

Prof. Mohamed Aly LOULY

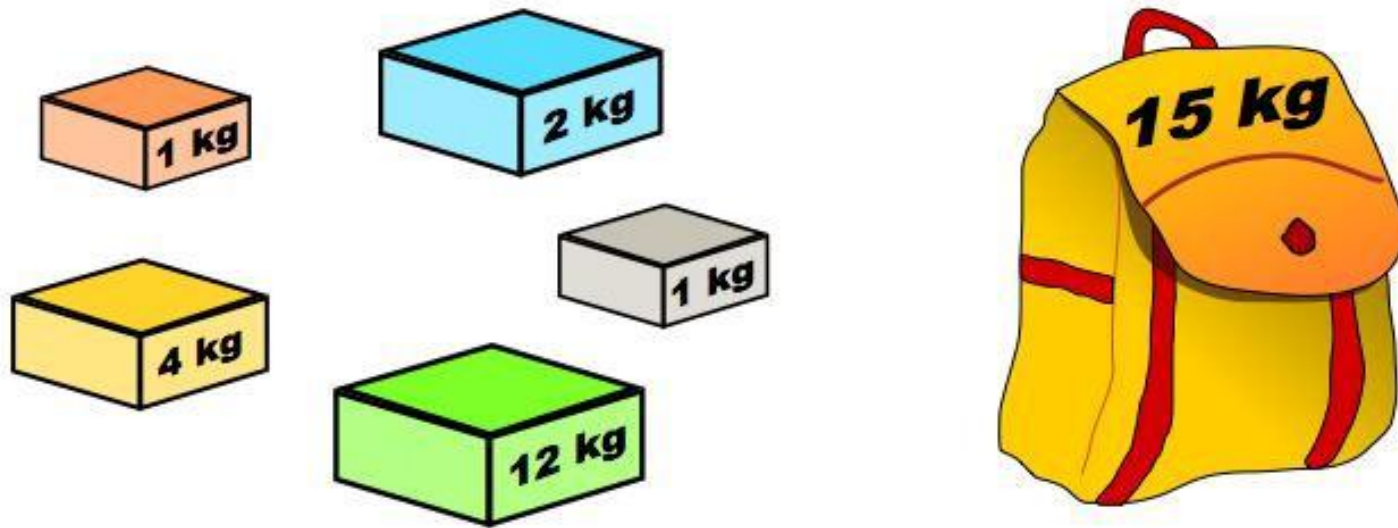
Recherche Opérationnelle

1

- ✓ Modélisation des problèmes de sac-à-dos
- ✓ Modélisation des problèmes de Bin Packing
- ✓ Modélisation des problèmes de découpe

PROBLÈME DU SAC-À-DOS

« KNAPSACK PROBLEM »



Prendre les objets les plus utiles en respectant la capacité.
Rappel : il y a 2^n partitions d'un ensemble de n objets.

PROBLÈME DU SAC-À-DOS BINAIRE

<< 0-1 KNAPSACK PROBLEM >>

Une société de transport cherche à maximiser le profit du chargement d'une camionnette d'une capacité maximale d'une tonne. Le tableau suivant donne la liste des 20 objets à transporter avec leur bénéfice et leur poids en kg.

Objet	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Bénéfice	54	82	69	23	29	26	86	34	18	94	99	11	99	83	99	64	10	49	40	25
Poids	66	80	77	67	83	58	52	69	97	87	90	54	71	74	78	69	89	79	75	91

PROBLÈME DU SAC-À-DOS BINAIRE

« 0-1 KNAPSACK PROBLEM »

b_i : le bénéfice de l'objet i .

p_i : le poids de l'objet i .

x_i : variable binaire égale 1 si l'objet i est chargé, 0 sinon.

$$\text{Maximiser } Z = \sum_{i=1}^n b_i x_i$$

$$\sum_{i=1}^n p_i x_i \leq \text{capacité}.$$

PROBLÈME DU SAC-À-DOS BORNÉ

<< BOUNDED KNAPSACK PROBLEM >>

Une société de transport cherche à maximiser le profit du chargement d'un camion d'une capacité maximale de 10 tonnes. Le tableau suivant donne les types d'objets à transporter, leur profit, leur poids en kg et le nombre d'objets disponibles de chaque type.

