável de folga quando a desigualdade é do tipo $\leqslant$ (LUENBERGER; YE, 2015). Assim, o problema fica:

$$a_j^t x - x_{n+1} = b_j, \text{ introduzindo a variável de excesso, e}$$
$$a_j^t x + x_{n+1} = b_j, \text{ introduzindo a variável de folga.}$$

Podem ser adicionadas tantas variáveis de excesso e/ou de folga quantas forem as restrições de desigualdade presentes no modelo original.

A forma padrão exige variáveis não-negativas. Quando ocorrer da formulação original do problema apresentar uma ou mais variáveis de decisão irrestritas ou livres, é necessário reescrever cada variável livre x_j como a diferença de duas variáveis não-negativas, x_{j1} e x_{j2}, ou seja

$$x_j = x_{j1} - x_{j2}.$$

A ideia é que qualquer quantidade (positiva, nula ou negativa) possa ser representada como a diferença de duas quantidades não-negativas, que serão incorporadas à lista de variáveis de decisão do problema.

O problema acrescido de variáveis de folga, excesso e com variáveis livres colocado na forma padrão fica:

$$\begin{array}{ll} \text{otimizar} & c^t x \\ \text{sujeito a} & A'x' + Is' = b' \\ & A''x'' - Is'' = b'' \\ & x' \text{ e } x'' \geqslant 0. \end{array}$$

Além disso, pode ocorrer ainda que as variáveis apareçam canalizadas, ou seja,

$$l \leqslant x \leqslant u.$$

Nesses casos, pode ser feita a seguinte mudança de variáveis:

$$\tilde{x} = x - l \Rightarrow x = \tilde{x} + l$$
$$\tilde{u} = u - l \Rightarrow u = \tilde{u} + l.$$

Essa mudança tem por objetivo anular o limite inferior de x. Acrescentando uma variável de folga, o problema se torna um problema de otimização linear na forma padrão com limite superior nas variáveis:

$$\begin{array}{rc} \text{otimizar} & c^t x \\ \text{sujeito a} & Ax = b \\ & x \geqslant 0. \end{array}$$

2.2 Programação Linear Inteira

Um problema de Programação Linear Inteira (PLI) são aqueles em que a função objetivo e as restrições são lineares, contudo, possuem uma ou mais variáveis de decisão com valores inteiros, que limita o conjunto de soluções (ARENALES et al., 2007). Matematicamente, pode ser descrito como:

$$\begin{array}{rc} \text{otimizar} & c^t x \\ \text{sujeito a} & Ax = b \\ & x \in Z_n^+. \end{array}$$

Quando o problema tem variáveis reais e inteiras, o problema é denotado de Programação Inteira Mista (PIM),

$$\begin{array}{rc} \text{otimizar} & c^t x \\ \text{sujeito a} & Ax + Dy = b \\ & x \in R_+^n \in Z_+^p. \end{array}$$

Se as variáveis tomam valores $0-1$, denotamos como problema de Programação Binário (PB),

$$\begin{array}{rc} \text{otimizar} & c^t x \\ \text{sujeito a} & Ax = b \\ & x \in B^n, \end{array}$$

em que B^n representa o espaço n dimensional binário.

Para resolver um problema de Programação Inteira (PI) é necessário aproximar os valores encontrados para o inteiro mais próximo, contudo, se os valores forem pequenos, esse procedimento é inviável, logo, nem sempre aproximar a solução do PL encontra a solução ótima do PI. Portanto, para a solução do problema, é necessário adicionar outros métodos que apresentam melhor desempenho para problemas inteiros, como o *branch-and-bound*, algoritmo de plano de cortes e o *branch-and-cut*, que é uma combinação dos dois primeiros métodos (ARENALES et al., 2007). Todos esses métodos estão baseados na enumeração implícita.

Esses tipos de problemas de Programação Linear são aplicados na vida real, tais como a atribuição de tarefa, roteamento, transporte, finança, produção, logística, humanitária, entre outros.

3 Modelagem do Problema

A logística consiste em organizar o fluxo de mercancia, energia ou informação com o objetivo de gerenciar a estratégia de aquisição, armazenamento de produtos e controle de estoque. Tem como objetivo otimizar os custos e gastos da logística de cada empresa, sendo importante utilizar a Pesquisa Operacional como ferramenta facilitadora para a tomada de decisão.

Na sequência, será apresentado o estudo de caso da otimização do custo de transporte de uma empresa do setor de confecção da cidade de Catalão, GO. O envio dos produtos dessa empresa são realizados por meio de empresas externas, e a seleção do melhor transportador foi feita intuitivamente pelo operador responsável.

