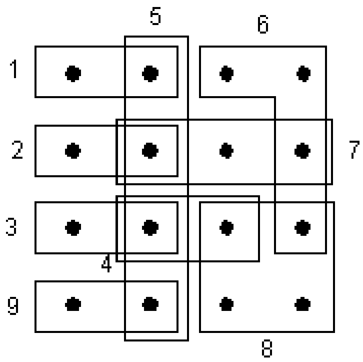
Covering
Set Packing Problem
Partitioning

SCP

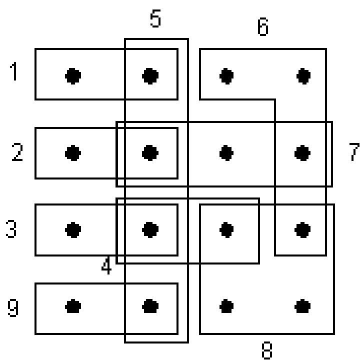
Set Covering location problem (SCP) : énoncé

Localiser le nombre minimum d'éléments pour « couvrir » toutes les demandes



Set Covering location problem (SCP) : énoncé

Localiser le nombre minimum d'éléments pour « couvrir » toutes les demandes

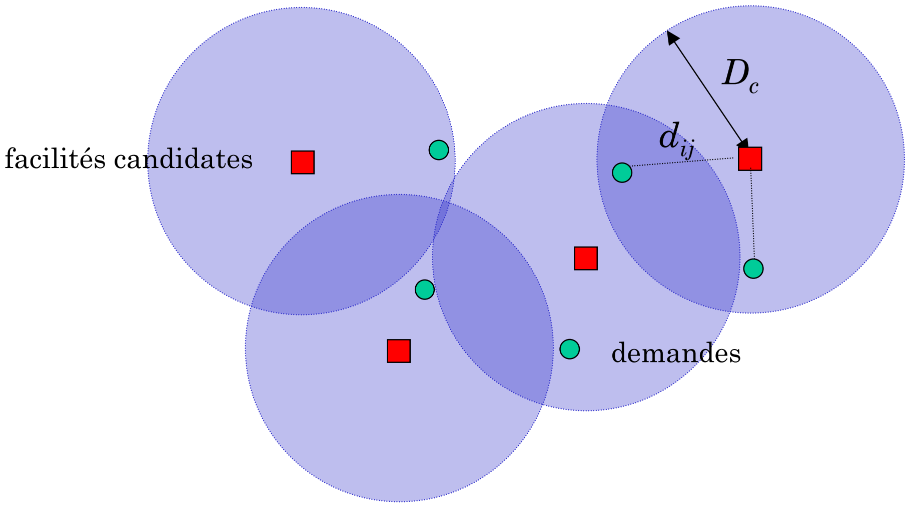


$$\left[\begin{array}{ll} \min z(x) & = \sum_{i=1}^{n} c_i x_i \\ & \sum_{i=1}^{n} t_{il} x_i \geq 1 \quad l = 1, \ldots, k \\ & t_{il} \in \{0,1\} \\ & x_i \in \{0,1\} \end{array}\right]$$

nb: SCP peut être présenté dans la classe des problème de « facility location »

Set Covering location problem (SCP) : énoncé

Localiser le nombre minimum de facilités pour « couvrir » toutes les demandes



Set Covering location problem (SCP) : formalisation

Notations:

- $\boldsymbol{I}$ = l'ensemble des demandes indexés par i
- $\boldsymbol{J}$ = l'ensemble des facilités candidates, indexé par j
- $\boldsymbol{d_{ij}}$ = distance entre la demande i et le site candidat j
- $\boldsymbol{D_c}$ = distance de couverture
- $\boldsymbol{N_i}$ = $\{j \mid d_{ij} \leq D_c\}$: ensemble de tous les localisations candidates pour couvrir le point i

Variables de décision :