Objet	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Bénéfice	54	82	69	23	29	26	86	34	18	94	99	11	99	83	99	64	10	49	40	25
Poids	66	80	77	67	83	58	52	69	97	87	90	54	71	74	78	69	89	79	75	91
Nombre	19	9	16	12	8	18	10	10	12	10	7	13	15	14	11	20	9	13	16	18

PROBLÈME DU SAC-À-DOS BORNÉ << BOUNDED KNAPSACK PROBLEM >>

b_i : le bénéfice de l'objet i .

p_i : le poids de l'objet i .

m_i : le nombre d'objets de type i .

x_i : nombre d'objets de type i chargés.

$$\text{Maximiser } Z = \sum_{i=1}^n b_i x_i$$

$$\sum_{i=1}^n p_i x_i \leq \text{capacité}$$

$$x_i \leq m_i, \quad i = 1, \dots, n.$$

PROBLÈME DU SAC-À-DOS NON BORNÉ « UNBOUNDED KNAPSACK PROBLEM »

b_i : le bénéfice de l'objet i .

p_i : le poids de l'objet i .

x_i : Nombre d'objets de type i chargés.

$$\text{Maximiser } Z = \sum_{i=1}^n b_i x_i$$

$$\sum_{i=1}^n p_i x_i \leq \text{capacité}$$

PROBLÈME DU SAC-À-DOS BORNÉ AVEC PLUSIEURS CONTENEURS

Une société de fret aérien cherche à maximiser le profit du chargement d'une flotte de 10 avions d'une capacité maximale de 100 tonnes chacun. Le tableau suivant donne les types d'objets à transporter, leur poids en tonnes, leur profit en millions d'Ouguiyas et le nombre d'objets disponibles de chaque type.

Objet	1	2	3	4	5	6	7	8	9	10
Bénéfice	60	55	42	36	98	70	82	29	80	78
Poids	5	4	3	2	9	6	8	1	4	7
Nombre	15	52	39	18	45	27	12	12	14	31