A figura abaixo representa a logística que a empresa de confecção precisa otimizar para reduzir os custos com transporte. As esferas laranjas representam as transportadoras, nomeadas como A, B, C1, C2 e D; as esferas azuis representam as cidades destino, nomeadas como SP (São Paulo), LJ (Laranjeira), SB (São Bernardo), SA (Santo André) e SJ (São José dos Campos).

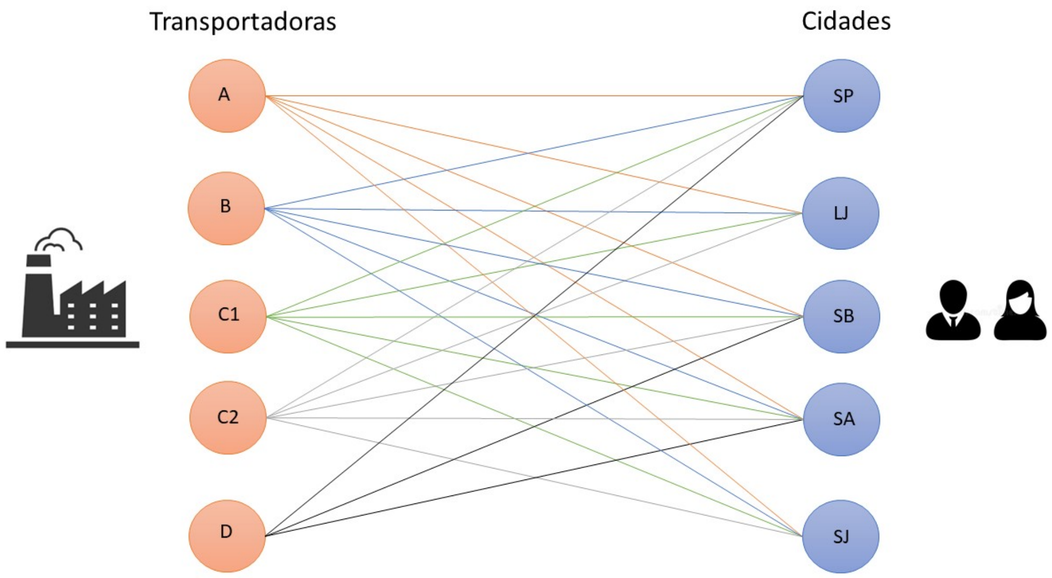

Figura 1 – Estudo de caso.

Os custos tem como base um padrão de caixa, que possui 70 cm de largura, 50 cm de comprimento, 50 cm de altura e pesa no máximo 12kg.

A seguinte tabela define os custos de transporte de cada caixa para cada cidade de destino, de acordo com cada transportadora.

Custo de transporte					
	Transportadora				
Cidade	A	B	C1	C2	D
São Paulo	$49,00$	$48,90$	$49,92$	$117,83$	$35,00$
Laranjeira	$56,00$	$55,72$	$49,92$	$132,97$	-
São Bernardo	$50,00$	$49,31$	$49,92$	$117,83$	$35,00$
Santo André	$52,00$	$55,08$	$49,92$	$117,83$	$35,00$
São José dos Campos	$46,00$	$44,00$	$49,92$	$117,83$	-

Tabela 1 – Custos de transporte para cada cidade.

O custo total gerado para transportar as mercadorias da empresa de confecção, durante seis meses, foi de R$$7692,81$.

Mês	Custo total
1	1288,22
2	1599,88
3	1439,82
4	908,53
5	1084,40
6	1371,96

Tabela 2 – Custo por mês.

Além dos custos, temos em cada mês a demanda de caixas a ser enviadas para cada cidade.

Demanda						
Mês	1	2	3	4	5	6
São Paulo	4	5	4	3	4	5
Laranjeira	8	10	6	6	7	10
São Bernardo	2	3	3	2	4	4
Santo André	6	8	4	4	5	8
São José dos Campos	2	1	2	1	3	3

Tabela 3 – Demanda para cada cidade.

Vamos supor que cada empresa transportadora pode enviar até dez caixas por cada envio.

Defina as variáveis de decisão:

x_{ij} : número de caixas enviadas a cidade i pela transportadora j.

O número de variáveis é definido de acordo a tabela 1.

Seja c_{ij} o custo de enviar para cidade i pela transportadora j e d_ik a demanda da cidade i no mês k. Então, o problema de programação inteira para cada mês k é dado por,

$$\begin{aligned} &\min \sum_{i,j} c_{ij} x_{ij} \\ s.a.\ & x_{ij} \leqslant 10 \quad \text{(capacidade das transportadoras)}, \\ & \sum_{j} x_{ij} = d_{ik}, \forall i \quad \text{(demanda)} \\ & x_{ij} \in Z_+, \forall i, j. \end{aligned}$$

3.1 Modelo Matemático

O modelo matemático tem o objetivo tratar o problema com a função de otimizar os custos do transporte, ou seja, tem por finalidade minimizar os custos de envio das mercadorias, o modelo utiliza as ferramentas da Programação Linear Inteira para sua resolução.

