Universidade Do Minho
Engenharia Informática
Investigação Operacional
A91677 - Vicente Castro; A91971 - Nuno Pereira
A94662 - Bruno Ribeiro; A100600 - Diogo Barros

TP1:
Problema de Empacotamento

23/03/2024

1 - Formulação do Problema

Neste trabalho prático, o problema consiste em empacotar itens de vários comprimentos em contentores de diferentes comprimentos. Existe um contentor com comprimento de 7 unidades, um contentor com comprimento de 10 unidades e um número ilimitado de contentores com comprimento de 11 unidades. Os itens a serem armazenados têm comprimentos de 2, 4 e 5 unidades.

O objetivo é **minimizar o desperdício de material/espaço**, empacotando os itens nos contentores de maneira eficiente, respeitando as capacidades dos contentores e do modelo de 'um-corte' de Dyckhoff.

O maior nº de estudante do grupo é o **100600 (xABCDE)** e com este número obtemos seguintes tabelas de quantidades dos vários contentores e itens.

CONTENTORES	
COMPRIMENTO	**QUANTIDADE DISPONÍVEL**
11	ILIMITADA
10	1
7	1

ITENS	
COMPRIMENTO	**QUANTIDADE DISPONÍVEL**
1	0
2	8
3	0
4	2
5	5

Fig. 1 - Tabelas de quantidades de contentores e de itens

2 - Modelo

2.1 - Variáveis de Decisão

i $\in \{3, 4, 5, 6, 7, 8, 9, 10, 11\}$;
j $\in \{2, 4, 5\}$;
O **xij**, representa o "um-corte" feito de tamanho **j**, a um determinado compartimento de tamanho **i**.

2.2 - Parâmetros

xij : partição resultante de um "um-corte".
O parâmetro a respeitar por parte de **xij**, é que por cada partição feita de tamanho **j**, num espaço de tamanho **i**, deve existir um item de tamanho **j** que **preencha uma ou ambas as partições** geradas pela operação de "um-corte".

2.3 - Função Objetivo

OBJETIVO: Minimizar a soma dos comprimentos dos contentores usados.
min: $\sum_{i=11,10,7} \sum_{j=2,4,5} ci * xij$;
ci: simboliza o custo de ocupar um espaço de tamanho **i**. Neste caso, o custo simboliza o comprimento total de um contentor de tamaho **i.**
xij: representa o "um-corte" feito de tamanho **j**, a um determinado compartimento de tamanho **i**

2.4 - Restrições

```
/* Objective function */
min: 11 x115 + 11 x114 + 11 x112 + 10 x105 + 10 x104 + 10 x102 + 7 x75 + 7 x74 + 7 x72 - 7 x92 - 7 x114;

/* Variable bounds */
R1.RESIDUAL1: x65 + x54 + x32 >= 0;
R2.RESIDUAL3: x85 + x74 + x52 - x32 >= 0;
//SUBTRAÇÃO DE x32, permite que se consiga ainda empacotar no contentor de comprimento de 3, um item de 2//
R3.RESIDUAL6: x115 + x104 + x82 - x65 - x64 - x62 >= 0;
//SUBTRAÇÃO DE x65, x64, x62, permite que se consiga ainda empacotar no contentor de comprimento de 6, itens de 2, 4, 5//
R4.RESIDUAL8: x102 - x85 - x84 - x82 >= 0;
//SUBTRAÇÃO DE x85, x84, x82, permite que se consiga ainda empacotar no contentor de comprimento de 8, itens de 2, 4, 5//
R5.RESIDUAL9: x112 - x95 - x94 - x92 >= 0;
//SUBTRAÇÃO DE x95, x94, x92, permite que se consiga ainda empacotar no contentor de comprimento de 9, itens de 2, 4, 5//

R6.EMPACOTAMENTO.ITEM2: x112 + x102 + x92 + x82 + x75 + x64 + x62 + x52 + 2 x42 + x32 >= 8;
R7.EMPACOTAMENTO.ITEM4: x114 + x104 + x95 + x94 + 2 x84 + x74 + x64 + x62 + x54 - x42 >= 2;
//SUBTRAÇÃO DE x42, permite que se consiga ainda empacotar no contentor de comprimento de 4, um item de 2//
R8.EMPACOTAMENTO.ITEM5: x115 + 2 x105 + x95 + x94 + x85 + x75 + x72 + x65 - x54 - x52 >= 5;
//SUBTRAÇÃO DE x54, x52, permite que se consiga ainda empacotar no contentor de comprimento de 8, itens de 2, 4//

R9.EMPACOTAMENTO.CONTENTOR11: x115 + x114 + x112 >= 0;
R10.EMPACOTAMENTO.CONTENTOR10: x105 + x104 + x102 <= 1;
R11.EMPACOTAMENTO.CONTENTOR7: x75 + x74 + x72 - x114 - x92 <= 1;
//SUBTRAÇÃO DE x114, x92, permite que se consiga ainda empacotar no contentor de comprimento de 11 e 9, itens de 2, 4, limitando o uso de contentores 7, visto que é <= 1//

int x32, x42, x52, x54, x62, x64, x65, x72, x74, x75, x82, x84, x85, x92, x94, x95, x102, x104, x105, x112, x114, x115;
```

Fig. 2 - Ficheiro Input com a função-objetivo e as suas restrições.

