



UNIVERSIDADE DO MINHO  
LICENCIATURA EM ENGENHARIA INFORMÁTICA

## Investigação Operacional

André Santos - a106854      Daniel Parente - a107363  
Pedro Ferreira - a107292

Ano Letivo 2024/2025

## 0. Valores a trabalhar

De acordo com a informação do enunciado, o valor de  $xABCDE$  a trabalhar neste projeto será 107363. Daqui, podemos tirar que  $x = 1$ ,  $A = 0$ ,  $B = 7$ ,  $C = 3$ ,  $D = 6$  e  $E = 3$ . Seguindo as regras definidas no enunciado, foi necessário calcular o número de contentores e de itens para utilizar. Esses resultados podem ser observados nos pontos a seguir:

### Conteúdo dos contentores

- Comprimento 11: disponibilidade ilimitada;
- Comprimento 10: quantidade disponível =  $B + 1 = 7 + 1 = 8$ ;
- Comprimento 7: quantidade disponível =  $D + 1 = 6 + 1 = 7$ .

A qual podemos sumarizar na Tabela 1:

| Comprimento | Quantidade disponível |
|-------------|-----------------------|
| 11          | ilimitada             |
| 10          | 8                     |
| 7           | 7                     |

Tabela 1: Lista de contentores.

### Itens a empacotar

- Para itens de comprimento 1, como  $B$  é ímpar, temos uma quantidade disponível de  $k_1 = 2$  itens;
- Para itens de comprimento 2, como  $C$  é ímpar, temos uma quantidade disponível de  $k_2 = 11$  itens;
- Para itens de comprimento 3, como  $D$  é par, temos uma quantidade disponível de  $k_3 = 0$  itens;
- Para itens de comprimento 4, como  $E$  é ímpar, temos uma quantidade disponível de  $k_4 = 11$  itens;
- Existem sempre 5 itens de comprimento 5.

Esta informação encontra-se sumarizada na Tabela 2:

| Comprimento | Quantidade |
|-------------|------------|
| 1           | 2          |
| 2           | 11         |
| 3           | 0          |
| 4           | 11         |
| 5           | 5          |

Tabela 2: Lista de itens a empacotar.

## Soma dos comprimentos dos itens

A soma do comprimento total de todos os itens é calculada como:

$$\underbrace{1 \times 2}_{\text{(item 1)}} + \underbrace{2 \times 11}_{\text{(item 2)}} + \underbrace{3 \times 0}_{\text{(item 3)}} + \underbrace{4 \times 11}_{\text{(item 4)}} + \underbrace{5 \times 5}_{\text{(item 5)}} = 2 + 22 + 0 + 44 + 25 = 93.$$

Portanto, o total de comprimento de itens a empacotar é **93**.

## 1. Formulação do Problema

O problema abordado é um **empacotamento unidimensional** (*bin packing problem*), consiste em empacotar um conjunto de itens de diferentes comprimentos em contentores de diferentes capacidades. Cada contentor possui um comprimento máximo que não pode ser excedido pela soma dos comprimentos dos itens nele contidos. O objetivo é minimizar a soma dos comprimentos dos contentores utilizados, ou seja, utilizar os contentores de forma eficiente.

### 1.1 Descrição do Problema

Temos:

- Uma lista de itens, onde cada item pertence a uma *classe* específica: **todos os itens da mesma classe têm o mesmo comprimento**;
- Um conjunto de classes de contentores (por exemplo, comprimentos 11, 10 e 7), sendo que algumas classes têm uma *disponibilidade limitada*, enquanto outras são *ilimitadas*;
- Cada contentor **não pode** exceder a sua capacidade ao acomodar os itens;
- **Todos** os itens têm de ser empacotados, e nenhum pode ser alocado para mais do que um contentor.

A *medida de eficiência* adotada é a **soma dos comprimentos dos contentores** efetivamente usados (que têm itens no interior). Assim, se um contentor de comprimento  $W_c$  for utilizado, contribui em  $W_c$  para o custo total.

### 1.2 Objetivo

O objetivo é **minimizar** a soma dos comprimentos dos contentores, ou seja, usar a menor quantidade possível de contentores, não ultrapassando nunca a capacidade de nenhum deles.