3.1.1 Variáveis de Decisão

Nas variáveis de decisão, as funções tem por objetivo determinar a quantidade de caixas que serão enviadas por determinada transportadora para cada cidade de destino, e são elas:

x_1 : número de caixas enviadas a cidade *SP* pela transportadora *A*

x_2 : número de caixas enviadas a cidade *SP* pela transportadora *B*

x_3 : número de caixas enviadas a cidade *SP* pela transportadora *C1*

x_4 : número de caixas enviadas a cidade *SP* pela transportadora *C2*

x_5 : número de caixas enviadas a cidade *SP* pela transportadora *D*

x_6 : número de caixas enviadas a cidade *LI* pela transportadora *A*

x_7 : número de caixas enviadas a cidade *LI* pela transportadora *B*

x_8 : número de caixas enviadas a cidade *LI* pela transportadora *C1*

x_9 : número de caixas enviadas a cidade *LI* pela transportadora *C2*

x_{10} : número de caixas enviadas a cidade *LI* pela transportadora *D*

x_{11} : número de caixas enviadas a cidade *SB* pela transportadora *A*

x_{12} : número de caixas enviadas a cidade *SB* pela transportadora *B*

x_{13} : número de caixas enviadas a cidade *SB* pela transportadora *C1*

x_{14} : número de caixas enviadas a cidade *SB* pela transportadora *C2*

x_{15} : número de caixas enviadas a cidade *SB* pela transportadora *D*

x_{16} : número de caixas enviadas a cidade *SA* pela transportadora *A*

x_{17} : número de caixas enviadas a cidade *SA* pela transportadora *B*

x_{18} : número de caixas enviadas a cidade *SA* pela transportadora *C1*

x_{19} : número de caixas enviadas a cidade *SA* pela transportadora *C2*

x_{20} : número de caixas enviadas a cidade *SA* pela transportadora *D*

x_{21} : número de caixas enviadas a cidade *SJ* pela transportadora *A*

x_{22} : número de caixas enviadas a cidade *SJ* pela transportadora *B*

x_{23} : número de caixas enviadas a cidade *SJ* pela transportadora *C1*

x_{24} : número de caixas enviadas a cidade *SJ* pela transportadora *C2*

x_{25} : número de caixas enviadas a cidade *SJ* pela transportadora *D*

Variando de x_1 a x_{25}, as variáveis de decisão representam o quantitativo de caixas que serão enviadas por cada transportadora. Então, irá interferir na solução do problema, visto que, o custo de cada caixa se altera para cada cidade destino diferente.

3.1.2 Função Objetivo

Para este problema, a função objetivo tem como finalidade minimizar o custo do transporte.

Minimizar: $Z = 49,00x_1 + 48,90x_2 + 49,92x_3 + 117,83x_4 + 35,00x_5 + 56,00x_6 + 55,72x_7 + 49,92x_8 + 132,97x_9 + 0,00x_{10} + 50,00x_{11} + 49,31x_{12} + 49,92x_{13} + 117,83x_{14} + 35,00x_{15} + 52,00x_{16} + 55,08x_{17} + 49,92x_{18} + 117,83x_{19} + 35,00x_{20} + 46,00x_{21} + 44,00x_{22} + 49,92x_{23} + 117,83x_{24} + 0,00x_{25}$

3.1.3 Restrições

É necessário inserir restrições neste modelo matemático, para limitar o número máximo de caixas que cada transportadora pode enviar. Desse modo, é definido que cada transportadora pode enviar no máximo dez caixas.

Outra restrição que se faz necessário inserir no problema é referente à transportadora *D*, que não tem frota para as cidades *LJ* e *SJ*, logo, o número de caixas para esse trecho será igual a zero.

Além destas restrições, é definido as restrições para a quantidade das caixas que cada transportadora irá enviar, de acordo com o a quantidade de caixas que a empresa necessita para transportar suas mercadorias para cada cidade de destino.

