Atividade 1 de MC859/MO824

Eduardo Barros Innarelli
e170161@dac.unicamp.br

Érico Iago Maldonado Faustino
e170609@dac.unicamp.br

1. Modelo Matemático

Considerando os parâmetros do problema de produção de papel, conforme ele fora enunciado, definimos as seguintes variáveis:

- $x_{p,l,f}$: quantidade de papel p produzida na máquina l da fábrica f, para cada $p \in P, l \in L, f \in F$;
- $y_{p,f,j}$: quantidade de papel p transportado da fábrica f até o cliente j, para cada $p \in P, f \in F, j \in J$.

Utilizando esses conjuntos, construímos o seguinte modelo de Programação Linear Inteira (PLI):

$$\min \sum_{p=1}^{P}\sum_{l=1}^{L}\sum_{f=1}^{F} p_{p,l,f} \cdot x_{p,l,f} + \sum_{p=1}^{P}\sum_{f=1}^{F}\sum_{j=1}^{J} t_{p,f,j} \cdot y_{p,f,j} \tag{1.1}$$

$$\text{s.a.} \sum_{l=1}^{L} x_{p,l,f} \geq \sum_{j=1}^{J} y_{p,f,j},\ \forall\ p \in \{1,\dots,P\}, f \in \{1,\dots,F\} \tag{1.2}$$

$$\sum_{f=1}^{F} y_{p,f,j} \geq D_{j,p},\ \forall\ p \in \{1,\dots,P\}, j \in \{1,\dots,J\} \tag{1.3}$$

$$\sum_{p=1}^{P} x_{p,l,f} \leq C_{l,f},\ \forall\ l \in \{1,\dots,L\}, f \in \{1,\dots,F\} \tag{1.4}$$

$$\sum_{p=1}^{P}\sum_{l=1}^{L} r_{m,p,l} \cdot x_{p,l,f} \leq R_{m,f},\ \forall\ m \in \{1,\dots,M\}, f \in \{1,\dots,F\} \tag{1.5}$$

$$x \in \mathbb{Z}, y \in \mathbb{Z} \tag{1.6}$$

Onde a função objetivo (1.1) visa minimizar o custo total de produção e transporte de papel; as restrições (1.2) relacionam as variáveis x e y, proibindo transportar mais do que é produzido de cada papel em cada fábrica; as restrições (1.3) fazem as demandas dos clientes serem atendidas; as restrições (1.4) e (1.5) limitam o quanto é produzido pela capacidade das máquinas e pela quantidade disponível de matéria-prima, respectivamente. Por fim, as restrições (1.6) são as restrições de integralidade.

Tabela 1: Resultados da modelagem do Problema de Produção de Papel implementado utilizando o software *Gurobi*

Número de clientes	Quantidade de Variáveis	Custo da Solução (PLI)	Tempo de Execução (PLI)	Custo da Solução (relaxado)	Tempo de Execução (relaxado)
50	36736	122244	$0,204s$	$122244,000$	$0,185s$
100	72360	148387	$0,433s$	$148387,000$	$0,386s$
150	279930	311300	$2,201s$	$311299,500$	$2,002s$
200	398520	486991	$3,578s$	$486986,875$	$2,146s$
250	997668	659287	$11,870s$	$659285,750$	$9,126s$
300	699060	539455	$12,227s$	$539453,567$	$6,426s$
350	1319472	724187	$15,610s$	$724184,500$	$12,170s$
400	1109075	590551	$16,362s$	$590549,000$	$9,024s$
450	1867320	657859	$22,438s$	$657857,250$	$29,620s$
500	2297750	743083	$59,191s$	$743082,750$	$54,870s$

2. Resultados Computacionais

O código foi escrito em um Jupyter Notebook, enviado junto com este relatório. Como exigido, utilizamos o software *Gurobi* (mais especificamente sua API Python) para implementar o modelo desenvolvido. Os testes foram rodados em uma máquina com um processador de 2.3 GHz e uma memória RAM de 16 GB. Na tabela 1, encontram-se os resultados obtidos tanto pela execução do modelo de PLI quanto pela execução do modelo de Programação Linear (PL), que consiste do primeiro com as restrições de integralidade relaxadas.

3. Análise dos Resultados

Como é possível observar e como era de se esperar, nota-se uma relação direta entre o número de clientes, a quantidade de variáveis e o tempo de execução, de forma que um cresce conforme o anterior aumenta. Vale ressaltar que, como instâncias geradas com os intervalos propostos pelo enunciado não possuíam soluções viáveis, foram utilizados os domínios reduzidos sugeridos pelo colega 'miguel.alarcon' no grupo da disciplina no Discord. Acreditamos que isso tenha impactado os resultados - mesmo a maior instância foi solucionada em menos de 1 minuto, o que surpreende dado que PLI é NP-Difícil.

Outra evidência de que os intervalos reduzidos podem ter influenciado nos tempos relatados está na pequena diferença entre os tempos de execução do modelo de PLI e do modelo de PL, apesar deste último ser menos restritivo e poder ser resolvido em tempo polinomial. Observe que o modelo relaxado chegou até a demorar mais para encontrar a solução ótima na instância com 450 clientes. Isto posto, acreditamos que seria interessante, para observar o comportamento assintótico de ambos os algoritmos, testar instâncias ainda maiores.

Por fim, constata-se também que o modelo de PL, apesar de normalmente gerar soluções inviáveis para o modelo de PLI, encontrou soluções com custos bem próximos (ou mesmo equivalentes, no caso das duas primeiras instâncias) dos custos das soluções ótimas do modelo original. Portanto, ao menos para instâncias nos domínios testados, a relaxação linear do modelo de PLI mostrou retornar bons limitantes duais para o problema. Isso também justifica a proximidade entre os tempos de execução dos dois modelos, assumindo que relaxações lineares são resolvidas para gerar planos de corte na resolução de um PLI.