Escola de Engenharia
Universidade do Minho

2023/2024

Trabalho Prático

Investigação Operacional

Relatório do Trabalho 1

Pedro Filipe Maneta Pinto - a104173

Marco António Fernandes Brito - a104187

Pedro Miguel de Abreu Arguinha - a104351

Pedro Seabra Vieira - a104352

Tomás Henrique Alves Melo - a104529

Índice

Índice de Figuras

Índice de Tabelas

1 Formulação do Problema

A formulação deste problema envolve a definição de variáveis de decisão, restrições e uma função objetivo. As variáveis de decisão representam as decisões sobre quais itens serão atribuídos a quais contentores, enquanto as restrições garantem que a capacidade de cada contentor não seja excedida. A função objetivo busca maximizar a eficiência do empacotamento, medida pela soma dos comprimentos dos contentores utilizados.

Para o problema de empacotamento em uma dimensão com contentores de diferentes capacidades, a formulação pode ser adaptada para acomodar essa característica específica. Além disso, dependendo da abordagem escolhida, como o modelo de 'um-corte' de Dyckhoff, as variáveis e restrições podem ser formuladas de forma ligeiramente diferente para refletir as operações de corte efetuadas nos objetos grandes para obter os itens desejados.

De acordo com o maior número de inscrição do grupo (104529), obtemos os seguintes dados relativos ao comprimento e à quantidade dos itens e dos contentores:

A = 0

B = 4, par → $k_1 = 0$

C = 5, ímpar → $k_2 = C + 8 \leftrightarrow k_2 = 5 + 8 = 13$

D = 2, par → $k_3 = 0$

E = 9, ímpar → $k_4 = E + 8 \leftrightarrow k_2 = 9 + 8 = 17$

Contentores	
Comprimento	**Quantidade disponível**
11	Ilimitada
10	5
7	3

Tabela 1 – Quantidade de contentores

Itens	
Comprimento	**Quantidade**
1	0
2	13
3	0
4	17
5	5

Tabela 2 – Quantidade de itens

Total de comprimento dos itens a empacotar = 13 * 2 + 17 *4 + 5 * 5 = **119**

2 Modelo de Programação Linear

2.1 Variáveis de decisão

Cortes Possíveis	Item Requerido	Resíduo
corte11_2	2	9
corte11_4	4	7
corte11_5	5	6
corte10_2	2	8
corte10_4	4	6
corte10_5	5	5
corte9_2	2	7
corte9_5	5	4
corte8_2	2	6
corte8_4	4	4
corte8_5	5	3
corte7_4	4	3
corte7_5	5	2
corte6_4	4	2
corte6_5	5	1
corte5_2	2	3
corte5_4	4	1
corte4_2	2	2
corte3_2	2	1

Tabela 3 – Variáveis de decisão

2.2 Parâmetros

Os parâmetros são detalhados nas Tabelas 1 e 2, anteriormente mencionadas. Temos uma quantidade ilimitada de contentores com comprimento 11 (bin_x), 5 contentores com comprimento 10 (bin_y), e 3 contentores com comprimento 7 (bin_z). Adicionalmente, temos disponíveis 13 itens de comprimento 2, 17 itens de comprimento 4 e 5 itens de comprimento 5.

$$\boldsymbol{bin\_x = corte11\_2 + corte11\_4 + corte11\_5};$$

$$\boldsymbol{bin\_y = corte10\_2 + corte10\_4 + corte10\_5};$$

$$\boldsymbol{bin\_z = corte7\_4 + corte7\_5};$$

Com estas expressões, podemos concluir que:

- bin_x será o somatório dos itens 2, 4 e 5, através dos cortes das peças de comprimento 11.
- bin_y será o somatório dos itens 2, 4 e 5, através dos cortes das peças de comprimento 10.
- bin_z será o somatório dos itens 2, 4 e 5, através dos cortes das peças de comprimento 7 (lembrando que o corte 'corte7_2' da peça de comprimento 7 dá origem às mesmas peças que o corte 'corte7_5', então deixamos só o corte7_5 para evitar redundância).

