Investigação Operacional - Trabalho Prático I

Problema de empacotamento a uma dimensão com o modelo de fluxos em arcos

Lucas Rafael da Cunha Franco Robertson	A89467
Diogo José Ribeiro e Ribeiro	A106906
Carolina Silva Martins	A107285
Filipa Cangueiro Gonçalves	A107329

15 de março de 2025

0. Dados do problema

Tendo em conta que o maior número de inscrição deste grupo é 107329, temos os seguintes contentores disponíveis e itens a empacotar:

Contentores

Comprimento	Quantidade Disponível
11	Infinito
10	8
7	3

Tabela 1: Número de contentores de cada comprimento disponíveis

Itens

Comprimento	Quantidade
1	2
2	11
3	0
4	17
5	5

Tabela 2: Número de itens de cada comprimento disponíveis

A soma total dos comprimentos dos itens é:

$$1\times2 + 2\times11 + 3\times0 + 4\times17 + 5\times5 = 117$$

1. Formulação do problema

Este problema envolve a alocação de itens em contentores de capacidades limitadas, de forma a minimizar a soma dos comprimentos dos contentores utilizados. Existem três tipos de contentores disponíveis: contentores de capacidade 11, que podem ser utilizados sem restrição; contentores de capacidade 10, dos quais apenas 8 estão disponíveis; e contentores de capacidade 7, limitados a 3 unidades.

Os itens a serem alocados também são limitados em quantidade e possuem quatro tamanhos distintos: itens de tamanho 1 (2 disponíveis), tamanho 2 (11 disponíveis), tamanho 4 (17 disponíveis) e tamanho 5 (5 disponíveis). A solução deve garantir que todos os itens sejam acomodados dentro dos contentores respeitando estas restrições de capacidade e disponibilidade.
A solução deve ser obtida através do desenvolvimento de um modelo de fluxo de arcos.

2. Modelo de Fluxo de arcos

O modelo de fluxo de arcos é uma forma de representar problemas de otimização usando uma rede composta por nós e arcos, onde um recurso deve ser transportado de um ponto a outro de forma eficiente. Cada nó representa um estado, e os arcos representam conexões entre nós, definindo como o fluxo pode se movimentar. Em cada arco, pode haver restrições de capacidade e custos associados ao transporte do fluxo.

No contexto do problema de bin packing, podemos imaginar os itens como um fluxo que precisa ser distribuído nos contentores disponíveis. Cada estado de um *bin* pode ser visto como um nó de destino, e os arcos representam a possibilidade de alocar determinados itens nesses contentores, respeitando as suas capacidades. O objetivo é minimizar o número total de espaço ocupado por os contentores usados, garantindo que todos os itens sejam alocados sem ultrapassar a capacidade dos contentores.

Além disso, podem existir variáveis de retorno, que representam recursos que voltam a um estado anterior simplificando o grafo, pois evita a duplicação de fluxos idênticos.

Como também, variáveis de desperdício (todas elas de tamanho 1), que contabilizam o fluxo não aproveitado, como espaço não utilizado em contentores semi-completos.

O modelo de fluxo ajuda a estruturar o problema de maneira clara, permitindo a formulação de restrições matemáticas e a aplicação de algoritmos eficientes para encontrar soluções.

Ao transformar o problema num modelo de rede, podemos utilizar técnicas como programação linear inteira para encontrar a melhor forma de distribuir os itens, minimizando o desperdício de espaço e o número de contentores utilizados.

