



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
ENGENHARIA DA COMPUTAÇÃO

RELATÓRIO DO TRABALHO FINAL DE PESQUISA OPERACIONAL

GIOVANNI BRUNO TRAVASSOS DE CARVALHO
WALLISSON DANTAS DA SILVA
JOSÉ EUGÊNIO CARVALHO DE SOUZA

JOÃO PESSOA, 2020.

GIOVANNI BRUNO TRAVASSOS DE CARVALHO

WALLISSON DANTAS DA SILVA

JOSÉ EUGÊNIO CARVALHO DE SOUZA

PROJETO FINAL - LOCALIZAÇÃO DE UPAs

Relatório solicitado pelo professor Teobaldo Leite Bulhões Júnior, do curso de Engenharia de Computação, para outorgamento da disciplina de Pesquisa Operacional.

JOÃO PESSOA, 2020.

Sumário

1 - Introdução	4
2 - Definição do Problema	4
3 - Modelagem	5
4 - Instruções	6
5 - Resultados	9
6 - Dificuldades e Melhorias	12
7 - Referências	12

1 - Introdução

O projeto final consiste na modelagem e implementação de um problema de Programação Linear Inteira, localização de facilidades. Na temática, o problema consiste em aplicar os conhecimentos adquiridos na disciplina de Pesquisa Operacional, tendo em vista a modelagem do problema, definindo as restrições básicas. Por fim a implementação do programa, desenvolvida em Python, para que funcione em qualquer conjunto de dados e forneça a resposta completa.

A implementação foi realizada na linguagem de programação Python, com auxílio da biblioteca de programação linear DOCPLEX(*Decision Optimization CPLEX*) e também a MATPLOTLIB.

2 - Definição do Problema

No problema de Localização de UPAs, existem n distritos numa região metropolitana, cuja distância entre um distrito e outro é dada por:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Com isso deseja-se escolher em quais localidades as UPAs devem ser instaladas. Em regra, é obrigatório que se siga algumas restrições, sendo elas:

- Distância entre distrito e UPA mais próxima deve ser no máximo X km;
- Distância entre distrito e segunda UPA mais próxima deve ser no máximo Y km;
- Distância entre duas UPAs de no máximo Z km.

É necessário que suas respectivas saídas, para resolução do problema, contenha as seguintes indicações:

- Saída 1 = Quais distritos foram instaladas UPAs;
- Saída 2 = Distância de cada distrito para a primeira e segunda UPA mais próxima.

O objetivo da problemática é buscar a minimização do número de UPAs, respeitando as restrições impostas.

3 - Modelagem

$$\text{Min} \quad \sum_{i=1}^n X_i \quad (\text{a})$$

$$\sum_{i=1; d_{ij} \leq K}^n X_i \geq 1 \quad , \forall j \in \{1, \dots, n\} \quad (\text{b})$$

$$\sum_{i=1; d_{ij} \leq Y}^n X_i \geq 2 \quad , \forall j \in \{1, \dots, n\} \quad (\text{c})$$

$$X_i + X_j \leq 1 \quad , \forall (i, j) \in \{1, \dots, n\} ; d_{ij} \leq Z; i \neq j \quad (\text{d})$$

$$X_i, X_j \in \{0, 1\} \quad , \forall (i, j) \in \{1, \dots, n\} \quad (\text{e})$$

Para resolução do problema foi preciso criar n variáveis binárias que são representadas no distrito n existindo ou não uma UPA. Os valores de K , Y e Z são fornecidos no arquivo de entrada, assim como as coordenadas (X, Y) de cada distrito.

- (a) Função Objetiva: Minimizar um somatório de todas as variáveis binárias para cada distrito;
- (b) Primeira Restrição: Para todo j pertencente a $\{1, \dots, n\}$, é feito um somatório das variáveis binárias, se a distância d_{ij} for menor que o valor K fornecido, cujo valor tem que ser maior ou igual a 1. Essa restrição indica que é preciso ter ao menos uma UPA dentro dessa distância mínima;
- (c) Segunda Restrição: Para todo j pertencente a $\{1, \dots, n\}$, é feito um somatório das variáveis binárias, se a distância d_{ij} for menor que o valor Y fornecido, cujo valor tem que ser maior ou igual a 2. Essa restrição indica que é preciso ter ao menos duas UPA dentro dessa distância mínima;
- (d) Terceira Restrição: Para todo i e j pertencente a $\{1, \dots, n\}$, se a distância d_{ij} for menor que Z , a soma das variáveis binárias tem que ser 1, ou seja, apenas um dos distritos poderá receber uma UPA;
- (e) Declaração das variáveis binárias.