### 1.3 Variáveis de Decisão

Para uma *formulação em fluxos em arcos* (*arc-flow*), escolhemos as seguintes variáveis:

- **Variáveis de fluxo**  $x_{c,i,j}$ : representando a quantidade de itens que percorrem o arco  $i \rightarrow j$  no grafo associado à classe de contentor  $c$  (*arc-flow*);

- **Variáveis *bin*  $y_c$** : representando o número de contentores da classe  $c$  efetivamente utilizados.

Essas variáveis codificam:

“Se  $x_{c,i,j} > 0$ , significa que um certo número de itens (da classe associada ao arco  $a$ ) está a transitar naquele “espaço” no contentor  $c$ . Já  $y_c$  indica quantos contentores do tipo  $c$  foram abertos/estão a ser usados.”

## 1.4 Coerência Global do Modelo

- A **conservação de fluxo** em cada nó do grafo garante que, ao “ocupar” uma parte da capacidade do contentor, deve-se seguir consistentemente até ao preenchimento completo;
- As **restrições de disponibilidade de contentores** garantem que não usamos mais contentores do que o permitido para cada classe;
- A **cobertura de itens** impõe que todos os itens sejam empacotados, sem faltar nem sobrar;
- A **função objetivo linear** minimiza a soma  $W_c \times y_c$  em cada classe de contentor, refletindo a soma total de comprimentos dos contentores usados.

## 1.5 Justificativa Teórica

A formulação *arc-flow* é um modelo amplamente estudado para problemas de corte e empacotamento, nomeadamente nos trabalhos do docente da cadeira, Valério de Carvalho (1999, 2002). Trata-se de construir um grafo de estados para cada tipo de bin, com nós representando estados de ocupação e arcos representando a inserção de um item. O uso de um modelo arc-flow abrange todas as possíveis combinações de itens que podem caber num contentor, sem explicitamente enumerar cada um de forma individual.

# 2. Modelo de Programação Linear

## 2.1 Parâmetros e Conjuntos

- Conjunto de classes de **itens**  $I = \{1, \dots, n\}$ . Para cada  $i \in I$ :
  - $q_i$ : quantidade de itens da classe  $i$  (número inteiro não negativo).
  - $l_i$ : comprimento do item da classe  $i$  (valor inteiro positivo até  $W_c$ , conforme o problema).
- Conjunto de classes de **contentores**  $C = \{1, \dots, m\}$ . Para cada  $c \in C$ :
  - $W_c$ : capacidade (comprimento máximo) do contentor classe  $c$  (valor inteiro positivo).
  - $M_c$ : quantidade máxima (inteira) de contentores da classe  $c$  disponível (pode ser  $\infty$  ou suficientemente grande, se for ilimitado).
- Para cada  $c \in C$ , definimos um *grafo*  $G_c = (V_c, A_c)$  de estados de ocupação:

- $V_c = \{0, 1, \dots, W_c\}$  são nós que representam o “estado” de ocupação (de *vazio* (0) até *cheio* ( $W_c$ )).
- $A_c$  é o conjunto de arcos  $(u \rightarrow v)$  sempre que  $v - u = l_i$  para algum item  $i \in I$  e  $0 \leq u < v \leq W_c$ . Assim, o arco  $(u \rightarrow u + l_i)$  significa “colocar um item da classe  $i$ ” e ocupar mais  $l_i$  unidades de comprimento.

## 2.2 Variáveis de Decisão

**Variáveis de fluxo**  $x_{c,i,j}$

$$x_{c,i,j} \in \mathbb{Z}_{\geq 0} \quad \forall c \in C, \forall (i,j) \in A_c.$$

Significado:

- $x_{c,i,j}$  indica *quantos itens* fluem pelo arco  $a$  no grafo do tipo de contentor  $c$ .
- Cada arco  $a = (u \rightarrow u + l_i)$  corresponde a inserir um item de comprimento  $l_i$ .
- Se  $x_{c,i,j} > 0$ , quer dizer que  $x_{c,i,j}$  itens daquele tipo foram “colocados” nessa “transição” de espaço ocupado.

**Variáveis de contentores**  $y_c$

$$y_c \in \mathbb{Z}_{\geq 0} \quad \forall c \in C.$$

Significado:

- $y_c$  representa o *número total de contentores do tipo  $c$*  efetivamente usados na solução.
- Se  $y_c = 0$ , não se usa nenhum bin do tipo  $c$ ; se  $y_c = 3$ , então usamos 3 contentores desse tipo, etc.