- Nas restrições do tipo **R1, R2, R3, R4, R5**, estamos a demonstrar todos os **tamanhos residuais** que podem ocorrer com o uso deste modelo. Nas residuais 3, 6, 8 e 9, encontrámos algumas **subtrações.** Estas subtrações permitem **"lembrar" ao LPSolve**, que já existem **cortes/empacotamentos de itens**, que conseguem anular aquele resíduo, fazendo com que o empacotamento seja o melhor possível e que não haja valores somados a mais.

- Nas restrições do tipo **R6, R7, R8**, juntámos todos os **"um-cortes" possíveis para cada item do projeto.** Nestas restrições, são feitas limitações para que os valores disponíveis de cada item não sejam ultrapassados.

- Nas restrições do tipo **R9, R10, R11**, temos os **cortes** a fazer a todos os contentores. Aqui assume-se que ao fazer um-corte a 1 contentor, esse contentor, já conta como utilizado. Logo, com estas restrições pretende-se **limitar ou não**, o uso de contentores, consoante o **nº de cada tipo de contentor disponível.**

3 - Solução Ótima

Objective | Constraints | Sensitivity

Variables	MILP ...	MILP ...	MILP ...	MILP ...	result
	55	52	51	50	50
x115	1	1	0	0	0
x114	5	5	5	3	3
x112	0	1	0	0	0
x105	1	1	1	1	1
x104	0	0	0	0	0
x102	0	0	0	0	0
x75	2	0	3	3	3
x74	0	0	0	1	1
x72	0	0	0	0	0
x92	0	0	0	0	0
x65	1	1	0	0	0
x54	0	0	0	0	0
x32	0	0	0	1	1
x85	0	0	0	0	0
x52	0	0	0	0	0
x82	0	0	0	0	0
x64	0	0	0	0	0
x62	0	0	0	0	0
x84	0	0	0	0	0
x95	0	1	0	0	0
x94	0	0	0	0	0
x42	3	4	3	2	2

Fig. 3 - Output com os resultados da função objetivo.

LPSolve IDE

```
SUBMITTED
Model size:       11 constraints,      22 variables,           62 non-zeros.
Sets:                                   0 GUB,                  0 SOS.

Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2.
The primal and dual simplex pricing strategy set to 'Devex'.

Found feasibility by dual simplex after           10 iter.

Relaxed solution                  49 after         10 iter is B&B base.

Feasible solution                 55 after         21 iter,         5 nodes (gap 12.0%)
Improved solution                 52 after         30 iter,        10 nodes (gap 6.0%)
Improved solution                 51 after         48 iter,        24 nodes (gap 4.0%)
Improved solution                 50 after        281 iter,       171 nodes (gap 2.0%)

Optimal solution                  50 after        756 iter,       490 nodes (gap 2.0%).

Relative numeric accuracy ||*|| = 4.44089e-016

 MEMO: lp_solve version 5.5.2.11 for 32 bit OS, with 64 bit REAL variables.
      In the total iteration count 756, 2 (0.3%) were bound flips.
      There were 247 refactorizations, 0 triggered by time and 0 by density.
       ... on average 3.1 major pivots per refactorization.
      The largest [LUSOL v2.2.1.0] fact(B) had 39 NZ entries, 1.2x largest basis.
      The maximum B&B level was 17, 0.4x MIP order, 7 at the optimal solution.
      The constraint matrix inf-norm is 2, with a dynamic range of 2.
      Time to load data was 0.006 seconds, presolve used 0.006 seconds,
       ... 0.037 seconds in simplex solver, in total 0.049 seconds.
```

Fig. 4 - Output do terminal do LPSolve.

3.1 - Explicação da Solução Ótima

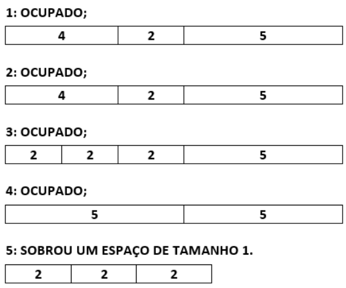

Fig. 5 - Empacotamentos resultantes da função obejtivo.

