Trabalho de Pesquisa Operacional

12.10 - Descentralização

Gabriel Davila - 171870

Gabriel Yamato Freitas - 172555

Guilherme Lima Correa - 173811

Leonardo Lima – 178549

Introdução

Um grande desafio das companhias atuais é atuar em mercados globalmente competitivos, com excelência em produtos e serviços oferecidos. Em busca dessa excelência, as companhias buscam reduzir custos. E nesse cenário, a pesquisa operacional se faz importante.

Este trabalho apresenta uma aplicação semelhante do modelo conhecido na literatura como problema de designação quadrática e é mais bem representado por meio de layout de facilidades (ARENALES, 2007). Os problemas de Layout possuem incontáveis aplicações práticas na indústria e podem ser caracterizados pelo arranjo físico de facilidades ao longo de determinada áreas, formando um Layout. Facilidades podem ser máquinas, centros de trabalhos, células de manufatura, departamentos de um edifício, armazéns, entre outras. Definir o arranjo físico é decidir onde colocar todas as facilidades, para que haja um "fluxo harmonioso" dos recursos, reduzindo, dessa forma, os custos.

Portanto, o objetivo do problema é justamente decidir a localização física de cada facilidade no layout de modo a minimizar uma função de custo, que pode ser calculada com base em diversos dados como, por exemplo, as distâncias entre as facilidades e os fluxos entre as mesmas, sejam eles de pessoas ou materiais.

Problema de modo estruturado

O problema desenvolvido pelo grupo é o 12.10. Este trata de uma descentralização, ou seja, como dispersar os departamentos da capital para outras cidades e se isso é, ou não é vantajoso para a empresa.

O problema é contextualizado por meio de uma companhia que deseja transferir seus departamentos para fora de Londres. Existem diversos benefícios, para se realizar essa mudança, concedidos pela "cidade destino" tanto diretamente como indiretamente, por meio de incentivos fiscais e maior facilidade de recrutamento. Esses benefícios foram cotados para todas as possíveis localizações de cada departamento.

Tabela 1: Benefícios (em milhares de libras por ano)			
	Bristol	**Brighton**	**Londres**
A	10	10	0
B	15	20	0
C	10	15	0
D	20	15	0
E	5	15	0

O problema consiste em determinar onde cada departamento deve ser alocado para minimizar o custo total anual.

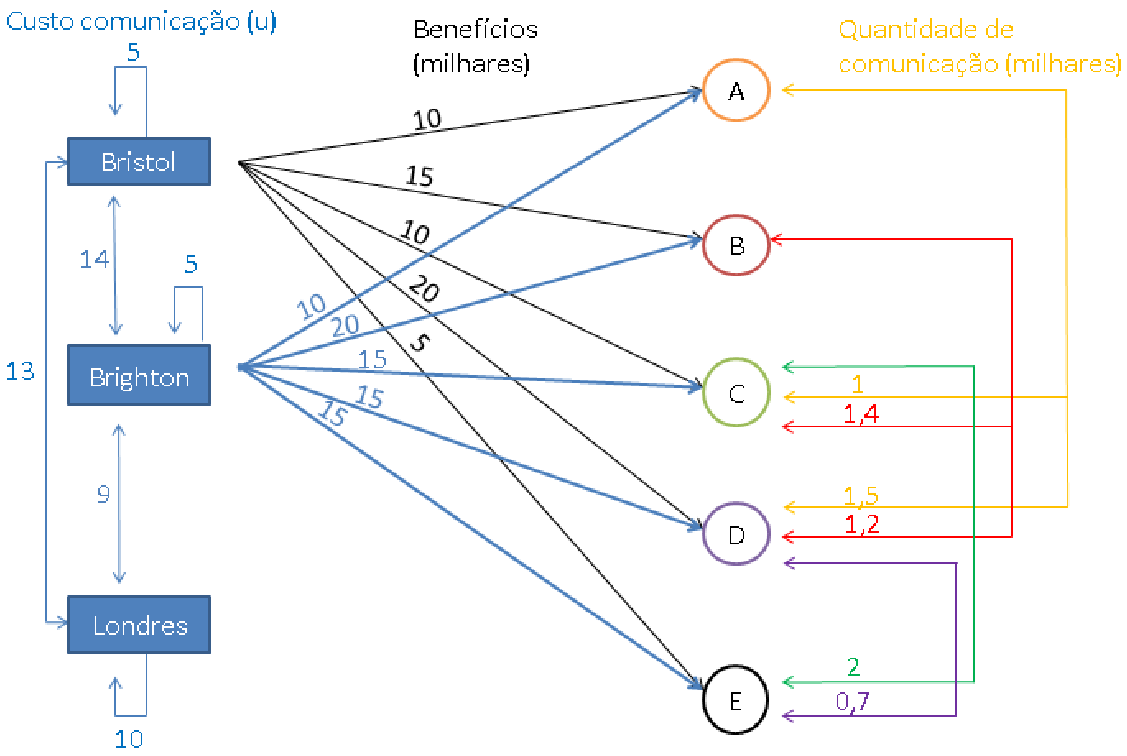
A companhia é composta por 5 departamentos (A, B, C, D, E). As possíveis cidades para realocação são Bristol, Brighton ou o departamento pode continuar em Londres. Nenhuma das cidades pode abrigar mais do que três departamentos. Além dos benefícios, são dados do problema o custo por unidade de comunicação entre as cidades j e l (Djl) e a quantidade de comunicação entre os departamentos i e k por ano (Cik).