- $\boldsymbol{x_j}$ = 1 si on la facilité j est utilisée, 0 sinon

Set Covering location problem (SCP) : modélisation

(1.1’) : minimiser le coût total de la configuration des facilités sélectionnées

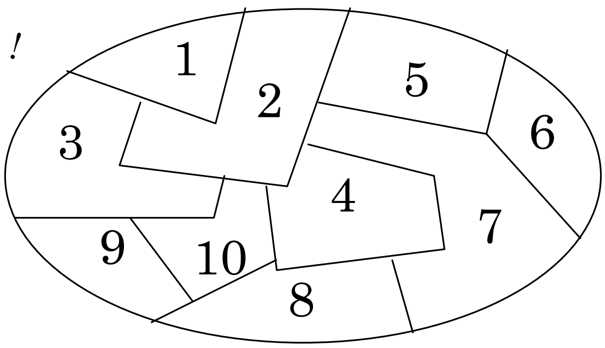
1) Set Covering location problem (SCP)

$$\left[\begin{array}{lll} \min z(x) = \sum_{j\in J} c_j x_j & & (1.1') \\ \text{s/c} & & \\ \sum_{j\in N_i} x_j \geq 1 & \forall i \in I & (1.2) \\ x_j \in \{0,1\} & \forall j \in J & (1.3) \end{array}\right]$$

(1.1) : minimise le nombre de facilités localisées
(1.2) : assure que chaque demande est couverte par au moins une facilité
(1.3) : une facilité est ouverte ou pas

Set Covering location problem (SCP) : illustration

Quel que soit le secteur où vous habitez, SuperBurger a un restaurant près de chez vous !



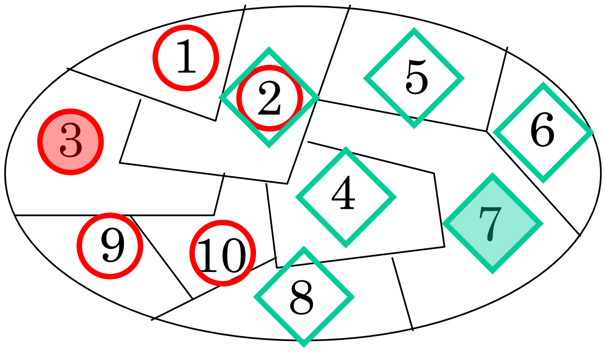
Combien de restaurant ouvrir et où ?

Réponse : 2 en 3 et 7

Problèmes difficiles; au pire :
- n=5 ⇒ 32 possibilités
- n=50 ⇒ 1 125 899 906 842 624 pos.
- n=500 ⇒ 3,27 E150 possibilités

Set Covering location problem (SCP) : illustration

Quel que soit le secteur où vous habitez, SuperBurger a un restaurant près de chez vous !



Combien de restaurant ouvrir et où ?

Réponse : 2 en 3 et 7

Passage à l'échelle; au pire :

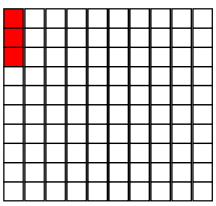
- n=5 ⇒ 32 possibilités
- n=50 ⇒ 1 125 899 906 842 624 pos.
- n=500 ⇒ 3,27 E150 possibilités

⇒ Problèmes difficiles

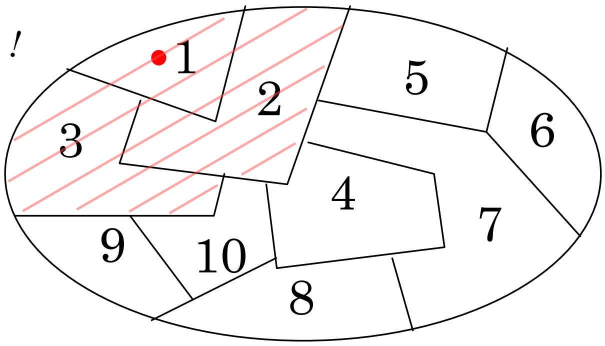
Set Covering location problem (SCP) : illustration

Quel que soit le secteur où vous habitez, SuperBurger a un restaurant près de chez vous !