## 2.3 Função Objectivo (Linear)

$$\min \sum_{c \in C} (W_c \cdot y_c).$$

Explicação:

- Cada bin do tipo  $c$ , se usado, tem um “custo” igual à sua capacidade  $W_c$ .
- Portanto, usar  $y_c$  contentores do tipo  $c$  acrescenta  $W_c \times y_c$  ao total de “comprimento de contentores” usado.
- O objetivo é *minimizar* a soma desses comprimentos para todos os tipos de contentores.

## 2.4 Restrições Lineares

(1) **Conservação de fluxo em nós internos.** Para cada  $c \in C$  e cada nó  $u \in V_c$  tal que  $0 < u < W_c$ :

$$\sum_{a=(v \rightarrow u) \in A_c} x_{c,a} = \sum_{a=(u \rightarrow w) \in A_c} x_{c,a}.$$

**Interpretação:**

- O fluxo total que “entra” no estado  $u$  tem de ser igual ao que “sai” de  $u$ ;
- Assegura que não há “perda” nem “acumulação” de itens a meio do grafo: qualquer item que chegou ao estado  $u$  *prosegue* até outro estado  $w$ .

(2) **Fluxo em 0 e em  $W_c$ .** Para cada  $c \in C$ :

$$\sum_{a=(0 \rightarrow w) \in A_c} x_{c,a} = y_c \quad \text{e} \quad \sum_{a=(v \rightarrow W_c) \in A_c} x_{c,a} = y_c.$$

**Interpretação:**

- No nó inicial 0, iniciamos  $y_c$  unidades de fluxo, pois queremos abrir  $y_c$  contentores do tipo  $c$ ;
- No nó final  $W_c$ , chegam exatamente  $y_c$  unidades de fluxo, correspondendo à ocupação de todos esses contentores (cada bin sai “completo” do grafo).

(3) **Limite de contentores.** Para cada  $c \in C$ :

$$y_c \leq M_c,$$

caso  $M_c$  seja finito. Se o tipo  $c$  for ilimitado, esta restrição pode ser omitida ou escrita como  $y_c \leq (\text{valor grande})$ .

(4) **Satisfação de todas as quantidades de itens.** Para cada tipo de item  $i \in I$ :

$$\sum_{c \in C} \sum_{(u \rightarrow u+l_i) \in A_c} x_{c,(u \rightarrow u+l_i)} = q_i.$$

**Interpretação:**

- Cada arco  $(u \rightarrow u + l_i)$  representa inserir um item de tipo  $i$  (comprimento  $l_i$ ) num contentor do tipo  $c$ ;
- Precisamos embalar *exatamente*  $q_i$  itens do tipo  $i$ , somando em todos os contentores de todos os tipos.

(5) **Inteiridade e não-negatividade.**

$$x_{c,a} \in \mathbb{Z}_{\geq 0}, \quad y_c \in \mathbb{Z}_{\geq 0}.$$

**Interpretação:**

- **Inteiridade:** As variáveis devem ser números inteiros não-negativos. Isso é necessário, pois estamos a lidar com quantidades discretas (itens e contentores);
- **Não-negatividade:** Nenhuma variável pode ter valores negativos, mantendo consistência com o contexto do problema (não podemos usar um número negativo de contentores, nem alocar uma quantidade negativa de itens).

## 2.5 Coerência Global do Modelo

- **Linearidade:** Todas as equações acima são lineares em termos das variáveis  $x_{c,a}$  e  $y_c$ . A função objetivo também é linear,  $\sum_c W_c y_c$ . Isto garante-nos que o modelo é solúvel de forma eficiente.
- **Regras de funcionamento do sistema:**
  - As restrições de fluxo garantem que os contentores não ultrapassem a capacidade (pois não existem arcos que excedam  $W_c$ ) e que cada bin, se aberto, corresponde a um caminho completo no grafo.
  - A limitação  $y_c \leq M_c$  assegura que não usamos mais contentores do que o fornecido.
  - A restrição de cobertura de itens garante que todos os itens sejam *colocados* (empacotados), através da soma dos fluxos dos arcos correspondentes aos itens de cada tipo.
- **Função objetivo como medida de eficiência:** A função objetivo visa minimizar o *comprimento total* dos contentores usados. Ou seja, minimiza a quantidade total de espaço usado, o que reflete a medida de eficiência especificada anteriormente.