Tabela 2: Quantidade de comunicação Cik **(em milhares de unidades)**					
	A	**B**	**C**	**D**	**E**
A	0,0	0,0	1,0	1,5	0,0
B	0,0	0,0	1,4	1,2	0,0
C	1,0	1,4	0,0	0,0	2,0
D	1,5	1,2	0.0	0,0	0,7
E	0,0	0,0	2,0	0,7	0,0

Tabela 3: Custo por unidade de comunicação entre as cidades Djl			
	Brighton	**Bristol**	**Londres**
Brighton	5	14	9
Bristol	14	5	13
Londres	9	13	10

Modelagem esquemática do problema

Figura 1 - Modelagem Esquemática



Modelagem matemática do problema.

Este problema envolve duas decisões de designação simultâneas. Existem *i* departamentos a serem alocadas a *j* cidades, tal que cada departamento é designado a um único local e cada local pode conter até três departamentos. O custo por unidade de comunicação entre as cidades j e l é Djl, a quantidade de comunicação entre os departamentos i e k por ano é Cik e os benefícios do departamento i alocado na cidade j é Bij. O problema consiste em designar os departamentos às cidades de forma a maximizar a receita anual da empresa. Esta receita é composta pela soma dos benefícios que cada departamento fornece ao ser transferido, menos o custo total de comunicação entre os departamentos. Portanto, temos:

Índices:

$i = A, B, C, D, E$ Departamentos;

$l = A, B, C, D, E$ Departamentos;

k = *Bristol, Brighton, Londres* Cidades;

j = *Bristol, Brighton, Londres* Cidades;

Dados:

$D_{i,l}$ Custo por unidade de comunicação entre as cidades j e l;

$C_{j,k}$ Quantidade de comunicação entre os departamentos i e k por ano;

B_{ij} Benefício do departamento i na cidade j;

Variáveis de decisão:

$$X_{ij} = \begin{cases} 1, \text{se o departamento } i \text{ é alocado na cidade } j \\ 0, \text{caso contrário} \end{cases};$$

$$X_{lk} = \begin{cases} 1, \text{se o departamento } i \text{ é alocado na cidade } j \\ 0, \text{caso contrário} \end{cases};$$

O problema é formulado matematicamente como:

$$\text{Maximizar } \sum_{i=1}^{5} \sum_{j=1}^{3} B_{ij} X_{ij} - \sum_{i=1}^{5} \sum_{j=1}^{3} \sum_{k=1}^{5} \sum_{l=1}^{3} \left(X_{ij} X_{lk} D_{il} C_{jk} \right) * \frac{1}{2}$$

