

Atividade 1

Lucas Guesser Targino da Silva - RA: 203534

Lucas Pedroso Cantanhede - RA: 202058

Luiz Fernando Bueno Rosa - RA: 221197

April 1, 2022

1 Modelo Matemático

1.1 Variáveis do Problema

- F : Número de fábricas das companhias.
- J : Número de clientes das companhias.
- L : Número de máquinas das fábricas.
- M : Número de matérias-primas das fábricas.
- P : Número de tipos de produtos produzidos pelas fábricas.
- $D_{j,p}$: Demanda do cliente j , em toneladas, do produto p .
- $r_{m,p,l}$: Quantidade de matéria-prima m , em toneladas, necessária para produzir uma tonelada do produto p na máquina l .
- $R_{m,f}$: Quantidade de matéria-prima m , em toneladas, disponível na fábrica f .
- $C_{l,f}$: Capacidade disponível de produção, em toneladas, da máquina l na fábrica f .
- $p_{p,l,f}$: Custo de produção, por tonelada, do produto p utilizando a máquina l na fábrica f .
- $t_{p,f,j}$: Custo de transporte, por tonelada, do produto p partindo da fábrica f até o cliente j .

1.2 Variáveis de Decisão

- $x_{p,l,f}$: quantidade, em toneladas, do produto p produzida na máquina l da fábrica f .
- $y_{p,f,j}$: quantidade, em toneladas, do produto p transportada da fábrica f para o cliente j .

1.3 Objetivo

Minimizar:

$$\sum_p \sum_l \sum_f p_{p,l,f} x_{p,l,f} + \sum_p \sum_f \sum_j t_{p,f,j} y_{p,f,j} \quad (1)$$

Sujeito a:

$$D_{j,p} = \sum_f y_{p,f,j} \quad \forall p \quad \forall j \quad (2)$$

$$R_{m,f} \geq \sum_p \sum_l r_{m,p,l} x_{p,l,f} \quad \forall m \quad \forall f \quad (3)$$

$$C_{l,f} \geq \sum_p x_{p,l,f} \quad \forall l \quad \forall f \quad (4)$$

$$\sum_l x_{p,l,f} = \sum_j y_{p,f,j} \quad \forall p \quad \forall f \quad (5)$$

$$x_{p,l,f}, y_{p,f,j} \geq 0 \quad \forall p \quad \forall l \quad \forall f \quad \forall j \quad (6)$$

A notação $\forall i$ acima (podendo i ser p, l, f, m, j) significa que a restrição se aplica a todos os valores do domínio discreto de i . Por exemplo, se o domínio de f for $\{1, 2, 3\}$, então a condição se aplica para $f = 1$, $f = 2$ e $f = 3$. Esse modelo ainda não usa [1].

Iremos explicar, brevemente, o que cada restrição significa:

1. A função objetivo 1 é calculada como sendo a soma do custo de produção (somatório à esquerda) e o custo de transporte (somatório à direita). O custo de produção é calculado como uma soma variando para cada produto p , máquina l e fábrica f da quantidade produzida, em toneladas, vezes o custo por tonelada $p_{p,l,f}$. O custo de transporte é calculado como uma soma variando para cada produto p , fábrica f e cliente j da quantidade transportada, em toneladas, vezes o custo de transporte $t_{p,f,j}$. Tendo em vista que queremos o menor custo possível, temos que o objetivo é minimizar o máximo possível o resultado dessa função.
2. A restrição 2 é a restrição da demanda onde restringimos que a demanda solicitada ($D_{j,p}$) por qualquer cliente j de um produto p , deve ser igual a somatória da quantidade de p de produtos produzidos e transportados para esse cliente j em todas as fábricas.
3. A restrição 3 é a restrição de produção de um produto p na fábrica f onde $r_{m,p,l}$ é a quantidade em toneladas do material m necessário para produzir uma tonelada de p na máquina l e $R_{m,f}$ é a quantidade de m na fábrica f .
4. A restrição 4 é a restrição da capacidade produtiva restringimos que a capacidade disponível, em toneladas, de produção de qualquer máquina l em uma fábrica f deve igual ou superior a somatória da quantidade, em toneladas, produzida de produtos p por essa mesma máquina l na fábrica f .
5. A restrição 5 é a restrição de compatibilidade entre o que foi produzido e que foi transportado. Nela consideramos que existe uma igualdade da somatória da quantidade, em toneladas, produzida em todas as fábricas de qualquer produto p em determinada máquina m com a somatória da quantidade, em toneladas, transportada para todos os clientes desse mesmo produto p produzido pela máquina m .
6. A restrição 6 é a restrição de não-negatividade, considerando que nem $x_{p,l,f}$ nem $y_{p,f,j}$ podem ser negativos independentes dos valores de p, l, f e j .