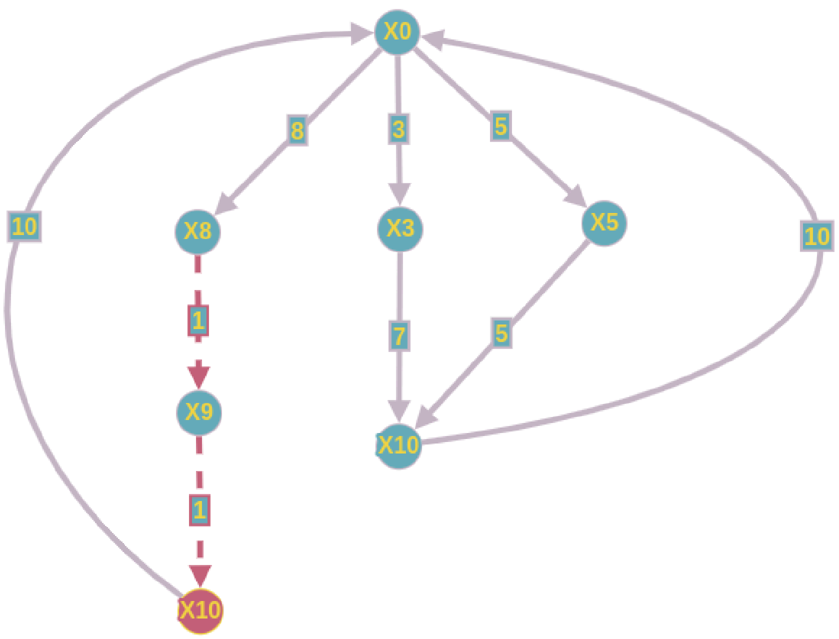

Figura 1: Exemplo de um grafo relativo ao modelo de fluxo de arcos

Este sendo um exemplo de um grafo de fluxo com contentores de 10 de tamanho onde os nós representam o estado de ocupação dos contentores e as arestas o espaço que o item ocupa.

Estão também representadas as variáveis de retorno que permitem reiniciar o fluxo.

E as variáveis de desperdício a vermelho que permitem completar contentores semi-Completos para permitir a contabilização dos contentores não completos.

Parâmetros

Antes de avançar para as variáveis e restrições é necessário mencionar os parâmetros do problema, como já foi mencionado anteriormente, o modelo de fluxo em arcos normalmente é usado para problemas de bin packing, pelo que precisamos de saber de antemão os K tamanhos diferentes de contentores e as suas quantidades, sendo elas representadas por :

- W_k, $k = 1, 2, \ldots, K$
- B_k, $k = 1, 2, \ldots, K$

Sendo que W_k representa o número de contentores de tamanho k, e B_k a quantidade que há disponível de contentores de tamanho W_k. Daqui também podemos tirar W_{max}, sendo ele igual à capacidade do maior contentor.

Outros parâmetros essenciais para o uso do modelo são os m tamanhos diferentes dos itens que queremos guardar e as suas quantidades, de um modo simétrico são representados por :

- w_i, $i = 1, 2, \ldots, m$
- b_i, $i = 1, 2, \ldots, m$

No caso do nosso problema atual podemos traduzir estas informações, de forma simpática para leitura, para:

- W_k, k = 7, 10, 11
- B_k, k = 7, 10, 11
- w_i, i = 1, 2, 4, 5
- b_i, i = 1, 2, 4, 5
- $W_{max} = 11$

Agora com estas bases podemos avançar para a explicação das variáveis.

Variáveis de decisão

As variáveis de decisão são do estilo $X_{d,e}$ que representarão tanto que espaço de cada contentor cada item ocupará, como também o número de contentores com cada capacidade a usar, sendo que os arcos dos items serão representado tendo em consideração este grafo G(V,A) com V = {0, 1, …, 11} e com A = {(e,d) : $0 \leq e < d \leq 11 \wedge d - e = w_i$, i = 1, 2, 4, 5} onde os vértices são a capacidade utilizada até ao momento de um qualquer contentor e os arcos o espaço ocupado por um dado objeto de tamanho *(d - e)*. As variáveis dos contentores serão representadas na forma {(k, 0) : k = Wk, k = 7, 10, 11} isto é, no contexto do modelo eles serão arcos do nodo com o valor da capacidade desse contentor para 0, sendo então chamados de arcos de contentor ou feedback, eles também nos garantem que um contentor só é contado como usado se houver um caminho de arcos de 0 a k a ser usado.

As variáveis pertencem todas a $\mathbb{N}$.

Variáveis de desperdício

No modelo também são considerados arcos de desperdício, representados matematicamente por tuplos do género $\{(d, d+1) : 0 \leq d \leq W_{max} - 1\}$, sendo então arcos de tamanho 1 que, ao contrário dos items que temos quantidades específicas, estes serão ilimitados.

Estas variáveis são essenciais ao modelo visto que podemos ficar sem items por arrumar antes de alcançar o nodo a partir do qual o arco de feedback iria partir, o que poderia dar-nos um resultado errado ou até nos impossibilitar de obter um resultado ao problema.

Restrições de quantidade de items

Esta restrição serve para garantir que cada item de cada tamanho tem um arco associado a ser usado. Matematicamente isto é:

$$\sum(X_{d,d+i}) = b_i, \ i = 1, 2, 4, 5$$

Esta restrição tem um pequeno detalhe, no caso de i = 1, poderíamos permitir que o somatório fosse maior ou igual a b_1 em vez de o forçar a ser igual, de modo a usarmos os arcos extra para as variáveis de desperdício o que seria bom para facilitar a resolução do problema à máquina, mas nós decidimos em vez disso considerar variáveis novas que representavam arcos iguais aos de items de tamanho 1, com a exceção no nome, designando-os então por L em vez de X, isto só nos ajuda na leitura do problema ao separar os items de tamanho 1 das variáveis de desperdício e a verificar no final se houve algum desperdício ou não, procurando simplesmente por variáveis com valores diferentes de 0 começadas por L.