PROBLÈME DU SAC-À-DOS BORNÉ AVEC PLUSIEURS CONTENEURS

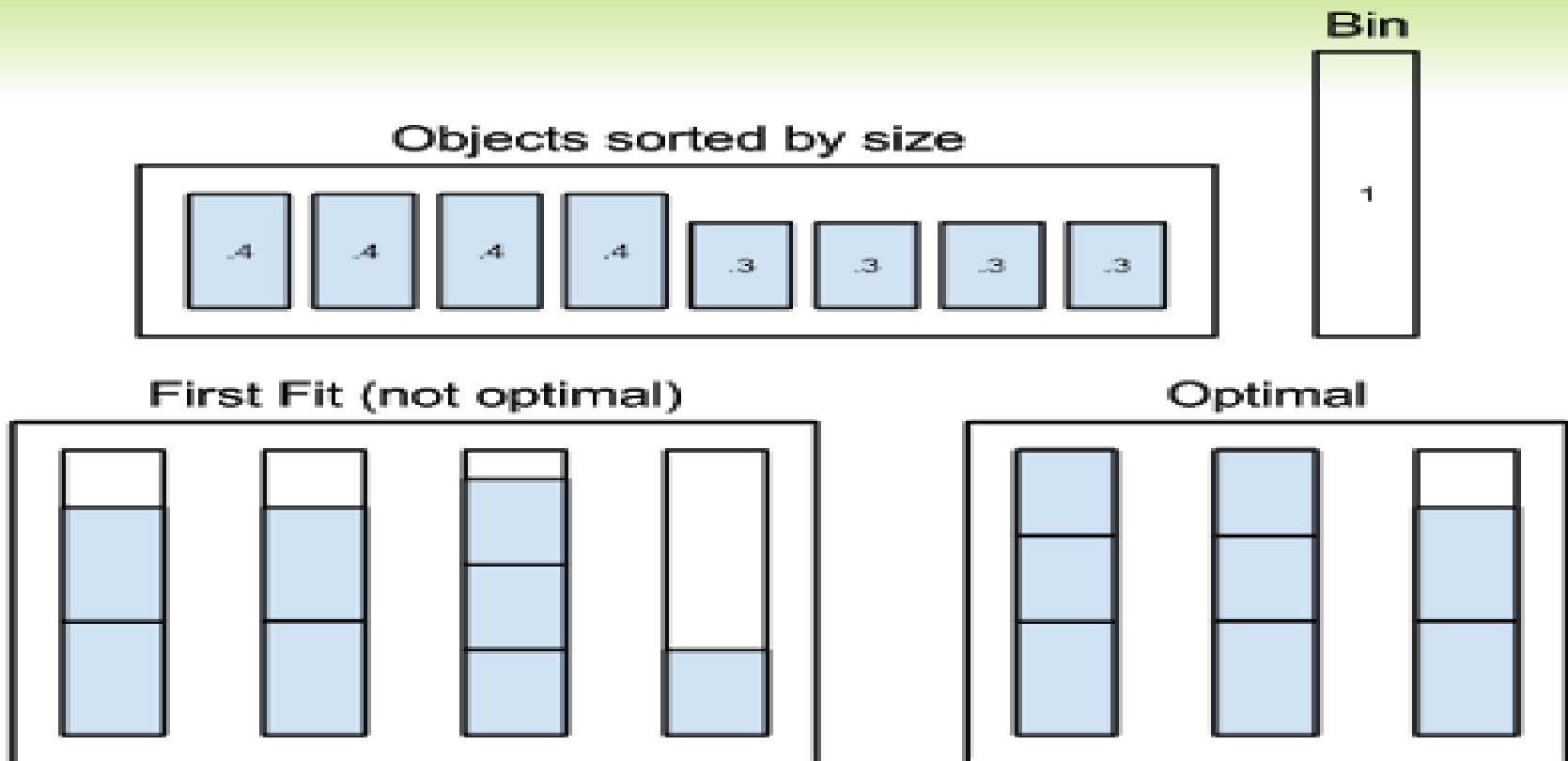
x_{ij} : nombre d'objets de type i chargés dans le conteneur j .

$$\text{Maximiser } Z = \sum_{i=1}^n \sum_{j=1}^m b_i x_{ij}$$

$$\sum_{i=1}^n p_i x_{ij} \leq \text{capacité}_j, \quad j = 1, \dots, m.$$

$$\sum_{j=1}^m x_{ij} \leq m_i, \quad i = 1, \dots, n.$$

LE BIN PACKING



Ranger n objets en utilisant le nombre minimum de conteneurs

MODÉLISATION DU PROBLÈME DU BIN PACKING BINAIRE

x_{ij} : variable binaire égale 1 si l'objet i est placé dans le conteneur j , 0 sinon.

y_j : variable binaire égale 1 si le conteneur j est utilisé, 0 sinon.

$$\text{Minimiser } Z = \sum_{j=1}^n y_j$$

$$\sum_{i=1}^n p_i x_{ij} \leq \text{capacité } y_j, \quad j = 1, \dots, n.$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n.$$

BIN PACKING AVEC PLUSIEURS OBJETS DE MÊME TYPE

x_{ij} : nombre d'objets de type i placés dans le conteneur j .

y_j : variable binaire égale 1 si le conteneur j est utilisé, 0 sinon.