Sujeito a:

$$\sum_{i=1}^{5} X_{ij} \leq 3 \qquad \forall \text{ j};$$

$$\sum_{j=1}^{3} X_{ij} = 1 \qquad \forall \text{ i};$$

Na função objetivo, o primeiro conjunto de somatórias refere-se aos benefícios da alocação de cada departamento. Portanto, este termo possui todas as possíveis destinações de cada departamento, porém, só influenciaram o valor da função objetivo, as variáveis de decisão que assumirem o valor 1. Por Exemplo, caso o departamento "A", seja alocado na cidade de Bristol, a variável $X_{A,BRIS}$ assumirá o valor 1 e, consequentemente, o benefício $1*B_{A,BRIS}$ será contabilizado na função objetivo.

No segundo conjunto de somatórias da função objetivo, estão descritas todas as relações de custos entre cada par de departamentos. Para que o custo total de comunicação entre dois departamentos (quantidade * custo unitário) seja contabilizado na função objetivo, as duas variáveis de decisão precisam assumir o valor 1, caso contrário, este termo é zerado. Por exemplo, vamos analisar o caso do departamento “B” ser alocado na cidade de Brighton, enquanto o departamento “E” é alocado na cidade de Londres. Neste caso, as variáveis de decisão $X_{B,BRIG}$ e $X_{E,LON}$ assumiram o valor 1 e, consequentemente, o termo $1*1*C_{B,E}*D_{BRIG,LON}$ será contabilizado na função objetivo.

Para este problema de descentralização, o custo de comunicação entre dois departamentos não muda ao permutar-se a ordem dos departamentos, por exemplo, na análise da relação A-B e, em seguida, B-A ambos apresentam o mesmo custo total de comunicação e estão descritos na função objetivo. Na implementação computacional, o grupo enfrentou uma dificuldade para eliminar a repetição destes “termos-irmãos”. Como solução, dividimos a parte dos custos totais na função objetivo por dois para eliminar duplicação dos termos contabilizados.

A primeira equação das restrições, defini a família de restrições que garante a alocação de no máximo três departamentos em cada cidade. Já a segunda família de restrições garante que cada departamento seja alocado em pelos uma cidade. Como o problema conta apenas com variáveis binárias, não é necessária uma restrição de não negatividade.

Implementação em Python da modelagem matemática

O primeiro passo para resolver computacionalmente o problema, foi transcrever os dados dispostos nas tabelas 1, 2 e 3 para um arquivo separado denominado “dados_descentralização.py”. Além dos dados citados no enunciado do problema, este arquivo também contém dois conjuntos: “departamentos” e “cidades”. Estes que serão utilizados, futuramente, como índices para cada variável de decisão. O conjunto de dados referentes aos benefícios de realocação, quantidade e custos de comunicação foram dispostos em dicionários do Python, enquanto as conjuntos estão dispostos em listas.

Fora decidido utilizar um arquivo separado para armazenar os dados com o intuito de facilitar uma possível atualização destes valores, sem comprometer a integridade do código principal. O nome “benefício” no programa está resumido como “bene”, o objetivo disto é simplificar a escrita.

Figura 2 - Benefícios e Custos

```
departamentos = ['A','B','C','D','E']
cidades = ['LON','BRIS','BRIG']

bene= {
        ('A','LON'):0,
        ('A','BRIS'):10000,
        ('A','BRIG'):10000,
        ('B','LON'):0 ,
        ('B','BRIS'):15000,
        ('B','BRIG'):20000,
        ('C','LON') :0,
        ('C','BRIS'):10000,
        ('C','BRIG'):15000,
        ('D','LON'):0,
        ('D','BRIS'):20000,
        ('D','BRIG'):15000,
        ('E','LON'):0,
        ('E','BRIS'):5000,
        ('E','BRIG'):15000,
        }

custos = {
          ('LON','LON'): 10,
          ('LON','BRIG'): 9,
          ('LON','BRIS'):13,
          ('BRIS','BRIS'):5,
          ('BRIS','BRIG'):14,
          ('BRIS','LON'):13,
          ('BRIG','BRIS'):14,
          ('BRIG','BRIG'):5,
          ('BRIG','LON'):9
          }
```