4 - Instruções

Primeiramente foi instalado o **Anaconda Navigator**, como uma instalação padrão, após a instalação abrimos ele e clicamos no CMD.exe Prompt.

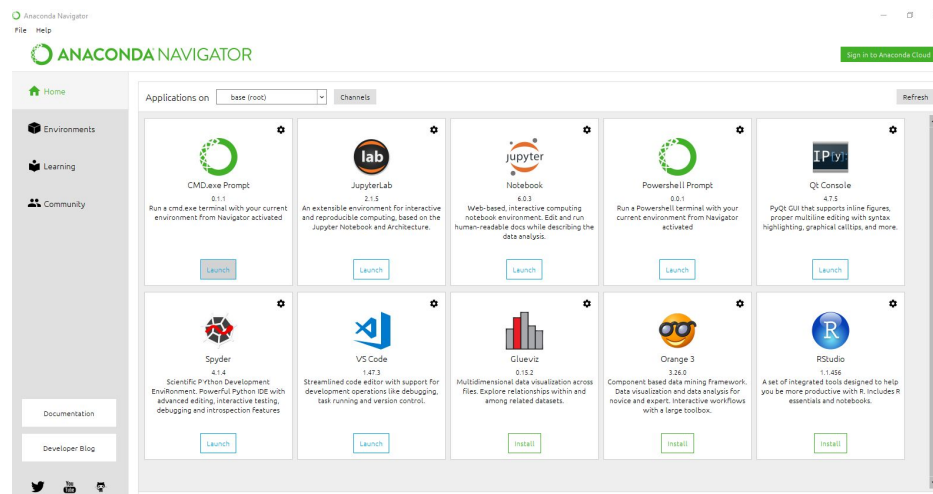


Figura 1 - Ambiente inicial do Anaconda

Após aberto o prompt deve ser visto essa tela:

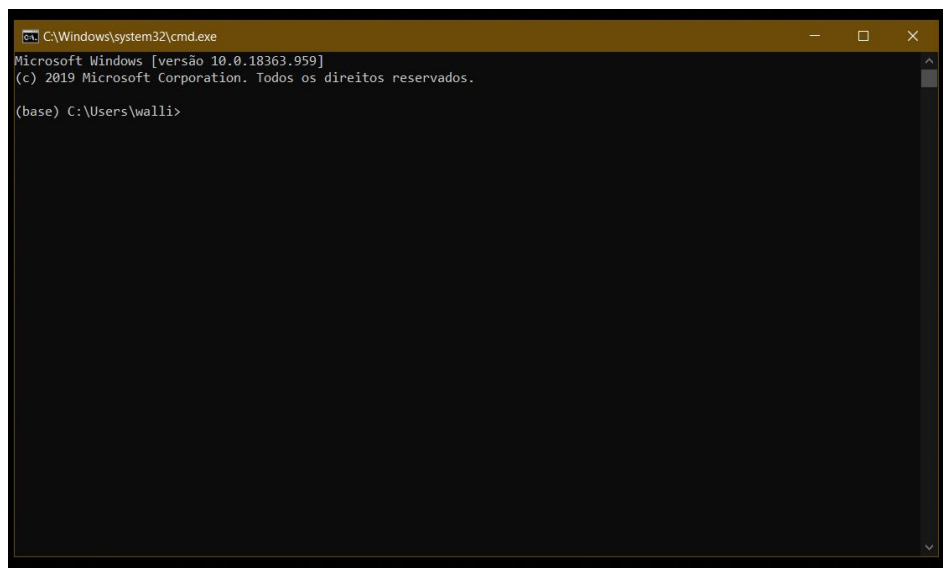
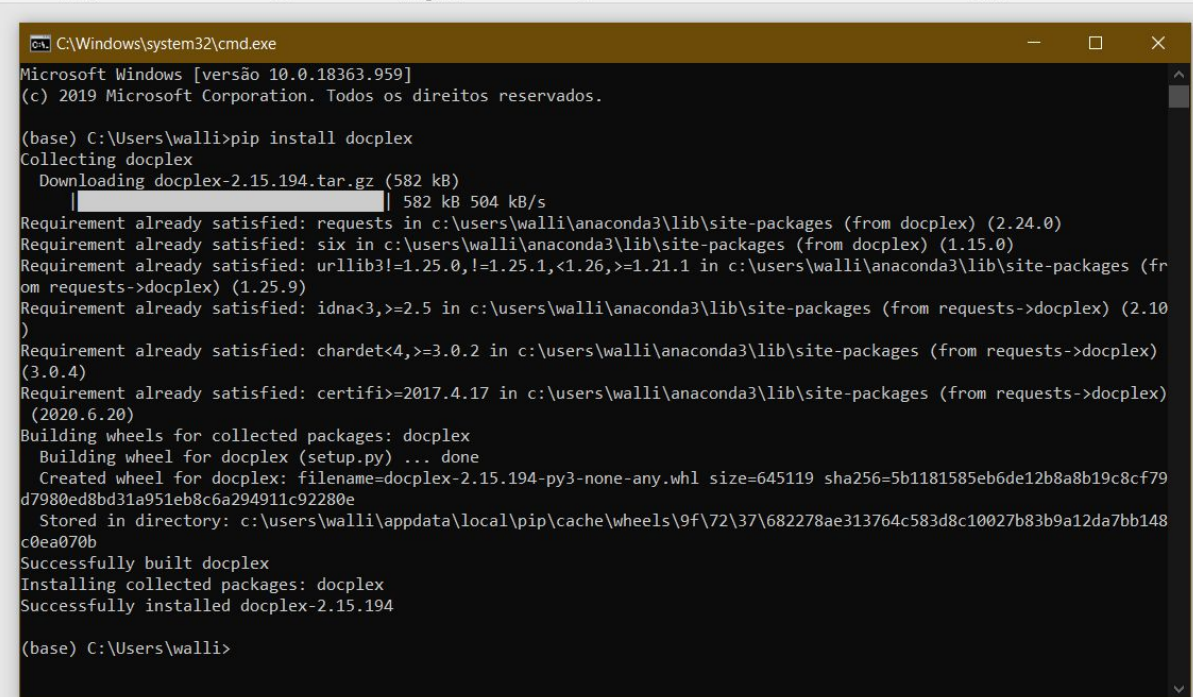