Ambas restrições pertencem aos números não-negativos e, por não ter como enviar a metade de uma caixa, que possui um padrão determinado anteriormente, então, as restrições devem pertencer aos números inteiros, e são representadas por:

$x_1 \leqslant 10$	Número máx. de caixas da transportadora *A* para *SP*
$x_2 \leqslant 10$	Número máx. de caixas da transportadora *B* para *SP*
$x_3 \leqslant 10$	Número máx. de caixas da transportadora *C1* para *SP*
$x_4 \leqslant 10$	Número máx. de caixas da transportadora *C2* para *SP*
$x_5 \leqslant 10$	Número máx. de caixas da transportadora *D* para *SP*
$x_6 \leqslant 10$	Número máx. de caixas da transportadora *A* para *LJ*
$x_7 \leqslant 10$	Número máx. de caixas da transportadora *B* para *LJ*
$x_8 \leqslant 10$	Número máx. de caixas da transportadora *C1* para *LJ*
$x_9 \leqslant 10$	Número máx. de caixas da transportadora *C2* para *LJ*
$x_{10} = 0$	Número máx. de caixas da transportadora *D* para *LJ*
$x_{11} \leqslant 10$	Número máx. de caixas da transportadora *A* para *SB*
$x_{12} \leqslant 10$	Número máx. de caixas da transportadora *B* para *SB*
$x_{13} \leqslant 10$	Número máx. de caixas da transportadora *C1* para *SB*
$x_{14} \leqslant 10$	Número máx. de caixas da transportadora *C2* para *SB*
$x_{15} \leqslant 10$	Número máx. de caixas da transportadora *D* para *SB*
$x_{16} \leqslant 10$	Número máx. de caixas da transportadora *A* para *SA*
$x_{17} \leqslant 10$	Número máx. de caixas da transportadora *B* para *SA*
$x_{18} \leqslant 10$	Número máx. de caixas da transportadora *C1* para *SA*

$x_{19} \leqslant 10$ Número máx. de caixas da transportadora *C2* para *SA*

$x_{20} \leqslant 10$ Número máx. de caixas da transportadora *D* para *SA*

$x_{21} \leqslant 10$ Número máx. de caixas da transportadora *A* para *SJ*

$x_{22} \leqslant 10$ Número máx. de caixas da transportadora *B* para *SJ*

$x_{23} \leqslant 10$ Número máx. de caixas da transportadora *C1* para *SJ*

$x_{24} \leqslant 10$ Número máx. de caixas da transportadora *C2* para *SJ*

$x_{25} = 0$ Número máx. de caixas da transportadora *D* para *SJ*

$x_1 + x_2 + x_3 + x_4 + x_5 =$ Número de caixas que deseja enviar para *SP*

$x_6 + x_7 + x_8 + x_9 + x_{10} =$ Número de caixas que deseja enviar para *LJ*

$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} =$ Número de caixas que deseja enviar para *SB*

$x_{16} + x_{17} + x_{18} + x_{19} + x_{20} =$ Número de caixas que deseja enviar para *SA*

$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} =$ Número de caixas que deseja enviar para *SJ*

$x_i \geqslant 0, \quad i = 1, 2, \cdots, 25$

Finalizada a definição do modelo, o próximo passo é implementar computacionalmente para determinar a solução ótima do problema em questão.

4 Resultados Computacionais

Os *software* escolhidos para a solução do problema de transporte foram o *Python*, utilizando a interface *Gurobipy*, que resolve modelos de programação matemática simples e o *Solver*, que é um suplemento do *Microsoft Excel*, que pode ser utilizado para teste de hipóteses, para encontrar um valor ideal (máximo ou mínimo), conforme restrições, utilizando o valores inseridos na planilha.

4.1 Resultados *Python*

A implementação do algoritmo em *Python* pode ser vista nos anexos A.

Com o objetivo de minimizar os custos e levando em consideração as restrições que foram descritas na capítulo anterior 3, que levam em consideração a capacidade de cada transportadora e o custo pelo transporte, o resultado obtido das transportadoras a serem utilizadas para o envio das caixas em cada mês são:

		Transportadora				
	Cidade	A	B	C1	C2	D
Mês 1	São Paulo		4			
	Laranjeiras			8		
	São Bernardo					2
	Santo André					6
	São José dos Campos		2			
Mês 2	São Paulo		5			
	Laranjeiras			10		
	São Bernardo					3
	Santo André					8
	São José dos Campos		1			
Mês 3	São Paulo		4			
	Laranjeiras			6		
	São Bernardo					3
	Santo André					4
	São José dos Campos		2			
Mês 4	São Paulo		3			
	Laranjeiras			6		
	São Bernardo					2
	Santo André					4
	São José dos Campos		1			
Mês 5	São Paulo		4			
	Laranjeiras			7		
	São Bernardo					4
	Santo André					5
	São José dos Campos		3			
Mês 6	São Paulo		5			
	Laranjeiras			10		
	São Bernardo					4
	Santo André					8
	São José dos Campos		3			

Tabela 4 – Distribuição das transportadoras por mês.

O custo total gerado para transportar as mercadorias da empresa de confecção, durante seis meses, foi de R$5951, 74.