## 3 Ficheiro de *input*

```
min: 11*y_11 + 10*y_10 + 7*y_7;

r11_0: x11_0_1 + x11_0_2 + x11_0_3 + x11_0_4 + x11_0_5 = y_11;
r11_1: x11_0_1 = x11_1_2 + x11_1_3 + x11_1_4 + x11_1_5 + x11_1_6;
r11_2: x11_0_2 + x11_1_2 = x11_2_3 + x11_2_4 + x11_2_5 + x11_2_6 +
      x11_2_7;
r11_3: x11_0_3 + x11_1_3 + x11_2_3 = x11_3_4 + x11_3_5 + x11_3_6 +
      x11_3_7 + x11_3_8;
r11_4: x11_0_4 + x11_1_4 + x11_2_4 + x11_3_4 = x11_4_5 + x11_4_6 +
      x11_4_7 + x11_4_8 + x11_4_9;
r11_5: x11_0_5 + x11_1_5 + x11_2_5 + x11_3_5 + x11_4_5 = x11_5_6 +
      x11_5_7 + x11_5_8 + x11_5_9 + x11_5_10;
r11_6: x11_1_6 + x11_2_6 + x11_3_6 + x11_4_6 + x11_5_6 = x11_6_7 +
      x11_6_8 + x11_6_9 + x11_6_10 + x11_6_11;
r11_7: x11_2_7 + x11_3_7 + x11_4_7 + x11_5_7 + x11_6_7 = x11_7_8 +
      x11_7_9 + x11_7_10 + x11_7_11;
r11_8: x11_3_8 + x11_4_8 + x11_5_8 + x11_6_8 + x11_7_8 = x11_8_9 +
      x11_8_10 + x11_8_11;
r11_9: x11_4_9 + x11_5_9 + x11_6_9 + x11_7_9 + x11_8_9 = x11_9_10 +
      x11_9_11;
r11_10: x11_5_10 + x11_6_10 + x11_7_10 + x11_8_10 + x11_9_10 = x11_10_11
;
r11_11: x11_6_11 + x11_7_11 + x11_8_11 + x11_9_11 + x11_10_11 = y_11;

r10_0: x10_0_1 + x10_0_2 + x10_0_3 + x10_0_4 + x10_0_5 = y_10;
r10_1: x10_0_1 = x10_1_2 + x10_1_3 + x10_1_4 + x10_1_5 + x10_1_6;
r10_2: x10_0_2 + x10_1_2 = x10_2_3 + x10_2_4 + x10_2_5 + x10_2_6 +
      x10_2_7;
r10_3: x10_0_3 + x10_1_3 + x10_2_3 = x10_3_4 + x10_3_5 + x10_3_6 +
      x10_3_7 + x10_3_8;
```

```

r10_4: x10_0_4 + x10_1_4 + x10_2_4 + x10_3_4 = x10_4_5 + x10_4_6 +
x10_4_7 + x10_4_8 + x10_4_9;
r10_5: x10_0_5 + x10_1_5 + x10_2_5 + x10_3_5 + x10_4_5 = x10_5_6 +
x10_5_7 + x10_5_8 + x10_5_9 + x10_5_10;
r10_6: x10_1_6 + x10_2_6 + x10_3_6 + x10_4_6 + x10_5_6 = x10_6_7 +
x10_6_8 + x10_6_9 + x10_6_10;
r10_7: x10_2_7 + x10_3_7 + x10_4_7 + x10_5_7 + x10_6_7 = x10_7_8 +
x10_7_9 + x10_7_10;
r10_8: x10_3_8 + x10_4_8 + x10_5_8 + x10_6_8 + x10_7_8 = x10_8_9 +
x10_8_10;
r10_9: x10_4_9 + x10_5_9 + x10_6_9 + x10_7_9 + x10_8_9 = x10_9_10;
r10_10: x10_5_10 + x10_6_10 + x10_7_10 + x10_8_10 + x10_9_10 = y_10;

r7_0: x7_0_1 + x7_0_2 + x7_0_3 + x7_0_4 + x7_0_5 = y_7;
r7_1: x7_0_1 = x7_1_2 + x7_1_3 + x7_1_4 + x7_1_5 + x7_1_6;
r7_2: x7_0_2 + x7_1_2 = x7_2_3 + x7_2_4 + x7_2_5 + x7_2_6 + x7_2_7;
r7_3: x7_0_3 + x7_1_3 + x7_2_3 = x7_3_4 + x7_3_5 + x7_3_6 + x7_3_7;
r7_4: x7_0_4 + x7_1_4 + x7_2_4 + x7_3_4 = x7_4_5 + x7_4_6 + x7_4_7;
r7_5: x7_0_5 + x7_1_5 + x7_2_5 + x7_3_5 + x7_4_5 = x7_5_6 + x7_5_7;
r7_6: x7_1_6 + x7_2_6 + x7_3_6 + x7_4_6 + x7_5_6 = x7_6_7;
r7_7: x7_2_7 + x7_3_7 + x7_4_7 + x7_5_7 + x7_6_7 = y_7;

r_item1:
x11_0_1 + x11_1_2 + x11_2_3 + x11_3_4 + x11_4_5 + x11_5_6 + x11_6_7 +
x11_7_8 + x11_8_9 + x11_9_10 + x11_10_11 +
x10_0_1 + x10_1_2 + x10_2_3 + x10_3_4 + x10_4_5 + x10_5_6 + x10_6_7 +
x10_7_8 + x10_8_9 + x10_9_10 +
x7_0_1 + x7_1_2 + x7_2_3 + x7_3_4 + x7_4_5 + x7_5_6 + x7_6_7
= 2;

r_item2:
x11_0_2 + x11_1_3 + x11_2_4 + x11_3_5 + x11_4_6 + x11_5_7 + x11_6_8 +
x11_7_9 + x11_8_10 + x11_9_11 +
x10_0_2 + x10_1_3 + x10_2_4 + x10_3_5 + x10_4_6 + x10_5_7 + x10_6_8 +
x10_7_9 + x10_8_10 +
x7_0_2 + x7_1_3 + x7_2_4 + x7_3_5 + x7_4_6 + x7_5_7
= 11;

r_item3:
x11_0_3 + x11_1_4 + x11_2_5 + x11_3_6 + x11_4_7 + x11_5_8 + x11_6_9 +
x11_7_10 + x11_8_11 +
x10_0_3 + x10_1_4 + x10_2_5 + x10_3_6 + x10_4_7 + x10_5_8 + x10_6_9 +
x10_7_10 +
x7_0_3 + x7_1_4 + x7_2_5 + x7_3_6 + x7_4_7
= 0;

r_item4:
x11_0_4 + x11_1_5 + x11_2_6 + x11_3_7 + x11_4_8 + x11_5_9 + x11_6_10 +
x11_7_11 +
x10_0_4 + x10_1_5 + x10_2_6 + x10_3_7 + x10_4_8 + x10_5_9 + x10_6_10 +
x7_0_4 + x7_1_5 + x7_2_6 + x7_3_7
= 11;

r_item5:
x11_0_5 + x11_1_6 + x11_2_7 + x11_3_8 + x11_4_9 + x11_5_10 + x11_6_11
+
x10_0_5 + x10_1_6 + x10_2_7 + x10_3_8 + x10_4_9 + x10_5_10 +
x7_0_5 + x7_1_6 + x7_2_7