Figura 2 - Prompt de comando inicializado.

Instalamos a biblioteca Cplex com o comando:

```
$ pip install docplex
```

Após a instalação, irá aparecer essa imagem no CMD. Sendo assim o ambiente estará pronto para usar a biblioteca do Cplex



```
C:\Windows\system32\cmd.exe
Microsoft Windows [versão 10.0.18363.959]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

(base) C:\Users\walli>pip install docplex
Collecting docplex
  Downloading docplex-2.15.194.tar.gz (582 kB)
    | 582 kB 504 kB/s
Requirement already satisfied: requests in c:\users\walli\anaconda3\lib\site-packages (from docplex) (2.24.0)
Requirement already satisfied: six in c:\users\walli\anaconda3\lib\site-packages (from docplex) (1.15.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\walli\anaconda3\lib\site-packages (from requests->docplex) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in c:\users\walli\anaconda3\lib\site-packages (from requests->docplex) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\walli\anaconda3\lib\site-packages (from requests->docplex) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\walli\anaconda3\lib\site-packages (from requests->docplex) (2020.6.20)
Building wheels for collected packages: docplex
  Building wheel for docplex (setup.py) ... done
  Created wheel for docplex: filename=docplex-2.15.194-py3-none-any.whl size=645119 sha256=5b1181585eb6de12b8a8b19c8cf79d7980ed8bd31a951eb8c6a294911c92280e
  Stored in directory: c:\users\walli\appdata\local\pip\cache\wheels\9f\72\37\682278ae313764c583d8c10027b83b9a12da7bb148c0ea070b
Successfully built docplex
Installing collected packages: docplex
Successfully installed docplex-2.15.194

(base) C:\Users\walli>
```

Figura 3 - Momento de conclusão da instalação e com êxito.

Essa biblioteca nos permite adicionar de forma rápida e fácil o poder de otimização do CPLEX, e possibilita modelar e resolver os problemas na API do Python.

Usamos também a biblioteca chamada Matplotlib, o Anaconda já vem com esse recurso instalado, mas caso fosse usado outro ambiente, seria necessário instalá-lo.

Matplotlib é uma biblioteca de Python extremamente útil para aplicações em engenharia, matemática, estatística, física e qualquer outra área que lide com números e gráficos. No entanto, essa biblioteca não está nativamente presentes no Python puro.