Mês	Custo total
1	$962,96$
2	$1172,70$
3	$828,12$
4	$700,22$
5	$992,04$
6	$1295,70$

Tabela 5 – Custo total por mês.

Levando em consideração os custos unitários dados na tabela 1, a capacidade máxima dos transportadores (dez caixas) e os resultados finais obtidos nas tabelas 4 e 5 pode-se concluir que pelo fato da demanda de cada cidade ser menor que a capacidade máxima da transportadora, então a seleção é dada pelas empresas de menor custo.

4.1.1 Análise de Sensibilidade

Vamos supor agora que a capacidade máxima de cada transportadora é quatro caixas por envio. Portanto, a atribuição agora não será para uma única empresa. Os resultados são mostrados nas tabelas 6 e 7.

		Transportadora				
	Cidade	A	B	C1	C2	D
Mês 1	São Paulo		4			
	Laranjeiras		4	4		
	São Bernardo					2
	Santo André			2		4
	São José dos Campos		2			
Mês 2	São Paulo	1	4			
	Laranjeiras	2	4	4		
	São Bernardo					3
	Santo André			4		4
	São José dos Campos		1			
Mês 3	São Paulo		4			
	Laranjeiras		2	4		
	São Bernardo					3
	Santo André					4
	São José dos Campos		2			
Mês 4	São Paulo		3			
	Laranjeiras		2	4		
	São Bernardo					2
	Santo André					4
	São José dos Campos		1			
Mês 5	São Paulo		4			
	Laranjeiras		3	4		
	São Bernardo					4
	Santo André			1		4
	São José dos Campos		3			
Mês 6	São Paulo	1	4			
	Laranjeiras	2	4	4		
	São Bernardo					4
	Santo André			4		4
	São José dos Campos		3			

Tabela 6 – Distribuição das transportadoras por mês.

O custo total gerado para transportar as mercadorias da empresa de confecção, durante seis meses, foi de R$6250, 58.

Mês	Custo total
1	$1016,00$
2	$1267,84$
3	$839,72$
4	$711,82$
5	$1024,36$
6	$1390,84$

Tabela 7 – Custo total por mês.

Pelos resultados anteriores, observa-se então que a distribuição deixa de ser a mais simples. De acordo com a demanda, a atribuição deve ser feita em várias operadoras, por isso o custo aumenta. Percebe-se que a seleção das empresas se dá de acordo com o custo, atribuindo-se o maior número possível de caixas (até quatro) pela operadora que oferecer o menor custo.

Vamos supor agora que foi criada uma política empresarial, que estabelece que se a transportadora B é usada para enviar para São José dos Campos, ela é usada também para o transporte para São Bernardo com pelo menos uma caixa. Daí, a nova restrição é:

$$x_{52} \leqslant M x_{42},$$

em que M é um número grande, x_{52} e x_{42} são a quantidade de caixas enviadas pela transportadora B a São José dos Campos e Santo André respetivamente. Portanto, a nova solução ótima para o mês 1 teve um pequeno aumento, gerando o custo de R$1021,16 e a distribuição é dada na tabela 8.

		Transportadora				
	Cidade	A	B	C1	C2	D
Mês 1	São Paulo		4			
	Laranjeiras		4	4		
	São Bernardo					2
	Santo André		1	1		4
	São José dos Campos		2			

Tabela 8 – Distribuição de transportadoras.

Tendo em vista que o custo para Santo André pela empresa B é considerado alto em relação ao transportador D e *C1*, observa-se que a alocação mínima é feita por esta empresa para atender à nova restrição. Ademais, o custo incrementa-se em R$5,16.

4.2 Resultados *Solver*

Foi também implementado o mesmo problema no *Microsoft Excel*, usando a função *Solver*, que faz parte da aba "dados" na linha superior das opções de comando.

A implementação é dada pelo preenchimento das células da área livre da planilha do *Excel* com os valores dos coeficientes da função objetivo 3.1.2 e posteriormente preencher as células com os coeficientes das restrições 3.1.3, na mesma direção, os valores das respectivas capacidades das empresas em transportar as caixas, por último, as restrições de demanda que ao contrário das anteriores que são restrições de desigualdade (menor ou igual, $\leqslant$), as restrições de demanda devem ser de igualdade, ou seja, a quantidade de caixa a ser enviada a cidade *SP* deve ser exatamente igual a sua demanda, e menor que a capacidade da empresa de enviar, caso a capacidade da empresa em questão ser menor que a demanda, devemos escolher outra empresa. Nessa configuração temos a seguinte arrumação na planilha do *Excel*:

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25			
Função Objetivo	49	48,9	49,92	117,83	35	56	55,72	49,92	132,97	0	50	49,31	49,92	117,83	35	52	55,08	49,92	117,83	35	46	44	49,92	117,83	0	907,36		
Variaveis de Decisão	0	0	0	0	4	0	0	8	0	0	0	0	0	0	2	0	0	0	0	6	0	2	0	0	0	0		
Restrições	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	<=	10
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	<=	10
	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	0
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	6	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	<=	10
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	<=	0
																										4	=	4
																										8	=	8
																										2	=	2
																										6	=	6
																										2	=	2

Figura 2 – Tabela de implementação no *Excel*.