Figura 3 - Quantidades

```
quantidade = {
              ('A','A'):0,
              ('A','B'):0,
              ('A','C'):1000,
              ('A','D'):1500,
              ('A','E'):0,
              ('B','A'):0,
              ('B','B'):0,
              ('B','C'):1400,
              ('B','D'):1200,
              ('B','E'):0,
              ('C','A'):1000,
              ('C','B'):1400,
              ('C','C'):0,
              ('C','D'):0,
              ('C','E'):2000,
              ('D','A'):1500,
              ('D','B'):1200,
              ('D','C'):0,
              ('D','D'):0,
              ('D','E'):700,
              ('E','A'):0,
              ('E','B'):0,
              ('E','C'):2000,
              ('E','D'):700,
              ('E','E'):0,
              }
```

Após a etapa inicial, implementa-se o modelo matemático. Este será destrinchado para uma melhor análise.

Figura 4 - Implementação do modelo matemático

```
#Importa o Solver Gurobi
from gurobipy import *

#Importa o arquivo que contém os dados do problema
from dados_descentralização import *

#Cria o ambiente do Gurobi atribui a variável 'm'
m = Model ('Descentralização')

#Cria as variáveis de decisão 'x' cujos índices serão os conjuntos departamentos e cidades
x = m.addVars(departamentos,cidades, vtype=GRB.BINARY, name='x')
m.update()

#Constroi a primeira família de restrições - Cada departamento só pode ser alocado em uma cidade
m.addConstrs((quicksum(x.select(d,'*'))==1 for d in departamentos),name='RDeps')

#Constroi a segunda família de restrições - Cada cidade só pode receber 3 departamentos no máx.
m.addConstrs((quicksum(x.select('*',c))<=3 for c in cidades),name='RCidades')

#Função objetivo
FO=-quicksum(x[departamento,cidade]*x[d,c]*quantidade[departamento,d]*custos[cidade,c]
for cidade in cidades for departamento in departamentos for c in cidades for d in departamentos
if quantidade[departamento,d]!=0)/2 + quicksum(x[departamento,cidade]*bene[departamento,cidade]
for cidade in cidades for departamento in departamentos if bene[departamento,cidade]!=0)

#Indica a função objetivo ao Solver e define o problema como maximização
m.setObjective(FO, GRB.MAXIMIZE)

#Comando que otimiza o problema
m.optimize()

#Imprimi na tela as variáveis, seus valores e o resultado da função objetivo
for v in m.getVars():
        print('%s %g' % (v.varName, v.x))
print('Obj: %g' % m.objVal)
```

As variáveis geradas pelo programa são:

Figura 5 - Variáveis Geradas

```
{('A', 'LON'): <gurobi.Var x[A,LON]>, ('A', 'BRIS'): <gurobi.Var x[A,BRIS]>, ('A',
'BRIG'): <gurobi.Var x[A,BRIG]>, ('B', 'LON'): <gurobi.Var x[B,LON]>, ('B', 'BRIS'):
<gurobi.Var x[B,BRIS]>, ('B', 'BRIG'): <gurobi.Var x[B,BRIG]>, ('C', 'LON'): <gurobi.Var
x[C,LON]>, ('C', 'BRIS'): <gurobi.Var x[C,BRIS]>, ('C', 'BRIG'): <gurobi.Var x[C,BRIG]>,
('D', 'LON'): <gurobi.Var x[D,LON]>, ('D', 'BRIS'): <gurobi.Var x[D,BRIS]>, ('D', 'BRIG'):
<gurobi.Var x[D,BRIG]>, ('E', 'LON'): <gurobi.Var x[E,LON]>, ('E', 'BRIS'): <gurobi.Var
x[E,BRIS]>, ('E', 'BRIG'): <gurobi.Var x[E,BRIG]>}
```

A família de restrições que determina que cada departamento pode estar alocado em apenas uma cidade é emitida do seguinte modo:

Figura 6 - Família de Restrições (departamentos)

```
<gurobi.LinExpr: x[A,LON] + x[A,BRIS] + x[A,BRIG]> == 1>
<gurobi.LinExpr: x[B,LON] + x[B,BRIS] + x[B,BRIG]> == 1>
<gurobi.LinExpr: x[C,LON] + x[C,BRIS] + x[C,BRIG]> == 1>
<gurobi.LinExpr: x[D,LON] + x[D,BRIS] + x[D,BRIG]> == 1>
<gurobi.LinExpr: x[E,LON] + x[E,BRIS] + x[E,BRIG]> == 1>
```

A família de restrições que define que cada cidade só pode receber, no máximo, 3 departamentos é expressa da seguinte forma:

Figura 7 - Família de Restrições (cidades)

```
<gurobi.LinExpr: x[A,LON] + x[B,LON] + x[C,LON] + x[D,LON] + x[E,LON]> <= 3>
<gurobi.LinExpr: x[A,BRIS] + x[B,BRIS] + x[C,BRIS] + x[D,BRIS] + x[E,BRIS]> <= 3>
<gurobi.LinExpr: x[A,BRIG] + x[B,BRIG] + x[C,BRIG] + x[D,BRIG] + x[E,BRIG]> <= 3>
```

Na função objetivo encontrou-se uma dificuldade na implementação computacional. O comando gerador da função cria uma expressão redundante, pois fornece o termo A-B e B-A. Para corrigir esse problema, divide-se a função por 2.