Caso queira instalar a biblioteca no Linux, basta abrir *prompt* de comando e digitar algum dos seguinte código:

```
$pip install matplotlib
```

```
$conda install matplotlib
```

5 - Resultados

O formato do arquivo de entrada foi definido da seguinte maneira:

n # Número de Distritos

K # Distância mínima de um distrito para UPA mais próxima

Y # Distância mínima de um distrito para segunda UPA mais próxima

Z # Distância máxima de uma UPA para outra

(x,y) # Coordenadas de cada distrito

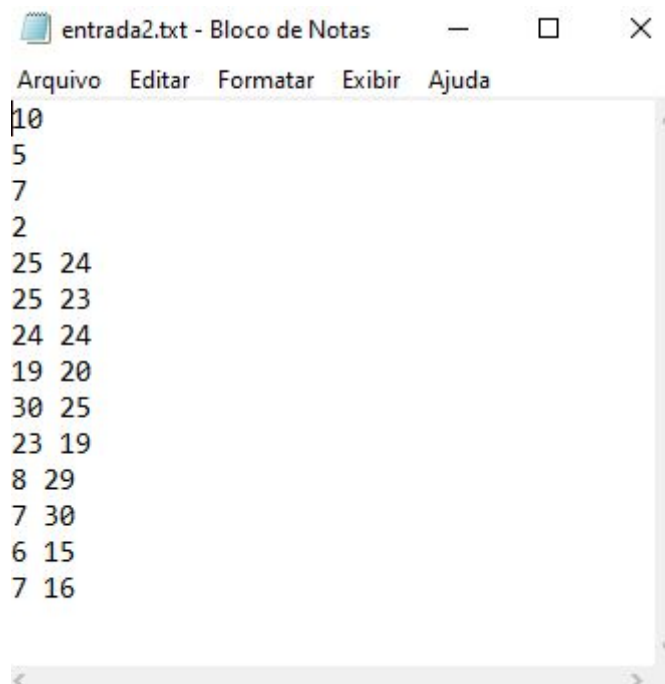


Figura 4 - Exemplo de arquivo de entrada.

Executando e printando o comando `export_to_string()`, conseguimos ver a modelagem feita pelo programa, e como resultado do exemplo de “entrada2.txt” obtivemos:


```

\Problem name: Localizacao de UPAs

Minimize
obj: X0 + X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9
Subject To
c1: X0 + X1 + X2 >= 1
c2: X0 + X1 + X2 + X5 >= 1
c3: X0 + X1 + X2 >= 1
c4: X3 + X5 >= 1
c5: X4 >= 1
c6: X1 + X3 + X5 >= 1
c7: X6 + X7 >= 1
c8: X6 + X7 >= 1
c9: X8 + X9 >= 1
c10: X8 + X9 >= 1
c11: X0 + X1 + X2 + X4 + X5 >= 1
c12: X0 + X1 + X2 + X3 + X4 + X5 >= 1
c13: X0 + X1 + X2 + X3 + X4 + X5 >= 1
c14: X1 + X2 + X3 + X5 >= 1
c15: X0 + X1 + X2 + X4 >= 1
c16: X0 + X1 + X2 + X3 + X5 >= 1
c17: X6 + X7 >= 1
c18: X6 + X7 >= 1
c19: X8 + X9 >= 1
c20: X8 + X9 >= 1
c21: X0 + X1 <= 1
c22: X0 + X2 <= 1
c23: X0 + X1 <= 1
c23: X0 + X1 <= 1
c24: X1 + X2 <= 1
c25: X0 + X2 <= 1
c26: X1 + X2 <= 1
c27: X6 + X7 <= 1
c28: X6 + X7 <= 1
c29: X8 + X9 <= 1
c30: X8 + X9 <= 1

Bounds
0 <= X0 <= 1
0 <= X1 <= 1
0 <= X2 <= 1
0 <= X3 <= 1
0 <= X4 <= 1
0 <= X5 <= 1
0 <= X6 <= 1
0 <= X7 <= 1
0 <= X8 <= 1
0 <= X9 <= 1

Binaries
X0 X1 X2 X3 X4 X5 X6 X7 X8 X9

```

Figura 5 - Comando `export_to_string()`.

Podemos observar na execução desse comando que a função objetiva descrita em `obj` foi feita corretamente, assim como as 30 restrições `c`, e as 10 variáveis binárias também foram criadas corretamente para o problema da entrada “entrada2.txt”.

Utilizamos o comando “`print_information()`” para obter outras informações sobre o modelo criado, como o número de variáveis e seu tipo, e o número de restrições e seu tipo.

```

In [383]: prob.print_information()
Model: Localizacao de UPAs
- number of variables: 10
  - binary=10, integer=0, continuous=0
- number of constraints: 30
  - linear=30
- parameters: defaults
- problem type is: MILP

```

Figura 6 - Comando `print_information()`.