2.3 Função Objetivo

O objetivo é minimizar a soma do comprimento dos contentores utilizados.

$$\min = 11 \times bin\_x + 10 \times bin\_y + 7 \times bin\_z$$

2.4 Restrições

As restrições que obtivemos garantem que os itens sejam empacotados corretamente nos contentores, respeitando as capacidades dos contentores e as dependências entre os diferentes comprimentos de itens.

O nosso primeiro pensamento foi implementar as restrições para os contentores:

As restrições dos contentores garantem que a quantidade total de itens empacotados em cada tipo de contentor (bin_x, bin_y, bin_z) não exceda a capacidade máxima dos contentores.

$$bin\_x \geq 0;$$
$$bin\_y \leq 5;$$
$$bin\_z \leq 3;$$

Com estas restrições, conseguimos determinar que existe um número ilimitado de contentores de capacidade 11, que deve haver no máximo 5 contentores de capacidade 10 e no máximo 3 contentores de capacidade 7.

Seguidamente, foi necessário implementar restrições sobre os cortes dos itens.

$$corte11\_2 + corte10\_2 + corte7\_5 + corte9\_2 + corte8\_2 + corte6\_4 + corte5\_2 + corte3\_2 + 2\,corte4\_2 = 13;$$

A primeira restrição implementada, garante que a soma total de todos os possíveis cortes, que podem gerar um item de comprimento 2, seja igual ao número de itens de comprimento 2 que precisamos acomodar nos contentores. Neste caso, o número é 13. Como referimos anteriormente, todos esses cortes resultam em um item de comprimento 2. Para esclarecer o termo 2 $corte4\_2$, neste cenário de corte, serão criados dois itens de comprimento 2.

$$\begin{aligned} &\boldsymbol{corte11\_4 + corte10\_4 + corte7\_4 + corte9\_5 + 2\,corte8\_4} \\ &\quad \boldsymbol{+ corte6\_4 + corte5\_4 - corte4\_2 = 17}; \end{aligned}$$

A segunda restrição que implementamos garante que a soma total de todos os possíveis cortes, que podem gerar um item de comprimento 4, seja igual ao número de itens de comprimento 4 que precisamos acomodar nos containers. Neste caso, o número máximo é 17. Como mencionamos anteriormente, todos esses cortes resultam em um item de comprimento 4.

No entanto, há uma exceção, subtraímos o corte 4_2. Isso ocorre porque podemos usar um item de comprimento 4 para originar um corte menor, neste caso, de comprimento 2.

$$\begin{aligned} &\boldsymbol{corte11\_5 + 2\,corte10\_5 + corte7\_5 + corte9\_5 + corte8\_5 + corte6\_5} \\ &\quad \boldsymbol{- corte5\_2 - corte5\_4 = 5} \end{aligned}$$

Esta restrição é semelhante à anterior, com a diferença de que desta vez é feita para os cortes, que originam itens de comprimento 5, com uma quantidade igual a 5.

$$\boldsymbol{corte11\_2 \geq corte9\_2 + corte9\_5};$$

$$\boldsymbol{corte10\_2 \geq corte8\_2 + corte8\_4 + corte8\_5};$$

$$\boldsymbol{corte11\_5 + corte10\_4 + corte8\_2 \geq corte6\_4 + corte6\_5};$$

$$\boldsymbol{corte10\_5 + corte9\_5 + corte8\_5 + corte7\_5 \geq corte5\_2 + corte5\_4};$$

$$\boldsymbol{corte9\_5 + corte8\_4 + corte7\_4 + corte6\_4 \geq corte4\_2};$$

$$\boldsymbol{corte8\_5 + corte7\_4 + corte5\_2 \geq corte3\_2};$$

Garante que o resíduo ou a peça requerida gerada por um corte seja maior ou igual ao próprio corte dessa mesma peça. Por exemplo, considere o corte11_2, que gera 2 itens de comprimentos 2 e 9. A soma dos possíveis cortes com essas peças deve ser inferior ou igual à peça inicial de comprimento 11. Esta restrição é fundamental para otimizar o uso do material durante o processo de corte.

