

Relatório 3º projecto ASA 2023/2024

Grupo: TP021

Alunos: Maria Ramos (105875) e Guilherme Campos (106906)

Descrição do Problema e da Solução:

Problema:

O Problema pretende a maximização do lucro diário de uma fábrica de brinquedos de Natal. A fábrica produz diariamente uma variedade de brinquedos, e possui um máximo de produção global. Além disso, cada brinquedo tem uma capacidade máxima de produção e lucro associado. A empresa vende também pacotes especiais contendo três brinquedos distintos, proporcionando um lucro adicional. A produção total diária está sujeita a restrições de linha de montagem e corte da floresta boreal.

O objetivo é desenvolver um programa que estime o lucro máximo diário que a empresa pode obter, considerando as restrições de produção e a venda dos pacotes especiais.

Solução:

A solução utiliza a biblioteca PuLP em Python para formular e resolver o problema de programação linear de maximização, visando determinar o lucro máximo diário na produção e venda de brinquedos, considerando as suas restrições. O código cria variáveis de decisão para a quantidade de cada brinquedo produzido, impõe restrições individuais e de pacotes especiais, e finalmente, utiliza um solver, GLPK, para obter a solução ótima, indicando o lucro máximo possível.

Identificação das variáveis do problema:

- n : Número de diferentes brinquedos passíveis de serem produzidos.
- p : Número de pacotes especiais.
- max_toys : Número máximo de brinquedos que podem ser produzidos por dia.
- $\text{toys}[i]$: Quantidade do brinquedo i produzida por dia.
- $\text{profits}[i]$: Lista que armazena o lucro de cada brinquedo individual.
- $\text{package}[i]$: Quantidade do pacote especial j vendido por dia.
- $\text{package_profits}[j]$: Lista que armazena o lucro de cada pacote especial.
- $\text{capacity}[i]$: Capacidade de produção máxima para o brinquedo i .

Especificação do programa linear:

Função Objetivo:

$$\text{Maximize } \sum_{i=1}^n \text{profits}_i * \text{toys}_i + \sum_{j=1}^p \text{package_profits}_j * \text{packages}_j$$

Restrições:

1. **Restrições de Capacidade de Produção individual:** As variáveis toys_i e package_j devem ser não negativas e cada brinquedo não deve ser produzido além da sua capacidade máxima.

$$\begin{aligned} \text{toys}_i, \text{packages}_j &\geq 0 \\ \text{toys}_i &\leq \text{capacity}_i \end{aligned}$$

2. **Restrição para a capacidade máxima de produção de cada brinquedo toys:** Cada brinquedo individual toys_i e cada pacote especial pack_j não podem exceder a capacidade máxima de produção de brinquedos, max_toys . Considerando j que $\text{pack}_j(i)$ um dos $p(i)$ pacotes que contem toy_i :

$$\sum_{i=1}^n \text{toys}_i + \sum_{j=1}^p \text{packages}_j(i) \leq \text{max_toys}$$

3. **Restrição global para o máximo de brinquedos que a fábrica pode produzir:** Garante que a produção total de brinquedos, individuais ou em pacotes, não excede max_toys .

Relatório 3º projecto ASA 2023/2024

Grupo: TP021

Alunos: Maria Ramos (105875) e Guilherme Campos (106906)

$$\sum_{i=1}^n \text{toys}_i + \sum_{j=1}^P \text{packages}_j \leq \text{max_toys}$$

Análise Teórica:

A complexidade do programa é expressa em função de N e P, onde N é o número de diferentes brinquedos e P o número de pacotes especiais.

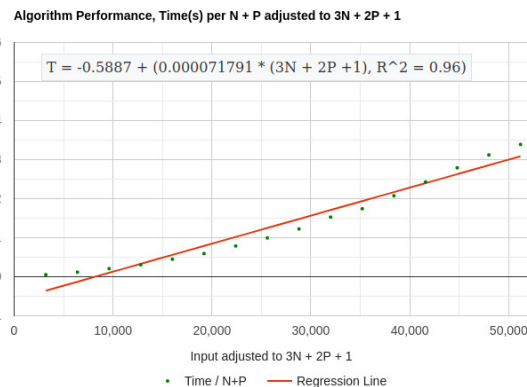
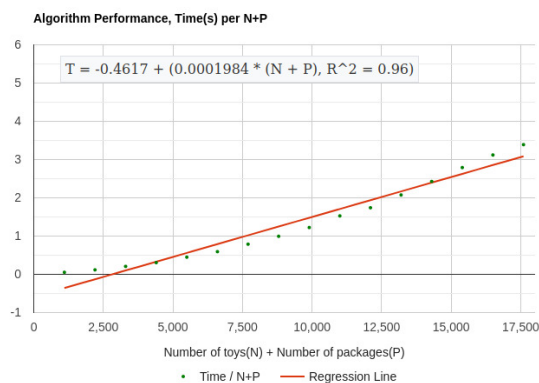
Para cada brinquedo, há uma variável e três restrições: um limite inferior, um limite superior e o número de pacotes que o contem. Para cada pacote, há uma variável e duas restrições: um limite inferior e um limite superior. Além das restrições individuais para brinquedos e pacotes, temos também uma restrição global que considera a capacidade máxima de produção da fábrica.

Conclui-se assim que o problema tem $O(N + P)$ variáveis e $O(3N + 2P + 1)$ restrições.

Avaliação Experimental dos Resultados:

N	P	N + P	3N + 2P + 1	Execution time(s)
1000	100	1100	3201	0.053
2000	200	2200	6401	0.117
3000	300	3300	9601	0.207
4000	400	4400	12801	0.306
5000	500	5500	16001	0.449
6000	600	6600	19201	0.593
7000	700	7700	22401	0.788
8000	800	8800	25601	0.994
9000	900	9900	28801	1.224
10000	1000	11000	32001	1.528
11000	1100	12100	35201	1.741
12000	1200	13200	38401	2.075
13000	1300	14300	41601	2.429
14000	1400	15400	44801	2.791
15000	1500	16500	48001	3.12
16000	1600	17600	51201	3.39

Para obter os dados de tempo de execução do programa foram utilizadas 16 instâncias em que o número de brinquedos e número de pacotes incrementou 1000 e 100, respetivamente. O tempo de execução de cada instância foi calculado através da média aritmética do tempo de execução de 5 testes cujo input N + P foi igual.



No primeiro gráfico, temos a relação entre o tamanho do input N + P com o respetivo tempo de execução, e a respetiva regressão linear que nos permite aproximar o comportamento do algoritmo em termos de eficiência para diferentes tamanhos de input não calculados. No segundo gráfico, temos um ajuste de variáveis consoante as restrições do problema analisadas na análise teórica. Para o mesmo tempo de execução, foi dado o plot de $3N + 2P + 1$. Podemos observar que gráficos de regressão linear são bastante análogos, o que demonstram a linearidade entre o tempo de execução o tamanho dos parâmetros do problema. Os valores dos coeficientes de determinação (R^2) para a regressão linear são aproximadamente 0.96, indicando um ajuste muito bom do modelo aos dados, e comprova a linearidade.