MECAI 2024 III Workshop de Matemática, Estatística e Computação Aplicadas à Indústria

Comparativo entre *MS Excel Solver* e *Python Gurobi* na resolução de problemas de otimização com 3 dimensões

Gabriel Bortoli[1]
Alexandre C. B. Delbem[2]
ICMC-USP

1 Motivação e cenário hipotético analisado

Há pouca literatura disponível com comparações entre os resultados obtidos no uso do *MS Excel Solver* e do pacote *Python Gurobi*, sendo um dos poucos artigos acadêmicos uma análise feita por To e Khakali [1], sobre a possibilidade de adição ou remoção de vagas em estacionamento, mantendo uma alocação ótima de de vagas. Além disso, em geral, os problemas apresentados para exemplificar casos de programação matemática são bidemensionais.

No exemplo apresentado, pensou-se em um problema com três dimensões, afim de averiguar se o código gerado em Python para resolvê-lo seria claro e de fácil entendimento, para ser reutilizado na resolução de outros problemas.

Para a análise compartiva entre os resultados obtidos pelos dois diferentes métodos, o seguinte cenário hipotético foi montado, com inspiração em uma montadora de automóveis, com fábricas em diferentes cidades, produzindo modelos diversos, afim de suprir a demanda de alguns destinos.

- Uma montadora de automóveis tem 3 fábricas em cidades diferentes, chamadas de Origem 1, 2 e 3 (primeira dimensão);
- Em cada Origem são produzidos 4 Carros, chamados de 1, 2, 3 e 4 (segunda dimensão);
- A demanda vem das cidades de Destino 1, 2, 3, 4 e 5 (terceira dimensão);
- É dada a distância entre as Origens e os Destinos definidas (primeiro parâmetro);
- O custo de produção de cada Carro em cada Origem (segundo parâmetro);
- Assim como o custo de transporte de cada Carro, fixo por unidade e adicional por distância (terceiro parâmetro - composto);
- Além da demanda por cada Carro em cada Destino (primeira restrição);
- Bem como a capacidade de produção de cada Carro em cada Origem (segunda restrição);
- O custo final de produção e entrega dos Carros depende do custo de produção em cada Origem e do transporte da Origem até o Destino (função objetivo);

[1] gbortoli@usp.br
[2] acbd@icmc.usp.br

- A partir da Demanda exigida e da Capacidade de produção, quantos e quais carros devem ser produzidos em cada Origem, afim de que o Custo Total seja Mínimo? (Variáveis de decisão nas 3 dimensões apresentadas).

2 Informações do Problema

Dado o texto, vê-se que há 60 variáveis de decisão a serem calculadas: são 4 modelos de Carros, para 5 Destinos, produzidos por 3 Origens, ou seja, 4 * 5 * 3 = 60.

Tem-se um 'cubo' de dados, em que pode-se imaginar cada componente da variável como um eixo: Carros no eixo X, Destinos no eixo Y e Origens no eixo Z.

Além disso, as restrições apresentam-se em conjuntos bidemensionais, como a Demanda, que cruza informações de Destino e Carro e a Capacidade de Produção, que considera a Origem e a quantidade de cada Carro.

3 Modelagem do problema no *Microsoft Excel* com *add-in Solver*

A primeira abordagem do problema foi feita utilizando-se o Solver, disponível para o Excel. Os resultados obtidos foram iguais tanto ao usar o método *Simplex LP* quanto o método *GRG Nonlinear*.

Para a modelagem do problema no Excel fez-se necessária sua decomposição em três problemas bidemsionais, tendo em vista as limitações das planilhas. Dessa forma foram feitas três tabelas com o cruzamento das informações de Destino e Carros produzidos, uma para cada Origem.

Criou-se também 4 tabelas auxiliares (além das 3 de parâmetros e 2 de restrições) para facilitar o cálculo de custo total e a visualização dos valores intermediários; além de tabelas para as comparações com as restrições.

O resultado final obtido de foi de **176.960,00** unidades monetárias.

4 Implementação em *Python* utilizando o pacote *Gurobi*

Para a implementação em Python, usou-se como único pacote auxiliar o *gurobipy*.

As 5 tabelas mencionadas previamente foram criadas como listas bidimensionais e foram definidas 3 variáveis para representar as Origens (O), Destinos (D) e Carros (C).

O modelo foi criado na variável 'm' e as 60 variáveis de decisão em 'x', com 3 dimensões (x[O,D,C]).

A função objetivo a ser minimizada foi definida como o somatório da soma do custo de produção, custo de transporte e custo por distância, variando em laços *for* concatenados, para as variáveis "O", "D"e "C":

```
m.setObjective(sum(x[o,d,c]*custo_prod[o][c] +
        x[o,d,c]*custo_transp[c][0] +
        x[o,d,c]*(distancia[o][d]*custo_transp[c][1])
        for o in range(O) for d in range(D) for c in range (C)),
        GRB.MINIMIZE)
```