- No caso **1, 2**, o tipo de empacotamento é o mesmo, foi usado inicialmente o corte de **x114**, que ocupou logo com os 2 únicos itens de tamanho **4**, as partes que correspondiam a esse tamanho. De seguida, seguiram-se 2 cortes do tipo **x75**, ocupando as 2 partições resultantes com os itens de tamanho **2 e 5**, resultando na ocupação completa de um contentor de tamanho **11**.

- No caso **3**, foi usado o corte do tipo **x114**, mas como se tinha esgotado os itens de tamanho, teve de ser efetuado um corte na partição de tamanho **4**, do tipo **x42**, para se inserir 2 itens de tamanho **2**, no espaço de tamanho **4** disponível. De seguida, foi efetuado um corte do tipo **x75**, ocupando as 2 partições resultantes com os itens de tamanho **2 e 5**, resultando na ocupação completa de mais um contentor de tamanho **11**.

- No caso **4**, realizou-se um corte do tipo **x105**, que originou 2 partições de tamanho **5**, que foram ocupadas totalmente por 2 itens de tamanho **5**, resultando na ocupação completa de um contentor de tamanho **10** e na esgotação dos itens de tamanho **5**.

- No caso **5**, realizou-se um corte do tipo **x74**, criando 2 partições de tamanho **3 e 4**. Na partição de tamanho **4**, efetuou-se um corte do tipo **x42**, que gerou 2 partições de tamanho **2**, que foram ocupadas por 2 itens de tamanho **2**. Sobrou assim uma partição de tamanho **3**, que sofreu um corte do tipo **x32**, que foi ocupado por um item de tamanho **2**, mas resultou na sobra de uma partição de tamanho **1**, que levou ao esgotamento de todos os itens, mas não resultou na ocupação completa do contentor de tamanho **7**.

FUNÇÃO OBJETIVO:

Z = (11 * 3) + (10 * 1) + (7 * 3) + (7 * 1) - (7 * 3) = **50**

4 - Validação do Modelo

CORTES	X114	X105	X75	X74	X42	X32	TOTAIS
ITENS DE TAMANHO 2	2	0	3	0	2	1	8
ITENS DE TAMANHO 4	2	0	0	0	0	0	2
ITENS DE TAMANHO 5	0	2	3	0	0	0	5
CONTENTORES DE TAMANHO 11	3	0	0	0	0	0	3
CONTENTORES DE TAMANHO 10	0	1	0	0	0	0	1
CONTENTORES DE TAMANHO 7	0	0	0	1	0	0	1

Fig. 6 - Tabela de valores de nº de itens, contentores e cortes usados.

Com ajuda da tabela e com base nas restrições criadas no LPSolve, vamos seguir então com a validação do projeto:

- **Quantidades de itens usadas:** inicialmente tinhamos 8 itens de tamanho **2**, 2 itens de tamanho **4** e 5 itens de tamanho **5**. Com base, nos dados da tabela e das restrições conseguimos empacotar todos os itens iniciais.

  DEMONSTRAÇÃO DA RESTRIÇÃO:
  Item de tamanho 2: 3 (x75) + 2 * 2 (x42) + 1 (x32) = 8 **(ESGOTOU)**;
  Item de tamanho 4: 3 (x114) + 1 (x74) - 2 (x42) = 2 **(ESGOTOU)**;
  Item de tamanho 5: 2 * 1 (x105) + 3 (x75) = 5 **(ESGOTOU)**.

- **Quantidades de contentores usados:** inicialmente tinhamos contentores ilimitados de tamanho 11, 1 contentor de tamanho 10 e 1 contentor de tamanho 7. Com base, nos dados da tabela e das restrições conseguimos respeitar todas as limitações dos contentores disponíveis.

  DEMONSTRAÇÃO:
  Contentor de tamanho 11: 3 (x114) = **ILIMITADO**;
  Contentor de tamanho 10: 1 (x105) = 1 **(ESGOTOU)**;
  Contentor de tamanho 7: 3 (x75) + 1 (x74) - 3 (x114) = 1 **(ESGOTOU)**;

- **Espaço usado:** este espaço consiste no respeito da limitação que o comprimento de um contentor traz para o problema. Com base, nos dados da tabela e das restrições conseguimos respeitar todas as limitações dos tamanhos de cada contentor.

  DEMONSTRAÇÃO:
  Comprimento 11 (x114, x75): (1 x 2) + (1 x 4) + (1 x 5) = 11 **(ESGOTOU)**;
  Comprimento 11 (x114, x75): (1 x 2) + (1 x 4) + (1 x 5) = 11 **(ESGOTOU)**;
  Comprimento. 11 (x114, x42, x75): (3 x 2) + (1 x 5) = 11 **(ESGOTOU)**;
  Comprimento 10 (x105): (2 x 5) = 10 **(ESGOTOU)**;
  Comprimento 7 (x74, x42, x32): (3 x 2) = 6 **(DEIXOU UM ESPAÇO DE TAMANHO 1 DISPONÍVEL)**;