Agora, executando o solver e printando sua solução através do comando “`display()`”, obtemos:

```

Version identifier: 12.10.0.0 | 2019-11-27 | 843d4de2ae
CPXPARAM_Read_DataCheck          1
CPXPARAM_RandomSeed              201903125
2 of 2 MIP starts provided solutions.
MIP start 'm1' defined initial solution with objective 5.0000.
Tried aggregator 1 time.
MIP Presolve eliminated 30 rows and 10 columns.
MIP Presolve modified 1 coefficients.
All rows and columns eliminated.
Presolve time = 0.00 sec. (0.03 ticks)

Root node processing (before b&c):
  Real time                =    0.02 sec. (0.04 ticks)
Parallel b&c, 4 threads:
  Real time                =    0.00 sec. (0.00 ticks)
  Sync time (average)      =    0.00 sec.
  Wait time (average)      =    0.00 sec.
-----
Total (root+branch&cut) =    0.02 sec. (0.04 ticks)
solution for: Localizacao de UPAs
objective: 5
X0=1
X3=1
X4=1
X6=1
X8=1

```

Figura 7 - Comando *display()*.

Através do comando acima podemos observar que o resultado da função objetiva em “*objective*”, que para o arquivo de entrada “*entrada2.txt*” foi 5. Também podemos observar as variáveis binárias com 1, que representam o distrito que receberá uma UPA.

Em seguida, *printamos* a saída da maneira que foi pedida no apêndice 2 da descrição do problema, mostrando em qual distrito serão localizadas as UPAs e a distância de um distrito que não tem UPA para a UPA mais próxima.

```

UPA localizada no distrito: 0
Distância do distrito 1 para a UPA mais próxima: 1.0
Distância do distrito 2 para a UPA mais próxima: 1.0
UPA localizada no distrito: 3
UPA localizada no distrito: 4
Distância do distrito 5 para a UPA mais próxima: 1.0
UPA localizada no distrito: 6
Distância do distrito 7 para a UPA mais próxima: 1.0
UPA localizada no distrito: 8
Distância do distrito 9 para a UPA mais próxima: 1.0

```

Figura 8 - Saída do programa.

Por fim, plotamos os distritos em um sistema de coordenadas, pintando o lugar que receberá UPA de vermelho, e os demais de preto:

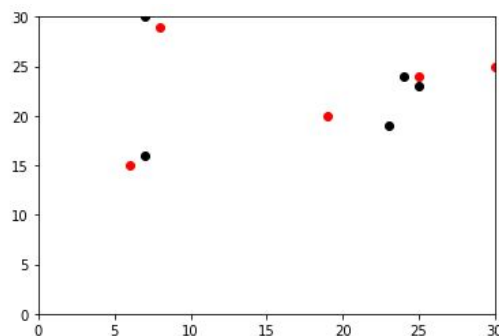


Figura 9 - Distritos no sistema de coordenadas.

6 - Dificuldades e Melhorias

O primeiro problema encontrado foi na criação da segunda restrição, ao colocar ' ≥ 2 ' alguns somatórios que só entraram duas variáveis deram erro, pois não era possível que uma variável binária fosse maior ou igual a 2. Portanto deixamos a segunda restrição com ' ≥ 1 '.



```
c91: x30 >= 2
```

Figura 10 - Restrição de número 91, exemplo de erro.

Tivemos dificuldade e por consequência não conseguimos printar corretamente a distância do distrito que não tem UPA para a UPA mais próxima, nem para segunda UPA mais próxima.

Por fim, para melhoria do projeto, também sugerimos plotar raios de valor K, Y e Z para cada distrito, facilitando a visualização do problema. Nessa mesma plotagem, é preciso ainda criar uma função que descubra os maiores valores de x e y para usar de coordenada na criação do sistema.

7 - Referências

1. [IBM Knowledge Center](#)
2. https://www.youtube.com/watch?v=hqGZzRh00y0&list=PL_wz_RHE6pEYEJO-vDNwHi2F7k-WNgOfQ
3. [IBMDecisionOptimization/docplex-examples: These samples demonstrate how to use the DOcplex library to model and solve optimization problems.](#)
4. <http://ibmdecisionoptimization.github.io/docplex-doc/#mathematical-programming-modeling-for-python-using-docplex-mp-docplex-mp>