Figura 8 - Função Objetivo

```
<gurobi.QuadExpr: 10000.0 x[A,BRIS] + 15000.0 x[B,BRIS] + 10000.0 x[C,BRIS] + 20000.0 x[D,BRIS] + 5000.0 x[E,BRIS]
+ 10000.0 x[A,BRIG] + 20000.0 x[B,BRIG] + 15000.0 x[C,BRIG] + 15000.0 x[D,BRIG] + 15000.0 x[E,BRIG] + [ -5000.0
x[A,LON] * x[C,LON] + -7500.0 x[A,LON] * x[D,LON] + -6500.0 x[A,LON] * x[C,BRIS] + -9750.0 x[A,LON] * x[D,BRIS] +
-4500.0 x[A,LON] * x[C,BRIG] + -6750.0 x[A,LON] * x[D,BRIG] + -7000.0 x[B,LON] * x[C,LON] + -6000.0 x[B,LON] *
x[D,LON] + -9100.0 x[B,LON] * x[C,BRIS] + -7800.0 x[B,LON] * x[D,BRIS] + -6300.0 x[B,LON] * x[C,BRIG] + -5400.0
x[B,LON] * x[D,BRIG] + -5000.0 x[C,LON] * x[A,LON] + -7000.0 x[C,LON] * x[B,LON] + -10000.0 x[C,LON] * x[E,LON] +
-6500.0 x[C,LON] * x[A,BRIS] + -9100.0 x[C,LON] * x[B,BRIS] + -13000.0 x[C,LON] * x[E,BRIS] + -4500.0 x[C,LON] *
x[A,BRIG] + -6300.0 x[C,LON] * x[B,BRIG] + -9000.0 x[C,LON] * x[E,BRIG] + -7500.0 x[D,LON] * x[A,LON] + -6000.0
x[D,LON] * x[B,LON] + -3500.0 x[D,LON] * x[E,LON] + -9750.0 x[D,LON] * x[A,BRIS] + -7800.0 x[D,LON] * x[B,BRIS] +
-4550.0 x[D,LON] * x[E,BRIS] + -6750.0 x[D,LON] * x[A,BRIG] + -5400.0 x[D,LON] * x[B,BRIG] + -3150.0 x[D,LON] *
x[E,BRIG] + -10000.0 x[E,LON] * x[C,LON] + -3500.0 x[E,LON] * x[D,LON] + -13000.0 x[E,LON] * x[C,BRIS] + -4550.0
x[E,LON] * x[D,BRIS] + -9000.0 x[E,LON] * x[C,BRIG] + -3150.0 x[E,LON] * x[D,BRIG] + -6500.0 x[A,BRIS] * x[C,LON]
+ -9750.0 x[A,BRIS] * x[D,LON] + -2500.0 x[A,BRIS] * x[C,BRIS] + -3750.0 x[A,BRIS] * x[D,BRIS] + -7000.0 x[A,BRIS]
* x[C,BRIG] + -10500.0 x[A,BRIS] * x[D,BRIG] + -9100.0 x[B,BRIS] * x[C,LON] + -7800.0 x[B,BRIS] * x[D,LON] +
-3500.0 x[B,BRIS] * x[C,BRIS] + -3000.0 x[B,BRIS] * x[D,BRIS] + -9800.0 x[B,BRIS] * x[C,BRIG] + -8400.0 x[B,BRIS]
* x[D,BRIG] + -6500.0 x[C,BRIS] * x[A,LON] + -9100.0 x[C,BRIS] * x[B,LON] + -13000.0 x[C,BRIS] * x[E,LON] +
-2500.0 x[C,BRIS] * x[A,BRIS] + -3500.0 x[C,BRIS] * x[B,BRIS] + -5000.0 x[C,BRIS] * x[E,BRIS] + -7000.0 x[C,BRIS]
* x[A,BRIG] + -9800.0 x[C,BRIS] * x[B,BRIG] + -14000.0 x[C,BRIS] * x[E,BRIG] + -9750.0 x[D,BRIS] * x[A,LON] +
-7800.0 x[D,BRIS] * x[B,LON] + -4550.0 x[D,BRIS] * x[E,LON] + -3750.0 x[D,BRIS] * x[A,BRIS] + -3000.0 x[D,BRIS] *
x[B,BRIS] + -1750.0 x[D,BRIS] * x[E,BRIS] + -10500.0 x[D,BRIS] * x[A,BRIG] + -8400.0 x[D,BRIS] * x[B,BRIG] +
-4900.0 x[D,BRIS] * x[E,BRIG] + -13000.0 x[E,BRIS] * x[C,LON] + -4550.0 x[E,BRIS] * x[D,LON] + -5000.0 x[E,BRIS] *
x[C,BRIS] + -1750.0 x[E,BRIS] * x[D,BRIS] + -14000.0 x[E,BRIS] * x[C,BRIG] + -4900.0 x[E,BRIS] * x[D,BRIG] +
-4500.0 x[A,BRIG] * x[C,LON] + -6750.0 x[A,BRIG] * x[D,LON] + -7000.0 x[A,BRIG] * x[C,BRIS] + -10500.0 x[A,BRIG] *
x[D,BRIS] + -2500.0 x[A,BRIG] * x[C,BRIG] + -3750.0 x[A,BRIG] * x[D,BRIG] + -6300.0 x[B,BRIG] * x[C,LON] + -5400.0
x[B,BRIG] * x[D,LON] + -9800.0 x[B,BRIG] * x[C,BRIS] + -8400.0 x[B,BRIG] * x[D,BRIS] + -3500.0 x[B,BRIG] *
x[C,BRIG] + -3000.0 x[B,BRIG] * x[D,BRIG] + -4500.0 x[C,BRIG] * x[A,LON] + -6300.0 x[C,BRIG] * x[B,LON] + -9000.0
x[C,BRIG] * x[E,LON] + -7000.0 x[C,BRIG] * x[A,BRIS] + -9800.0 x[C,BRIG] * x[B,BRIS] + -14000.0 x[C,BRIG] *
x[E,BRIS] + -2500.0 x[C,BRIG] * x[A,BRIG] + -3500.0 x[C,BRIG] * x[B,BRIG] + -5000.0 x[C,BRIG] * x[E,BRIG] +
-6750.0 x[D,BRIG] * x[A,LON] + -5400.0 x[D,BRIG] * x[B,LON] + -3150.0 x[D,BRIG] * x[E,LON] + -10500.0 x[D,BRIG] *
x[A,BRIS] + -8400.0 x[D,BRIG] * x[B,BRIS] + -4900.0 x[D,BRIG] * x[E,BRIS] + -3750.0 x[D,BRIG] * x[A,BRIG] +
-3000.0 x[D,BRIG] * x[B,BRIG] + -1750.0 x[D,BRIG] * x[E,BRIG] + -9000.0 x[E,BRIG] * x[C,LON] + -3150.0 x[E,BRIG] *
x[D,LON] + -14000.0 x[E,BRIG] * x[C,BRIS] + -4900.0 x[E,BRIG] * x[D,BRIS] + -5000.0 x[E,BRIG] * x[C,BRIG] +
-1750.0 x[E,BRIG] * x[D,BRIG] ]>
```