2.5 Ficheiro Input

```
/* Função Objetivo */
min : 11 bin_x + 10 bin_y + 7 bin_z;

/* Restrição de quantidade itens */
corte11_2 + corte10_2 + corte9_2 + corte8_2 + corte7_5 + corte6_4 + corte5_2 + 2 corte4_2 + corte3_2 = 13;
corte11_4 + corte10_4 + corte9_5 + 2 corte8_4 + corte7_4 + corte6_4 + corte5_4 - corte4_2 - corte4_3 = 17;
corte11_5 + 2 corte10_5 + corte9_5 + corte8_5 + corte7_5 + corte6_5 - corte5_2 - corte5_4 = 5;

/* Restrição de cortes */
corte11_2 ⩾ corte9_2 + corte9_5;
corte11_5 + corte10_4 + corte8_2 ⩾ corte6_4 + corte6_5;
corte10_2 ⩾ corte8_2 + corte8_4 + corte8_5;
corte10_5 + corte9_5 + corte8_5 + corte7_5 ⩾ corte5_2 + corte5_4;
corte9_5 + corte8_4 + corte7_4 + corte6_4 ⩾ corte4_2;
corte8_5 + corte7_4 + corte5_2 ⩾ corte3_2;


/* Restrição de contentores */
bin_x = corte11_2 + corte11_4 + corte11_5;
bin_y = corte10_2 + corte10_4 + corte10_5;
bin_z = corte7_4 + corte7_5;

bin_x ⩾ 0;
bin_y ⩽ 5;
bin_z ⩽ 3;

/* Váriaveis */
int corte11_2, corte11_4, corte11_5;
int corte10_2, corte10_4, corte10_5;
int corte9_2, corte9_5; /* 6_4 = 6_5 */
int corte8_2, corte8_4, corte8_5;
int corte7_4, corte7_5;
int corte6_4, corte6_5;
int corte5_2, corte5_4;
int corte4_2;
int corte3_2;
```

Figura 1 – Ficheiro input

3 Solução Ótima

A solução ideal para este problema é a que minimiza o espaço necessário para acomodar os itens nos contentores. Estratégias que maximizem a utilização do espaço e minimizem ou eliminem por completo o espaço não utilizado dentro dos contentores devem ser priorizadas para otimizar o empacotamento, também passa por reduzir o número de cortes feitos sobre as peças maiores para obter os itens desejados.

Itens:	Tamanho	Quantidade	Total por Item
Itens1	1	0	0
Itens2	2	13	26
Itens3	3	0	0
Itens4	4	17	68
Itens5	5	5	25
Total:		35	119

Figura 2 – Quantidade total de itens

Entre todas as combinações possíveis para acomodar os itens, buscamos aquelas cuja soma total de comprimentos seja o mais próxima possível de 119, idealmente atingindo exatamente esse valor. Utilizando o software LPSolve, identificamos três soluções viáveis. A Solução A representa a melhor opção dentro das restrições e parâmetros previamente estabelecidos. As Soluções B e C resultam de ajustes nessas restrições para explorar diferentes cenários: a Solução C exclui o uso de contentores de tamanho 7, enquanto a Solução B reduz o uso de contentores de tamanho 10 e aumenta os contentores de tamanho 11.

	A	B	C
bin 11	6	7	7
bin 10	5	4	5
bin 7	1	1	0
Total	123	124	127

Figura 3 – 3 melhores soluções

```
Value of objective function: 123.00000000

Actual values of the variables:
bin_x                          6
bin_y                          5
bin_z                          1
corte11_2                      6
corte10_2                      4
corte9_2                       0
corte8_2                       1
corte7_5                       1
corte6_4                       1
corte5_2                       0
corte4_2                       0
corte3_2                       0
corte11_4                      0
corte10_4                      1
corte9_5                       6
corte8_4                       3
corte7_4                       0
corte5_4                       3
corte4_3                       0
corte11_5                      0
corte10_5                      0
corte8_5                       0
corte6_5                       1
```