1.4 Geração das instâncias

Conforme solicitado no enunciado, geramos 10 instâncias aleatórias variando a quantidade de clientes $|J| = 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000$ e os demais parâmetros do enunciado, apresentados em 1.1, de maneira aleatória e uniforme seguindo as restrições dos coeficientes abaixo:

- $|F| \in [|J|, 2|J|]$
- $|L| \in [5, 10]$
- $|M| \in [5, 10]$
- $|P| \in [5, 10]$
- $D_{j,p} \in [10, 20]$
- $r_{m,p,l} \in [1, 5]$
- $R_{m,f} \in [800, 1000]$
- $C_{l,f} \in [80, 100]$
- $p_{p,l,f} \in [10, 100]$
- $t_{p,f,j} \in [10, 20]$

O módulo de geração de números aleatórios empregado foi `numpy.random` da biblioteca `numpy` [2]. Utilizou-se a semente de números aleatórios **17558175** para gerar, de forma que tenha-se as mesmas instâncias de problemas em diferentes execuções do código.

1.5 Configuração da Máquina

O problema foi executado num ideapad S145 81S90005BR: Lenovo IdeaPad S145 Notebook Intel Core i5-8265U (6MB Cache, 1.6GHz), 8GB DDR4-SDRAM, 460 GB SSD, Intel UHD Graphics 620.

O sistema operacional foi o Fedora 35 executando o Python 3.7.12 e Gurobi Optimizer v9.5.1rc2.

1.6 Resultados

Instância(J)	Qtde. de variáveis	Qtde. de restrições	Custo da Solução	Tempo de execução(s)
100	100320	3422	253714.11620693645	3.263497
200	668817	9698	567490.8153045126	18.304874
300	1441176	18141	744867.1786728669	60.564301
400	1909440	16070	994991.7809754083	97.239349
500	1771000	13700	790455.3006384401	50.536251
600	3981420	20454	1929256.3805292659	222.136828

Devido a limitações de recursos computacionais (memória), não foi possível rodar instâncias geradas com $J \geq 700$. Porém, os dados coletados já são suficientes para o propósito de aprendizado do trabalho.

1.7 Análise

1.7.1 Tempo de execução

A partir dos dados da tabela anterior, utilizamos uma regressão linear para estudar o crescimento do tempo de execução. A suspeita inicial é que encontraríamos um crescimento polinomial, isto é, que cresce em $O(n^k)$.

Mas, primeiramente, observe na tabela 1.6 de resultados acima a instância $J = 500$. É o primeiro caso que possa apontar um comportamento não-linear da nossa função, então um experimento interessante seria plotar os nossos tempos de execução ora levando em conta $J = 500$ ora não e tentar estimar visualmente o crescimento da nossa função.

Os gráficos a seguir são uma plotagem do tempo de execução em relação ao número de clientes $|J|$ juntamente com uma estimativa linear desses dados, gerada a partir do algoritmo de regressão linear.