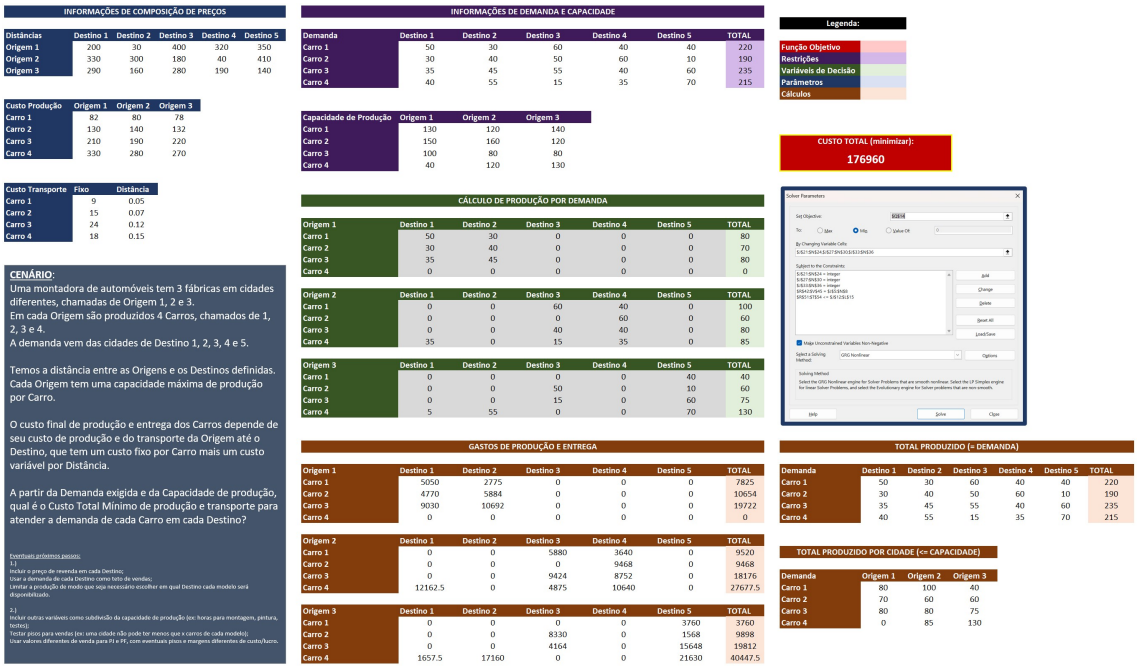

Figura 1: Configuração no Excel.

A definição das restrições é a parte que exige maior atenção, especialmente em relação a ordem e escopo das somatórias e laços *for* a serem usados. No exemplo, há duas restrições, sendo a primeira a Demanda, onde a soma da produção em cada Origem (de cada carro para cada destino) deve ser igual à necessidade em cada Destino por cada Carro; já a segunda dá-se na relação entre a soma da produção (em cada origem, de cada carro) para todos os Destinos, que deve ser igual ou inferior à capacidade de cada Origem para cada Carro. Dessa forma temos:

```
m.addConstrs(sum(x[o,d,c] for o in range(O)) ==
    demanda[d][c] for d in range(D) for c in range(C))
m.addConstrs(sum(x[o,d,c] for d in range(D)) <=
    capacidade[o][c] for o in range (O) for c in range(C))
```

O resultado final foi calculado em 5 iterações, sendo igual ao valor obtido no *Excel*, de **176.960,00** unidades monetárias.

5 Conclusões

Observou-se que os resultados obtidos nos dois métodos foram iguais, validando a solução.

Em relação à codificação em *Python*, onde matrizes tridimensionais foram usadas na definição das variáveis de decisão, notou-se que não há uma diferença relevante em relação aos problemas bidimensionais, ainda que a complexidade, especialmente na definição das restrições, seja aumentada. Portanto é necessário atentar-se ainda mais nos laços de repetições, somatórios e ordem de variáveis.

```
print("Valor da solução ótima:\t", round(m.objVal))

for o in range(O):
  print("\nOrigem ",o+1,":")
  print("\tDestino 1:\tDestino 2:\tDestino 3:\tDestino 4:\tDestino 5:")
  for c in range(C):
      print("Carro",c+1,":\t","\t\t".join(str(round(x[o,d,c].X)) for d in range(D)))
```

```
Valor da solução ótima:  176960

Origem  1 :
        Destino 1:      Destino 2:      Destino 3:      Destino 4:      Destino 5:
Carro 1 :       50              30              0               0               0
Carro 2 :       30              40              0               0               0
Carro 3 :       35              45              0               0               0
Carro 4 :       0               0               0               0               0

Origem  2 :
        Destino 1:      Destino 2:      Destino 3:      Destino 4:      Destino 5:
Carro 1 :       0               0               60              40              0
Carro 2 :       0               0               0               60              0
Carro 3 :       0               0               40              40              0
Carro 4 :       35              0               15              35              0

Origem  3 :
        Destino 1:      Destino 2:      Destino 3:      Destino 4:      Destino 5:
Carro 1 :       0               0               0               0               40
Carro 2 :       0               0               50              0               10
Carro 3 :       0               0               15              0               60
Carro 4 :       5               55              0               0               70
```

Figura 2: Output final no Python.

6 Disponibilização dos arquivos

Todos os arquivos utilizados e mencionados estão disponíveis no GitHub: `https://gitlab.uspdigital.usp.br/gbortoli/ms-excel-solver-vs-python-gurobi`.

Referências

[1] C. To and S. Khakali, J. Seif. Optimal Allocation of Campus Parking Spaces to Personal Vehicles: A Case Study, *IISE Annual Conference and Expo 2022*, 2022. ISBN: 9781713858072.

[2] J. A. Rabi, M. O. Santos. *Aulas da Disciplina de Tópicos em Pesquisa Operacional - MAI5032*. MECAI - ICMC, São Carlos, 2002.