$$\text{Minimiser } Z = \sum_{j=1}^m y_j$$

$$\sum_{i=1}^n p_i x_{ij} \leq \text{capacité } y_j, \quad j = 1, \dots, m.$$

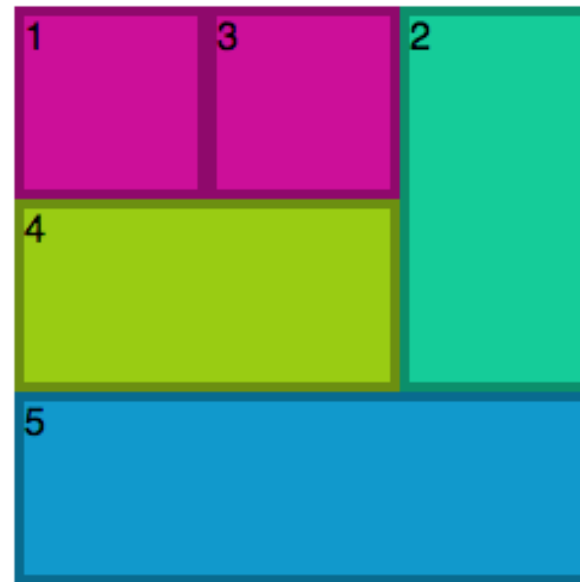
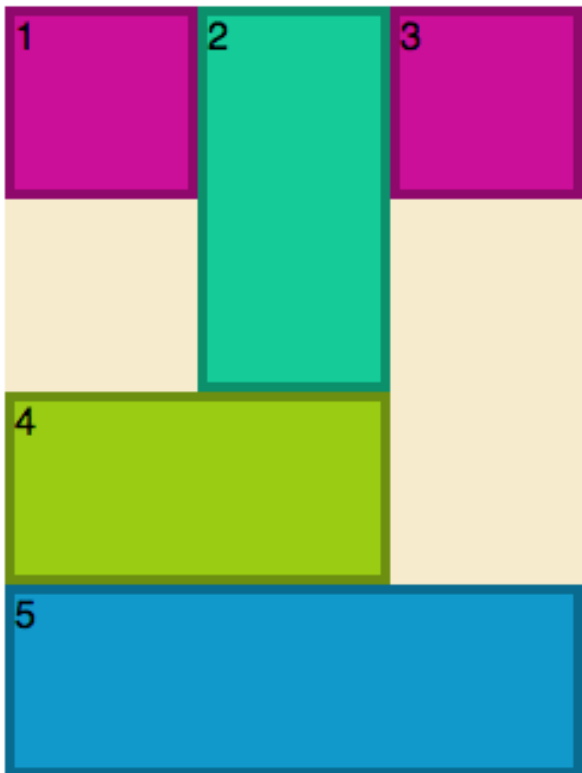
$$\sum_{j=1}^m x_{ij} = m_i, \quad i = 1, \dots, n.$$

BIN PACKING AVEC PLUSIEURS OBJETS DE MÊME TYPE

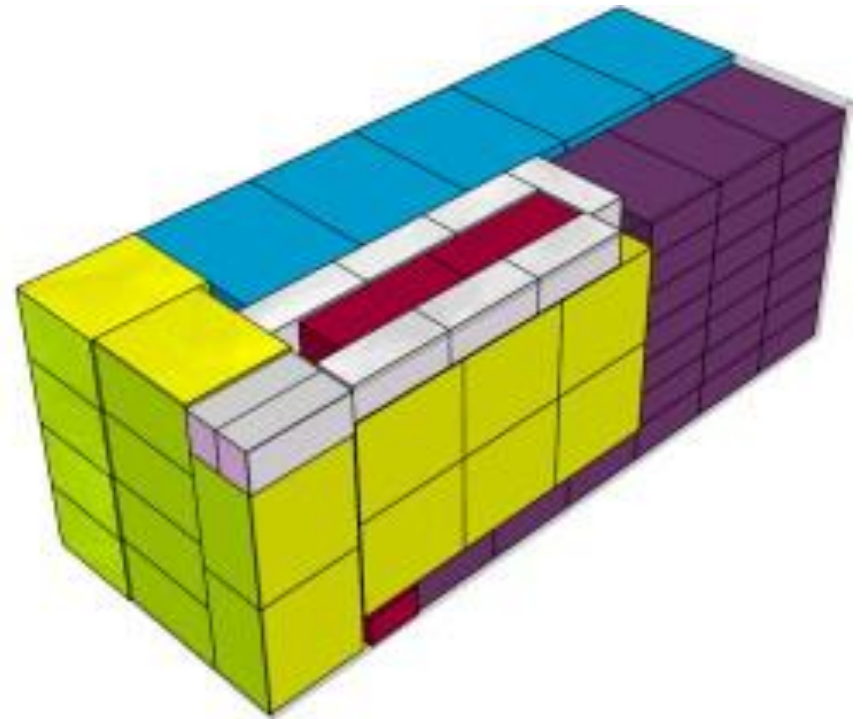
La société de fret aérien cherche maintenant à transporter tous les objets disponibles avec le nombre minimum de vols d'avions d'une capacité maximale de 100 tonnes chacun. Le tableau suivant rappelle les types d'objets à transporter, leur poids en tonnes et le nombre d'objets disponibles de chaque type.