Na configuração da tabela acima, temos que a matriz principal (identidade 25×25) vezes o vetor $x = (x_1, .., x_25)^t$ deve ser menor ou igual ao vetor de capacidades dessas empresas de transportar cada trecho b, que é o seguinte vetor:

$b = (10, 10, 10, 10, 10, 10, 10, 10, 10, 0, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 0)^t$, perceba que as capacidades x_{10} e x_{25} são iguais a zero, pois as transportadoras não fazem tais transportes para as cidades respectivas dessas variáveis.

As restrições posteriores são as de demanda que estão abaixo da matriz identidade, para o mês de maio as demandas foram: $d = (4, 8, 2, 6, 2)^t$, dessa forma pedimos então que a soma das variáveis 5 a 5 sejam exatamente iguais a essas demandas, ou seja,

$$x_1 + x_2 + x_3 + x_4 + x_5 = \text{demanda para } SP = 4$$

$$x_6 + x_7 + x_8 + x_9 + x_{10} = \text{demanda para } LJ = 8$$

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = \text{demanda para } SB = 2$$

$$x_{16} + x_{17} + x_{18} + x_{19} + x_{20} = \text{demanda para } SA = 6$$

$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = \text{demanda para } SJ = 2$$

Próximo passo é pedir que as variáveis de decisão (x_i, $i = 1, \cdots, 25$) (segunda linha na tabela do *Excel*) sejam todos positivos e inteiros na função onde inserimos as restrições no *Solver* e pedimos para minimizar a função objetivo (pode ser encontrada multiplicando as variáveis de decisão vezes as células dos coeficientes da função objetivo) sujeita as restrições de demanda e capacidade das transportadoras.

Tais configurações nos dão que $x_5 = 4$, $x_8 = 8$, $x_{15} = 2$, $x_{20} = 6$, e $x_{22} = 2$, gerando um custo de R$907, 36 para o mês de maio, bem menor que o custo gerado para o mês de maio pela escolha intuitiva do responsável da empresa decidida inicialmente (custo de R$1288,22).

Para os meses subsequentes basta alterar as células de demanda para cada mês, por exemplo no mês de junho teve o vetor demanda de $d = (5, 10, 3, 8, 1)^t$, para as respectivas cidades *SP*, *LJ*, *SB*, *SA* e *SJ*, que inicialmente tiveram um custo de R$1599,88, mas com a escolha do *Solver* obtemos a redução do custo para R$1103,20.

Os resultados são mostrados na tabela 9, contendo a distribuição das transportadoras separadas por mês e na tabela 10, com o custo total por mês.

		Transportadora				
	Cidade	A	B	C1	C2	D
Mês 1	São Paulo					4
	Laranjeiras			8		
	São Bernardo					2
	Santo André					6
	São José dos Campos		2			
Mês 2	São Paulo					5
	Laranjeiras			10		
	São Bernardo					3
	Santo André					8
	São José dos Campos		1			
Mês 3	São Paulo					4
	Laranjeiras			6		
	São Bernardo					3
	Santo André					4
	São José dos Campos		2			
Mês 4	São Paulo					3
	Laranjeiras			6		
	São Bernardo					2
	Santo André					4
	São José dos Campos		1			
Mês 5	São Paulo					4
	Laranjeiras			7		
	São Bernardo					4
	Santo André					5
	São José dos Campos		3			
Mês 6	São Paulo					5
	Laranjeiras			10		
	São Bernardo					4
	Santo André					8
	São José dos Campos		3			

Tabela 9 – Distribuição das transportadoras por mês.

O custo total gerado para transportar as mercadorias da empresa de confecção, durante seis meses, foi de R$5604, 24.

Mês	Custo total
1	$907,36$
2	$1103,20$
3	$772,52$
4	$658,52$
5	$936,44$
6	$1226,20$

Tabela 10 – Custo total por mês.

O custo total gerado pela implementação no *Excel* foi menor que o gerado pela implementação no *Python*. Portanto, como o modelo matemático é para um problema de minimização de custos, o melhor resultado foi obtido pela modelagem no *Excel*.