```



```

= 5;

r_lim10: y_10 <= 8;
r_lim7: y_7 <= 7;

int y_11, y_10, y_7;

int
  x11_0_1, x11_0_2, x11_0_3, x11_0_4, x11_0_5,
  x11_1_2, x11_1_3, x11_1_4, x11_1_5, x11_1_6,
  x11_2_3, x11_2_4, x11_2_5, x11_2_6, x11_2_7,
  x11_3_4, x11_3_5, x11_3_6, x11_3_7, x11_3_8,
  x11_4_5, x11_4_6, x11_4_7, x11_4_8, x11_4_9,
  x11_5_6, x11_5_7, x11_5_8, x11_5_9, x11_5_10,
  x11_6_7, x11_6_8, x11_6_9, x11_6_10, x11_6_11,
  x11_7_8, x11_7_9, x11_7_10, x11_7_11,
  x11_8_9, x11_8_10, x11_8_11,
  x11_9_10, x11_9_11,
  x11_10_11;

int
  x10_0_1, x10_0_2, x10_0_3, x10_0_4, x10_0_5,
  x10_1_2, x10_1_3, x10_1_4, x10_1_5, x10_1_6,
  x10_2_3, x10_2_4, x10_2_5, x10_2_6, x10_2_7,
  x10_3_4, x10_3_5, x10_3_6, x10_3_7, x10_3_8,
  x10_4_5, x10_4_6, x10_4_7, x10_4_8, x10_4_9,
  x10_5_6, x10_5_7, x10_5_8, x10_5_9, x10_5_10,
  x10_6_7, x10_6_8, x10_6_9, x10_6_10,
  x10_7_8, x10_7_9, x10_7_10,
  x10_8_9, x10_8_10,
  x10_9_10;

int
  x7_0_1, x7_0_2, x7_0_3, x7_0_4, x7_0_5,
  x7_1_2, x7_1_3, x7_1_4, x7_1_5, x7_1_6,
  x7_2_3, x7_2_4, x7_2_5, x7_2_6, x7_2_7,
  x7_3_4, x7_3_5, x7_3_6, x7_3_7,
  x7_4_5, x7_4_6, x7_4_7,
  x7_5_6, x7_5_7,
  x7_6_7;