Objet	1	2	3	4	5	6	7	8	9	10
Poids	5	4	3	2	9	6	8	1	4	7
Nombre	15	52	39	18	45	27	12	12	14	31

LE BIN PACKING EN 2D

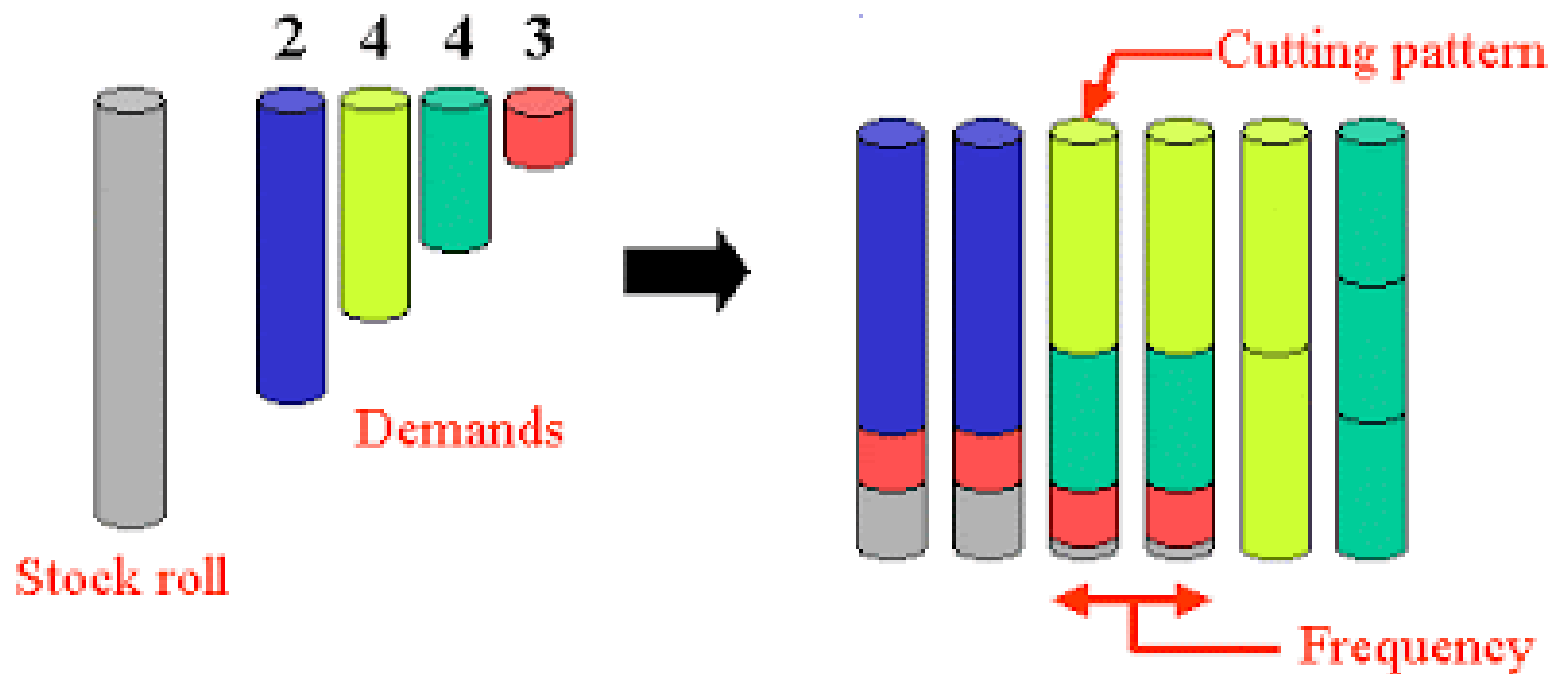