4.2.1 Análise de Sensibilidade

Perceba que em nenhum dos casos a demanda foi superior a capacidade das empresas, fazendo com a as restrições de capacidade possam oscilar dentro dessa folga $f_i = capacidade_i - demanda$.

Como, por exemplo, o mês de julho teve o vetor demanda $d = (4, 6, 3, 4, 2)^t$ e a decisão ótima foi $x_5 = 4$, $x_8 = 6$, $x_{15} = 3$, $x_{20} = 4$, $x_2 2 = 2$, gerando um custo bem abaixo dos R$ 1.439,82 pago naquele mês sem a tomada de decisão ótima, e com custo de R$772,52 nessa configuração.

Como a escolha ótima foi $x_5 = 4$, $x_8 = 6$, $x_{15} = 3$, $x_{20} = 4$, $x_{22} = 2$ e $x_5 \leqslant 10, x_8 \leqslant 10, x_{15} \leqslant 10, x_{20} \leqslant 10, x_{22} \leqslant 10$, então, a capacidade das transportadoras suportaria a diminuição em até seis unidades para x_5, quatro unidades para x_8, sete unidades para x_{15}, seis unidades para x_{20} e oito unidades para x_{22}, e ainda assim a solução seria ótima, se manteria com o mesmo vetor de decisão ótimo e com o mesmo custo final ótimo. O mesmo raciocínio vale para os outros meses.

Podemos usar essa folga como uma opção para diminuir o preço, aproveitando trajetos semelhantes dependendo da localização geográfica da cidade em questão, afim de usar um mesmo frete atendendo demandas de cidades diferentes que por serem próximas podem ajudar na logística do transporte e consequentemente, na diminuição do valor do frete. Tais valores podem ser negociados com as transportadoras.

5 Considerações Finais

O objeto de estudo deste trabalho foi um problema de transporte, que teve por objetivo minimizar os custos de logística de uma empresa do setor de confecção da cidade de Catalão, GO.

Para solução do problema de transporte, foi necessário analisar os dados e identificar que por se tratar de um problema de Programação Linear, também se fez necessário tratar o problema com variáveis inteiras, logo, um problema de Programação Linear Inteira.

Para automatizar a construção e solucionar o problema, o modelo matemático de Programação Linear foi colocado na forma padrão e posteriormente, desenvolvido o modelo no *software Python*, com a interface *Gurobipy* e no *Solver*, suplemento do *Microsoft Excel*.

O modelo implementado obteve soluções satisfatórias, além de solucionar o problema, diminuiu os custos que a empresa gastava com a logística anterior, feita intuitivamente pelo operador responsável da empresa de confecção. Na comparação entre os dois programas utilizados, temos como menor custo total do *Software Python* o valor de R$5951, 74, para o *Solver* do *Excel* o valor de R$5604, 24, uma diminuição significativa se comparado ao valor gasto inicialmente pela empresa de R$7692, 81. Portanto, o melhor resultado foi obtido com a modelagem no *Solver* do *Excel*.

Assim, o modelo proposto soluciona o problema em questão, sendo possível também modificações referentes a alterações cotidianas da demanda da empresa de confecção.

Referências

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional*. Rio de Janeiro: Elsevier, 2007. Citado 2 vezes nas páginas 7 e 8.

GOLDBARG, M.; LUNA, H. P. L. *Otimização Combinatória e Programação Linear*. 2.ed.. ed. Rio de Janeiro: Elsevier, 2005. Citado na página 5.

LUENBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3319188410. Citado 2 vezes nas páginas 5 e 6.

Anexos

ANEXO A – Modelagem do problema em *Python*

Projeto final Programação inteira

Camila Fabricio Kerkhoff

Catalina Jaramillo Villalba

Sergio Kardec Soares Batista

In [1]:
```
import gurobipy as gp
from gurobipy import GRB
```

In [2]:
```
# Parameters

# Transportation cost to Sao Paulo (SJ), Laranjeiras (LJ), Sao Bernardo (SB),
 Santo Andre (SA) e Sao Jose dos Campos (SJ)

cost = {'SP':{'A': 49,'B':48.9,'C1':49.92,'C2':117.83,'D':117.83},
        'LJ':{'A':56,'B':55.72,'C1':49.92,'C2':132.97},
        'SB':{'A':50,'B':49.31,'C1':49.92,'C2':117.83,'D':35},
        'SA':{'A':52,'B':55.08,'C1':49.92,'C2':117.83,'D':35},
        'SJ': {'A':46,'B':44,'C1':49.92,'C2':117.83}
        }

# Demanda de cada cidade

demand_SP = [4,5,4,3,4,5]
demand_LJ = [8,10,6,6,7,10]
demand_SB = [2,3,3,2,4,4]
demand_SA = [6,8,4,4,5,8]
demand_SJ = [2,1,2,1,3,3]

cities = list(cost.keys())
```