```

## 4 Ficheiro de *output*

Value of objective function: 93.00000000

Actual values of the variables:

|          |   |
|----------|---|
| y_11     | 6 |
| y_10     | 2 |
| y_7      | 1 |
| x11_0_1  | 0 |
| x11_0_2  | 6 |
| x_11_0_3 | 0 |
| x11_0_4  | 0 |
| x11_0_5  | 0 |
| x11_1_2  | 0 |
| x11_1_3  | 0 |

|           |   |
|-----------|---|
| x11_1_4   | 0 |
| x11_1_5   | 0 |
| x11_1_6   | 0 |
| x11_2_3   | 2 |
| x11_2_4   | 1 |
| x11_2_5   | 0 |
| x11_2_6   | 0 |
| x11_2_7   | 3 |
| x11_0_3   | 0 |
| x11_3_4   | 0 |
| x11_3_5   | 0 |
| x11_3_6   | 0 |
| x11_3_7   | 2 |
| x11_3_8   | 0 |
| x11_4_5   | 0 |
| x11_4_6   | 0 |
| x11_4_7   | 0 |
| x11_4_8   | 0 |
| x11_4_9   | 1 |
| x11_5_6   | 0 |
| x11_5_7   | 0 |
| x11_5_8   | 0 |
| x11_5_9   | 0 |
| x11_5_10  | 0 |
| x11_6_7   | 0 |
| x11_6_8   | 0 |
| x11_6_9   | 0 |
| x11_6_10  | 0 |
| x11_6_11  | 0 |
| x11_7_8   | 0 |
| x11_7_9   | 0 |
| x11_7_10  | 0 |
| x11_7_11  | 5 |
| x11_8_9   | 0 |
| x11_8_10  | 0 |
| x11_8_11  | 0 |
| x11_9_10  | 0 |
| x11_9_11  | 1 |
| x11_10_11 | 0 |
| x10_0_1   | 0 |
| x10_0_2   | 2 |
| x10_0_3   | 0 |
| x10_0_4   | 0 |
| x10_0_5   | 0 |
| x10_1_2   | 0 |
| x10_1_3   | 0 |
| x10_1_4   | 0 |
| x10_1_5   | 0 |
| x10_1_6   | 0 |
| x10_2_3   | 0 |
| x10_2_4   | 0 |
| x10_2_5   | 0 |
| x10_2_6   | 2 |
| x10_2_7   | 0 |
| x10_3_4   | 0 |
| x10_3_5   | 0 |
| x10_3_6   | 0 |
| x10_3_7   | 0 |

|          |   |
|----------|---|
| x10_3_8  | 0 |
| x10_4_5  | 0 |
| x10_4_6  | 0 |
| x10_4_7  | 0 |
| x10_4_8  | 0 |
| x10_4_9  | 0 |
| x10_5_6  | 0 |
| x10_5_7  | 0 |
| x10_5_8  | 0 |
| x10_5_9  | 0 |
| x10_5_10 | 0 |
| x10_6_7  | 0 |
| x10_6_8  | 0 |
| x10_6_9  | 0 |
| x10_6_10 | 2 |
| x10_7_8  | 0 |
| x10_7_9  | 0 |
| x10_7_10 | 0 |
| x10_8_9  | 0 |
| x10_8_10 | 0 |
| x10_9_10 | 0 |
| x7_0_1   | 0 |
| x7_0_2   | 0 |
| x7_0_3   | 0 |
| x7_0_4   | 0 |
| x7_0_5   | 1 |
| x7_1_2   | 0 |
| x7_1_3   | 0 |
| x7_1_4   | 0 |
| x7_1_5   | 0 |
| x7_1_6   | 0 |
| x7_2_3   | 0 |
| x7_2_4   | 0 |
| x7_2_5   | 0 |
| x7_2_6   | 0 |
| x7_2_7   | 0 |
| x7_3_4   | 0 |
| x7_3_5   | 0 |
| x7_3_6   | 0 |
| x7_3_7   | 0 |
| x7_4_5   | 0 |
| x7_4_6   | 0 |
| x7_4_7   | 0 |
| x7_5_6   | 0 |
| x7_5_7   | 1 |
| x7_6_7   | 0 |