LE BIN PACKING EN 3D



PROBLÈME DE DÉCOUPE

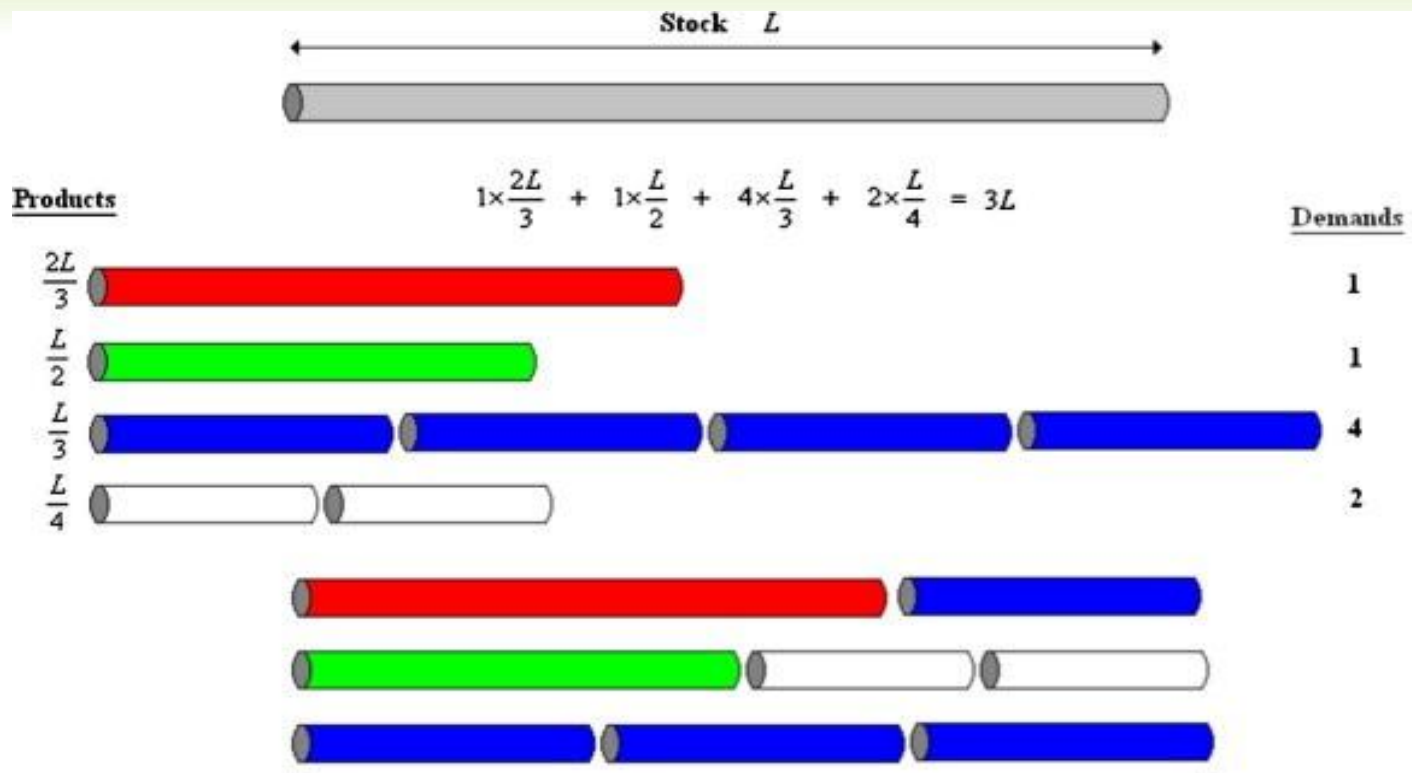
« CUTTING STOCK PROBLEM »