In [3]:
```
# Model for month 1

month = 1

model = gp.Model('Transportation cost')

x_SP = model.addVars(range(1,len(cost['SP'])+1),vtype=gp.GRB.INTEGER, name="x_
SP")
x_LJ = model.addVars(range(1,len(cost['LJ'])+1),vtype=gp.GRB.INTEGER, name="x_
LJ")
x_SB = model.addVars(range(1,len(cost['SB'])+1),vtype=gp.GRB.INTEGER, name="x_
SB")
x_SA = model.addVars(range(1,len(cost['SA'])+1),vtype=gp.GRB.INTEGER, name="x_
SA")
x_SJ = model.addVars(range(1,len(cost['SJ'])+1),vtype=gp.GRB.INTEGER, name="x_
SJ")
```

```
Restricted license - for non-production use only - expires 2022-01-13
```

In [4]:
```
# Each company can ship cap[j] boxes to city j.

cap = [10,10,10,10,10]

capacity_SP = model.addConstrs((x_SP[j] <= cap[0] for j in range(1,len(cost['S
P'])+1)), name='capacity to SP')
capacity_LJ = model.addConstrs((x_LJ[j] <= cap[1] for j in range(1,len(cost['L
J'])+1)), name='capacity to LJ')
capacity_SB = model.addConstrs((x_SB[j] <= cap[2] for j in range(1,len(cost['S
B'])+1)), name='capacity to SB')
capacity_SA = model.addConstrs((x_SA[j] <= cap[3] for j in range(1,len(cost['S
A'])+1)), name='capacity to SA')
capacity_SJ = model.addConstrs((x_SJ[j] <= cap[4] for j in range(1,len(cost['S
J'])+1)), name='capacity to SJ')
```

In [5]:
```
# Satisfy demand

c_demand_SP = model.addConstr(x_SP.sum() == demand_SP[month-1])
c_demand_LJ = model.addConstr(x_LJ.sum() == demand_LJ[month-1])
c_demand_SB = model.addConstr(x_SB.sum() == demand_SB[month-1])
c_demand_SA = model.addConstr(x_SA.sum() == demand_SA[month-1])
c_demand_SJ = model.addConstr(x_SJ.sum() == demand_SJ[month-1])
```

In [6]:
```
# Minimize cost

model.setObjective(gp.quicksum(list(cost['SP'].values())[j-1]*x_SP[j] for j in
range(1,len(cost['SP'])+1))
                   + gp.quicksum(list(cost['LJ'].values())[j-1]*x_LJ[j] for j
in range(1,len(cost['LJ'])+1))
                   +gp.quicksum(list(cost['SB'].values())[j-1]*x_SB[j] for j i
n range(1,len(cost['SB'])+1))
                   + gp.quicksum(list(cost['SA'].values())[j-1]*x_SA[j] for j
in range(1,len(cost['SA'])+1))
                   + gp.quicksum(list(cost['SJ'].values())[j-1]*x_SJ[j] for j
in range(1,len(cost['SJ'])+1)), GRB.MINIMIZE)
```

In [7]:
```
model.optimize()
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 28 rows, 23 columns and 46 nonzeros
Model fingerprint: 0x496c889b
Variable types: 0 continuous, 23 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [4e+01, 1e+02]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+00, 1e+01]
Found heuristic solution: objective 1095.2000000
Presolve removed 28 rows and 23 columns
Presolve time: 0.02s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.05 seconds
Thread count was 1 (of 8 available processors)

Solution count 2: 962.96 1095.2

Optimal solution found (tolerance 1.00e-04)
Best objective 9.629600000000e+02, best bound 9.629600000000e+02, gap 0.0000%
```

In [8]:
```
for v in model.getVars():
    if v.x > 1e-6:
        print(v.varName,'=',v.x,'. Enviar a ', v.varName[2:4], v.x, 'caixas pe
la empresa transportadora',list(cost[v.varName[2:4]].keys())[int(v.varName[-2
])-1])

# Display optimal total matching score
print('Costo total de transporte no mes ',month, ':', model.objVal)
```

```
x_SP[2] = 4.0 . Enviar a  SP 4.0 caixas pela empresa transportadora B
x_LJ[3] = 8.0 . Enviar a  LJ 8.0 caixas pela empresa transportadora C1
x_SB[5] = 2.0 . Enviar a  SB 2.0 caixas pela empresa transportadora D
x_SA[5] = 6.0 . Enviar a  SA 6.0 caixas pela empresa transportadora D
x_SJ[2] = 2.0 . Enviar a  SJ 2.0 caixas pela empresa transportadora B
Costo total de transporte no mes  1 : 962.96
```