Restrições de quantidade de contentores

Esta restrição tem como objetivo não permitir que se use mais contentores do que aqueles que nos foram alocados, matematicamente falando:

$$X_{k,0} \leq W_k, \ k = 7, 10, 11$$

Restrições de controlo de fluxo

Estas restrições garantem que o número de arcos a partir de um nodo tem de ser igual ao número de arcos que lá chegaram, excepto se um arco está a partir do nodo 0, e através dos arcos de contentor garante-se também que por cada arco que saiu de 0, houve um contentor que foi utilizado.

$$\sum(X_{d,e}) = \sum(X_{e,f}),\ \forall_{(d,e),(e,f)} \in A \text{ ou arco de contentor}$$

3. Ficheiro de input e explicação do modelo

Função objetivo:

min: 11 * X11_0 + 10 * X10_0 + 7 * X7_0

Esta função representa que o modelo deve minimizar o uso das variáveis de retorno (X11_0, X10_0,X7_0) pois as mesmas significam que um novo container está a ser utilizado.

Como também se multiplica as variáveis de retorno pelo respetivo peso para obrigar o modelo a minimizar não só o número de contentores usados como também utilizar a menor quantidade de espaço ocupado pelos contentores possível (esse sim sendo o objetivo referido no enunciado)

Ficheiro de input:

```
min: 11*X11_0 + 10*X10_0 + 7*X7_0;

/* Restrições de quantidade de itens */

X0_1 + X1_2 + X2_3 + X3_4 + X4_5 + X5_6 + X6_7 + X7_8 + X8_9 + X9_10 + X10_11 = 2;

X0_2 + X1_3 + X2_4 + X3_5 + X4_6 + X5_7 + X6_8 + X7_9 + X8_10 + X9_11 = 11;

X0_4 + X1_5 + X2_6 + X3_7 + X4_8 + X5_9 + X6_10 + X7_11 = 17;

X0_5 + X1_6 + X2_7 + X3_8 + X4_9 + X5_10 + X6_11 = 5;

// Restrições de fluxo

X0_1 + X0_2 + X0_4 + X0_5 = X11_0 + X10_0 + X7_0;
```

```
L1_2 + X1_2 + X1_3 + X1_5 + X1_6 = X0_1;

L2_3 + X2_3 + X2_4 + X2_6 + X2_7 = X0_2 + X1_2 + L1_2;

L3_4 + X3_4 + X3_5 + X3_7 + X3_8 = X1_3 + X2_3 + L2_3;

L4_5 + X4_5 + X4_6 + X4_8 + X4_9 = X0_4 + X2_4 + X3_4 + L3_4;

L5_6 + X5_6 + X5_7 + X5_9 + X5_10 = X0_5 + X1_5 + X3_5 + X4_5 + L4_5;

L6_7 + X6_7 + X6_8 + X6_10 + X6_11 = X1_6 + X2_6 + X4_6 + X5_6 + L5_6;

L7_8 + X7_8 + X7_9 + X7_11 + X7_0 = X2_7 + X3_7 + X5_7 + X6_7 + L6_7;

L8_9 + X8_9 + X8_10 = X3_8 + X4_8 + X6_8 + X7_8 + L7_8;

L9_10 + X9_10 + X9_11 = X4_9 + X5_9 + X7_9 + X8_9 + L8_9;

L10_11 + X10_11 + X10_0 = X5_10 + X6_10 + X8_10 + X9_10 + L9_10;

L10_11 + X6_11 + X7_11 + X9_11 + X10_11 = X11_0;


/* Restrição de uso dos bins */

X10_0 <= 8;

X7_0 <= 3;


/* Definir todas as variáveis como inteiras */

int X0_1, X1_2, X2_3, X3_4, X4_5, X5_6, X6_7, X7_8, X8_9, X9_10, X10_11;

/* Variáveis de alocação para os itens de tamanho 2 */

int X0_2, X1_3, X2_4, X3_5, X4_6, X5_7, X6_8, X7_9, X8_10, X9_11;

/* Variáveis de alocação para os itens de tamanho 4 */

int X0_4, X1_5, X2_6, X3_7, X4_8, X5_9, X6_10, X7_11;

/* Variáveis de alocação para os itens de tamanho 5 */

int X0_5, X1_6, X2_7, X3_8, X4_9, X5_10, X6_11;

/* Variáveis de desperdício */

int L1_2, L2_3, L3_4, L4_5, L5_6, L6_7, L7_8, L8_9, L9_10, L10_11;

int X11_0, X10_0, X7_0;
```