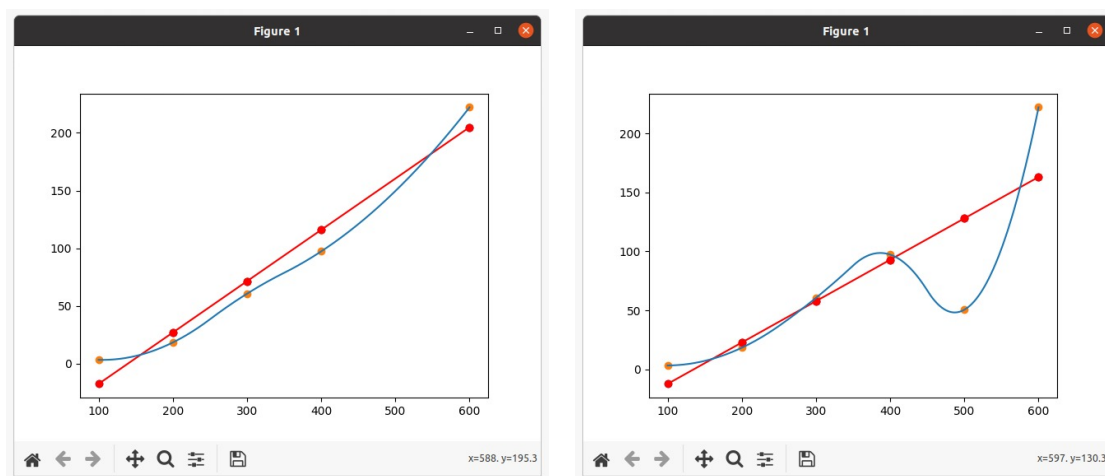


Figure 1: O gráfico da esquerda apresenta os dados sem o ponto $J = 500$ enquanto o da direita inclui esse ponto. Perceba que, sem o outlier, identificamos uma tendência linear.

No entanto, seriam necessários mais dados para podermos ter uma acurácia melhor de como o tempo de execução se comporta, pois se isolarmos os pontos $J = 400$, $J = 500$ e $J = 600$, percebemos uma tendência exponencial no gráfico.

Mas é certo afirmar que a tendência é linear na maior parte das amostras recolhidas, o que vai de acordo com as nossas suspeitas.

1.7.2 Minimização de custos

Podemos realizar as mesmas análises em relação à minimização da função objetivo. Hipotetizamos que, como a função objetivo é linear, então a sua minimização deve ser proporcional ao aumento da demanda (número dos clientes).

Conseguimos averiguar pelos dois gráficos anteriores que o crescimento da minimização da função objetivo possui a mesma tendência que o crescimento do tempo de execução, o que confirma nossa hipótese.

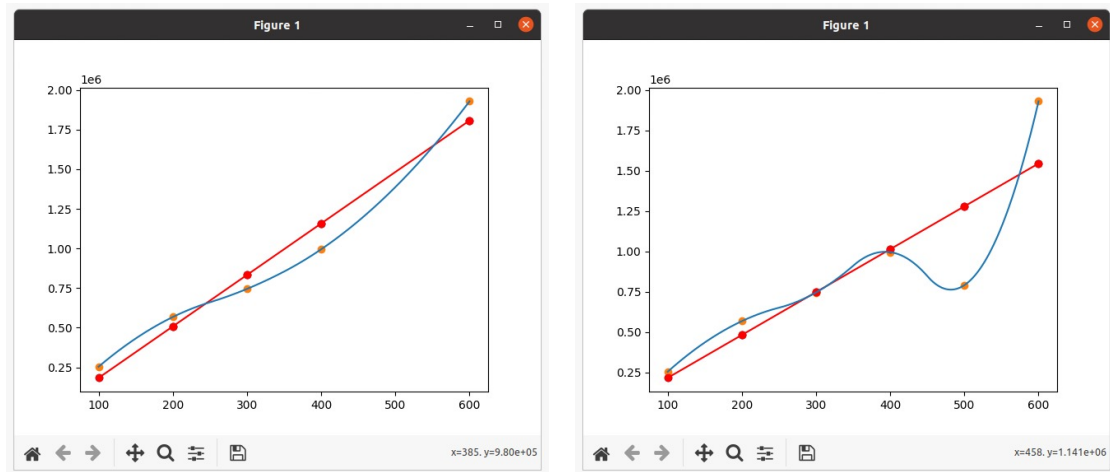


Figure 2: O gráfico da esquerda apresenta os dados sem o ponto $J = 500$ enquanto o da direita inclui esse ponto. Perceba que, sem o outlier, identificamos, novamente, uma tendência linear.

2 Referências

References

- [1] Gurobi Optimization. Gurobi optimization. <https://www.gurobi.com/>. [acessado em 2022-03-30].
- [2] NumPy Developers. Numpy documentation. <https://numpy.org/doc/stable/index.html>. [acessado em 2022-04-01].