Fabriquer les objets demandés avec le nombre minimum de barres ou la perte minimale des résidus.

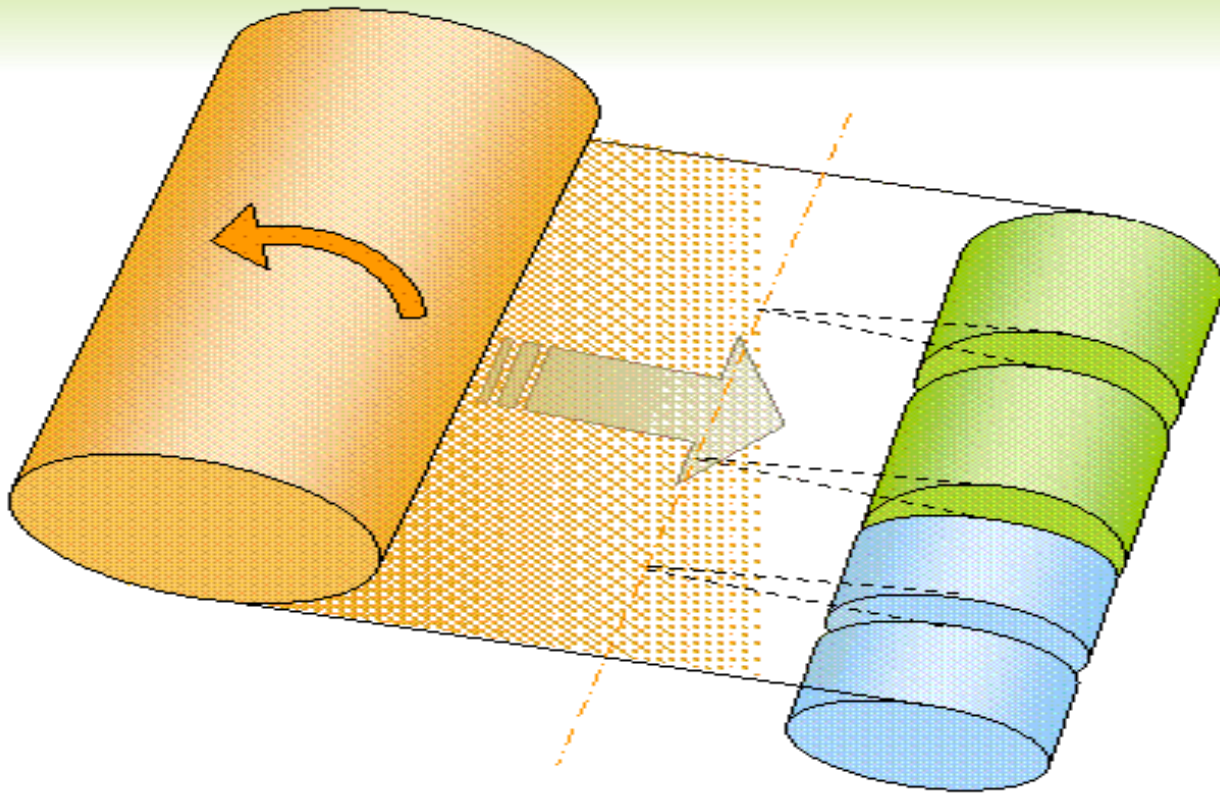
PROBLÈME DE DÉCOUPE

« CUTTING STOCK PROBLEM »



Exemple avec 3 patrons de découpe sans perte

DÉCOUPE DES BOBINES



Problème à **une dimension** en découpant la largeur

EXEMPLE DE PROBLÈME DE DÉCOUPE

Le plan de découpe qui minimise le nombre de bobines de 3 mètres nécessaire pour satisfaire la demande suivante.