Por fim, o programa apresenta como solução ótima os seguintes valores para cada variável:

Figura 9 - Resultados

```
Optimal solution found (tolerance 1.00e-04)
Best objective 1.490000000000e+04, best bound 1.490000000000e+04, gap 0.0000%
x[A,LON] 0
x[A,BRIS] 1
x[A,BRIG] 0
x[B,LON] 0
x[B,BRIS] 0
x[B,BRIG] 1
x[C,LON] 0
x[C,BRIS] 0
x[C,BRIG] 1
x[D,LON] 0
x[D,BRIS] 1
x[D,BRIG] 0
x[E,LON] 0
x[E,BRIS] 0
x[E,BRIG] 1
Obj: 14900
```

Note-se que os departamentos A e D estão localizados em Bristol, enquanto os departamentos B, C e E estão localizados em Brighton. O valor ótimo da função objetivo é 14900 libras.

Análise de Sensibilidade

Diante do resultado ótimo considerou-se interessante realizar um único tipo de análise de sensibilidade: a mudança do RHS da restrição que limita o número de departamentos por cidade. Os outros tipos de análises não são cabíveis pois seria necessário estimar valores hipotéticos sem base para as quantidades e custos de comunicação e os benefícios de alocação. E, portanto a análise de mudança no RHS não é necessário estimar e sim uma relaxar para as capacidades de alocação de departamentos, de cada cidade.

RHS=2

Figura 10 - Análise de Sensibilidade (RHS=2)

```
x[A,BRIS] 1
x[B,BRIG] 1
x[C,BRIG] 1
x[D,BRIS] 1
x[E,LON] 1
Obj: -7400
```

RHS=4

Figura 11 - Análise de Sensibilidade (RHS=4)

```
x[A,LON] 1
x[B,BRIG] 1
x[C,BRIG] 1
x[D,BRIG] 1
x[E,BRIG] 1
Obj: 16000
```

RHS=5

Figura 12 - Análise de Sensibilidade (RHS=5)

```
x[A,BRIG] 1
x[B,BRIG] 1
x[C,BRIG] 1
x[D,BRIG] 1
x[E,BRIG] 1
Obj: 36000
```

Restringir que cada cidade abranja somente um departamento não faz sentido para esse problema, pois o número de cidades disponíveis é menor que a quantidade de departamentos a serem alocados. Passando para próximo valor inteiro "2" , o problema alterou suas variáveis básicas assim como o valor de sua função objetivo, porém os benefícios ainda não cobrem os custos dos departamentos e a função objetivo da negativa.

Aumentando mais duas unidades "4" do RHS desta restrição, as variáveis básicas também se alteram, aumentando a função objetivo para 16000 Libras, um resultado melhor que o resultado da restrição original do problema "3". Uma tendência observada é que diferente do cenário original onde as localidades são bem distribuídas, neste cenário os departamentos tendem a se alocar na cidade de Brighton.

Por fim, permitindo que cada cidade possa alocar até 5 departamentos, ou seja, é como se a segunda restrição fosse excluída. Constatamos que todos os departamentos são alocados em Brighton e o lucro total é de 36000 Libras. O que é um ponto relevante para a empresa revisar a possibilidade de aumentar a capacidade de alocação de departamentos na cidade de Brighton. Comparando a diferença de lucro obtido nesse cenário com o cenário original, com o gasto necessário para essa expansão da capacidade de cada cidade.

Referências Bibliográficas

ARENALES, M.; ARMENTANO, V. A.; MORABITO, R.; YANASSE, H. H. Pesquisa Operacional. Rio de Janeiro: Campus/elsevier, 2007. p181.