## 5 Solução

### Valor da Solução Ótima

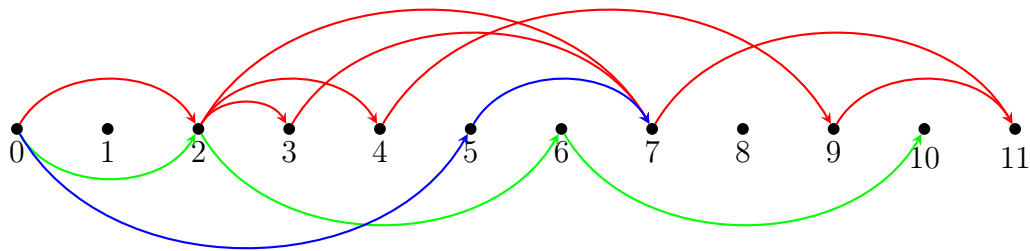
A solução ótima encontrada pelo solver apresenta um valor de função objetivo de:

$$\text{Valor Ótimo} = 93 \quad (1)$$

Isso significa que a soma dos comprimentos dos contentores utilizados na solução ótima é de 93 unidades, o que corresponde ao comprimento total de itens a empacotar, não havendo, por isso, espaços desperdiçados nos bins.

## Grafo Subjacente ao Modelo de Fluxo em Arcos

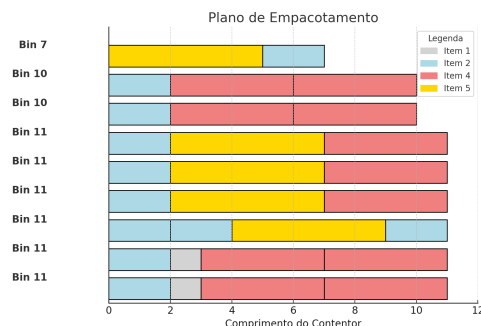
A figura abaixo representa o grafo do modelo de fluxo em arcos, indicando os fluxos não nulos na solução ótima.



Os nós representam os diferentes estados dos itens e contentores no processo de empacotamento, e as setas indicam os caminhos seguidos pelo fluxo de itens. Os arcos a vermelho representam os itens alocados em contentores de tamanho 11, os a verde representam os itens alocados em contentores de tamanho 10, e os azuis os alocados em contentores de tamanho 7.

## Plano de Empacotamento

Abaixo, apresentamos um plano visual dos contentores utilizados e os itens alocados em cada um deles. Cada contentor é representado por uma barra com o seu comprimento total, e os itens são ilustrados dentro do respectivo contentor.



Contentores idênticos foram agregados para evitar redundância na apresentação.

## 6 Validação do Modelo

Para garantir que o modelo e a solução obtida são corretos, foram seguidos os seguintes procedimentos:

- **Verificação das Restrições:** Conferimos se todas as restrições do modelo foram respeitadas na solução fornecida pelo solver.

- **Comparação com Casos Simples:** Foram testados pequenos exemplos cuja solução ótima é conhecida para validar a coerência do modelo.
- **Análise da Estrutura do Fluxo:** O fluxo dos itens foi conferido manualmente no grafo gerado, garantindo que a solução reflete corretamente as regras do problema.

A partir dessas verificações, e tendo também em conta de que a soma dos comprimentos dos contentores usados iguala a soma dos itens a alocar, e que não é possível obter essa soma usando menos contentores, foi possível concluir que a solução fornecida pelo solver é a melhor solução admissível e representa corretamente uma decisão ótima para o problema proposto.