4. Ficheiro de output:

Value of objective function: 117.00000000

Actual values of the variables:

X11_0	7
X10_0	4
X0_1	1
X6_7	1
X0_2	10
X1_3	1
X2_6	5
X3_7	1
X6_10	4
X7_11	7
X2_7	5

5. Interpretação da solução ótima

Observando os valores acima (X11_0,X10_0,X7_0) consegue-se perceber quantos contentores foram utilizados e de cada tipo:

Comprimento dos Contentores	Contentores utilizados
11	7
10	4
7	0

Tabela 3: Número de contentores de cada tipo utilizados na solução ótima

O valor da função objetivo corresponde à soma do comprimento dos contentores utilizados (117), que é exatamente igual à soma dos tamanhos dos itens, o que significa que a solução tem 0 desperdicio de espaço. Ou seja, todos os contentores ficaram 100% cheios, o que faz sentido pois 11*7 + 4*10 é igual a 117.

Observando os valores da solução ótima, podemos representá-la através do seguinte grafo:

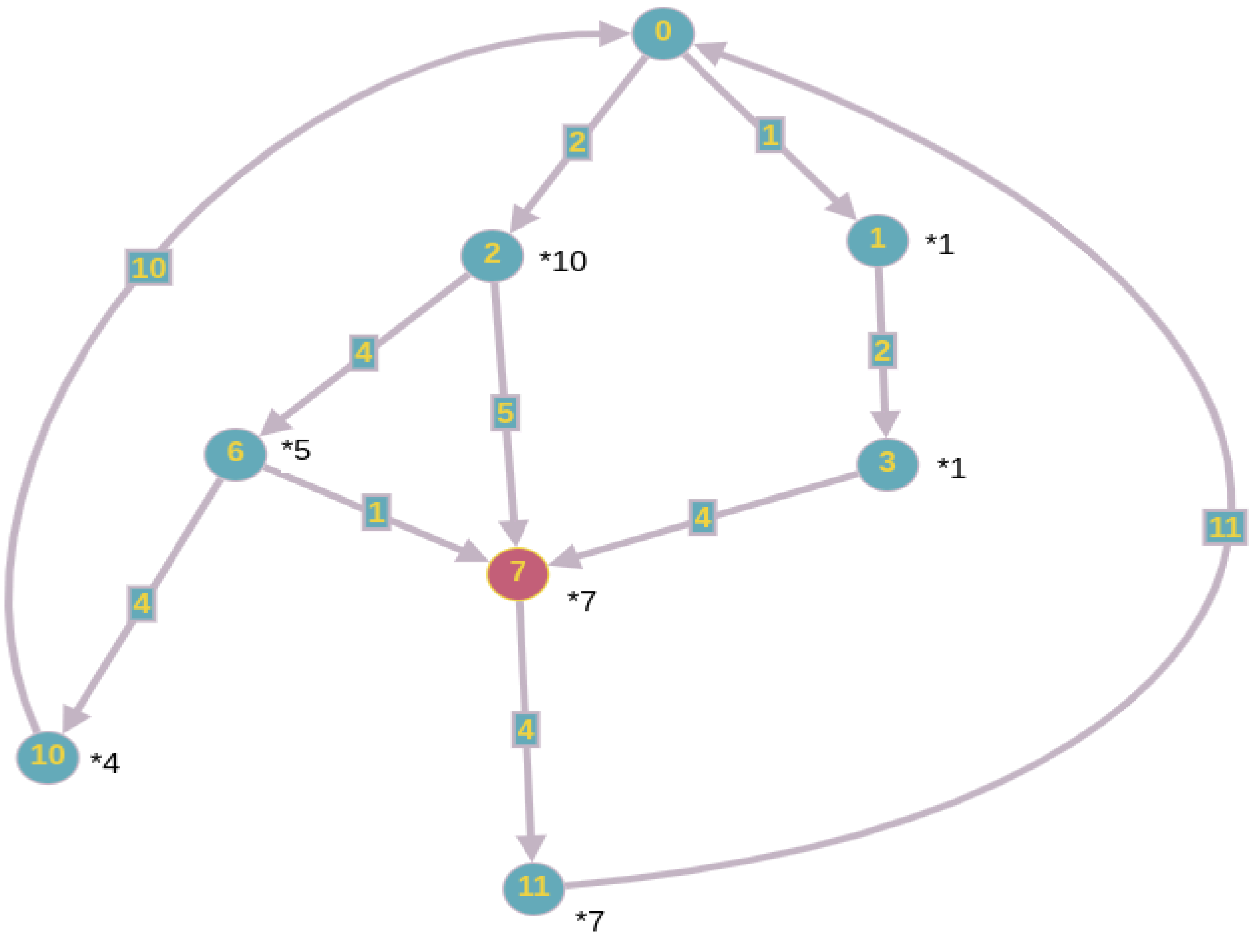

Figura 2: Grafo de fluxo da solução ótima

Observando o grafo, conseguimos deduzir o seguinte plano de empacotamento:

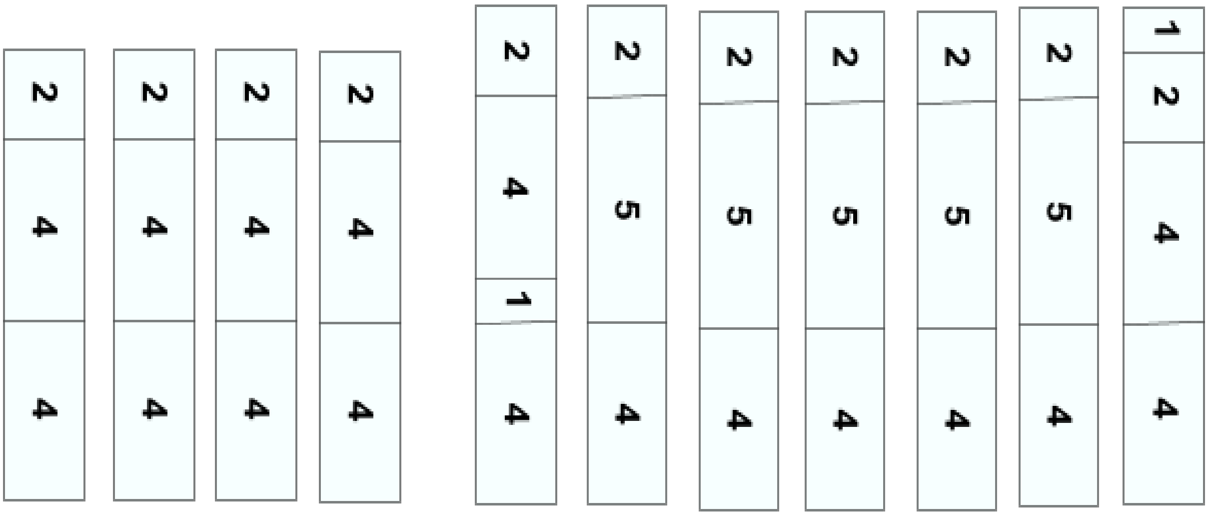

Figura 3: Plano de empacotamento com base na solução ótima

6. Validação do modelo

Para validar o modelo temos que ver se certos pressupostos foram obedecidos no modelo LP e se a solução os reflete.

Primeiro podemos reparar que todos os itens foram utilizados sem haver usos a mais ou a menos de qualquer um dos itens.

Segundo podemos reparar que a solução ideal não deu um valor inferior à soma dos itens, pois nesse caso mostraria uma irregularidade com o objetivo do modelo LP.

E por último os limites de stock dos contentores de tamanho 10 (8 disponíveis) e 7 (3 disponíveis) não foram ultrapassados pois apenas 4 dos de 10 de tamanho foram utilizados e nenhum de tamanho 7 foi utilizado.

Para além de validarmos os pressupostos também podemos perceber que o nosso modelo LP dá como solução objetiva o mesmo valor da soma dos itens, o que significa que a solução é ótima.
Também podemos perceber que usamos o mínimo de contentores precisos pois caso usássemos os contentores de maior tamanho seriam precisos pelo menos 11 contentores esses de tamanho 11 (11*11 = 121) pois caso se usasse apenas 10 de tamanho 11 seria impossível a alocação dos itens todos (11*10 = 110 ; 110 < 117).

Como na nossa solução usou-se 11 contentores, 7 de tamanho 11 e 4 de tamanho 10, podemos perceber que também usamos o mínimo de containers possíveis para a nossa quantidade de itens.

7.Conclusão

Ao longo da elaboração deste projeto fomos capazes de resolver o problema proposto obtendo assim um conhecimento extenso do modelo de fluxos em arcos. Este conhecimento permitiu-nos implementar um programa que utilizasse o modelo referido para resolver o problema de empacotamento de forma a minimizar a soma dos comprimentos dos contentores utilizados.