Rouleau i	Largeur en cm L_i	Demande D_i
1	80	240
2	100	190
3	140	120
4	180	100

LES PATRONS DE DÉCOUPE

Rouleau	P1	P2	P3	P4	P5	P6	P7	P8	P9
80 cm	3	0	0	1	0	2	1	2	0
100 cm	0	3	0	0	1	1	2	0	1
140 cm	0	0	2	0	0	0	0	1	1
180 cm	0	0	0	1	1	0	0	0	0
Perte	60	0	20	40	20	40	20	0	60

MODÉLISATION DU PROBLÈME DE DÉCOUPE

x_j : le nombre de barres coupées selon le patron j.

d_i : la demande en produits de type i.

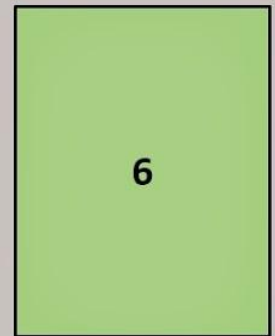
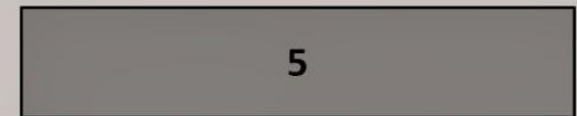
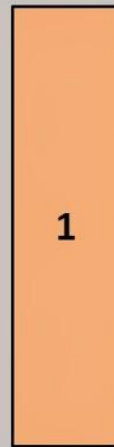
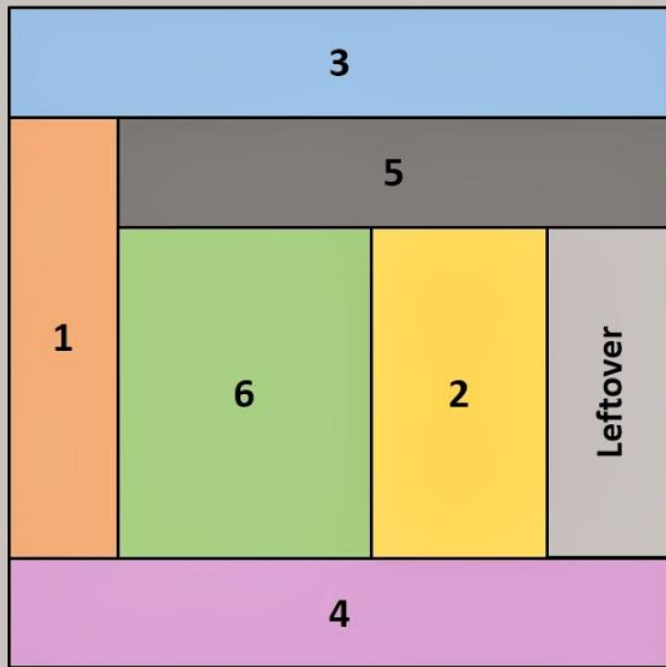
a_{ij} : le nombre de produits i présents dans le patron j.

p_j : la perte associée au patron j.

Minimiser le nombre de bars $\sum_{j=1}^m x_j$ ou la perte $\sum_{j=1}^m p_j x_j$

$$\sum_{j=1}^m a_{ij} x_j \geq d_i, \quad i = 1, \dots, n.$$

PROBLÈME DE DÉCOUPE EN 2D



PROBLÈME DE DÉCOUPE EN 3D