Figura 4 – Output A reduzido do LPSolve

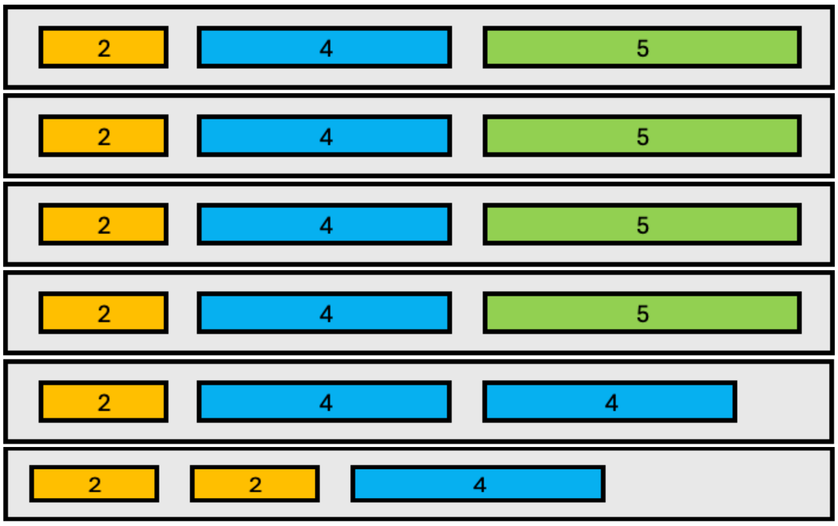
Analisando as três soluções obtidas, podemos descartar as soluções B e C, uma vez que a restante possui um tamanho inferior aos valores 127 e 124. Portanto, a Solução A é considerada a melhor opção, visto que oferece o menor comprimento total, e também mais próximo do valor ideal de 119.

4 Validação do Modelo

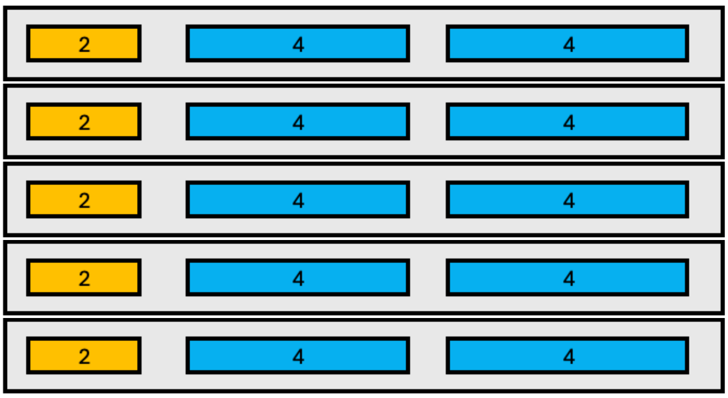
A avaliação do modelo é apresentada de modo a comprovar que as variáveis e restrições implementadas, de facto produzem o resultado esperado, possibilitando a obtenção da solução ótima.

Esta representação gráfica permite visualizar de forma mais intuitiva como os itens são distribuídos pelos contentores.

Em projetos que visam simular um elemento da realidade, a validação do modelo é um passo crucial para garantir que o programa/solução esteja a desempenhar da forma como pretendemos que funcione. No entanto, neste problema em específico, não estamos necessariamente a tentar replicar um ambiente real, e sim, à procura de uma solução ótima com base em dados fornecidos pelo enunciado, obtidos de acordo com o maior número de aluno entre elementos do grupo (104529).



Distribuição dos contentores de comprimento 11



Distribuição dos contentores de comprimento 10

Distribuição dos contentores de comprimento 7

5 Referências Bibliográficas

Dyckhoff, H. (1981). A New Linear Programming Approach to the Cutting Stock Problem. Operations Research, 29(6), 1092-11041

Alves, C. M. M. (Fev 2005). Cutting and Packing: Problems, Models and Exact Algorithms. Recuperado de https://core.ac.uk/download/pdf/55603804.pdf