Si X1=1



région 1 ok
région 2 ok
région 3 ok



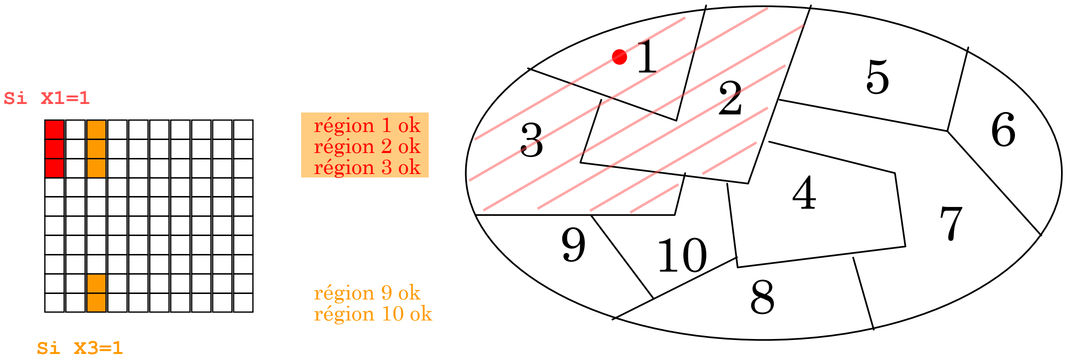
xi = { 1 si un restaurant est ouvert en région i ; 0 sinon }

Un restaurant ouvert en **1** satisfait aussi le slogan pour les régions 2 & 3 :

Chaque région doit vérifier le slogan;
pour la région i : variables ≥ 1 (région i)

x1 + ... ≥ 1 (région 1)
x1 + ... ≥ 1 (région 2)
x1 + ... ≥ 1 (région 3)

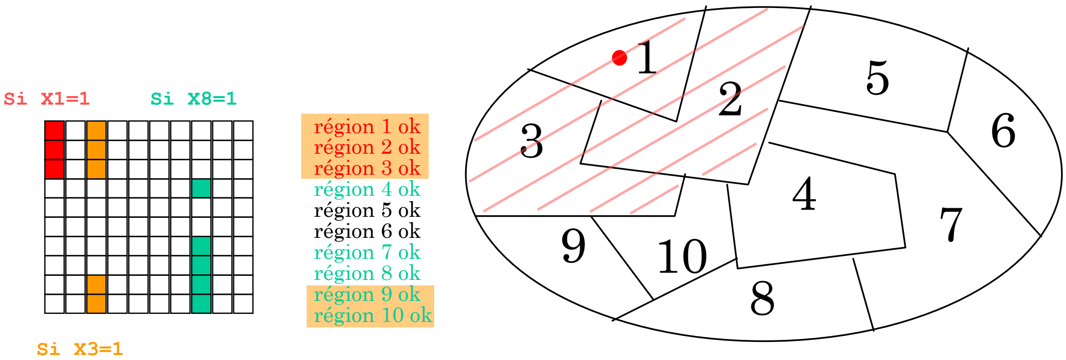
Set Covering location problem (SCP) : illustration



Un restaurant ouvert en **1**, **3** satisfait aussi le slogan pour les régions :

$$
\begin{array}{lllllll}
x1 + \dots + x3 + \dots & \geq & 1 & \text{(région 1)} \\
x1 + \dots + x3 + \dots & \geq & 1 & \text{(région 2)} \\
x1 + \dots + x3 + \dots & \geq & 1 & \text{(région 3)} \\
\quad \vdots \qquad \vdots & & & \\
\dots + x3 + \dots & \geq & 1 & \text{(région 9)} \\
\dots + x3 + \dots & \geq & 1 & \text{(région 10)}
\end{array}
$$

Set Covering location problem (SCP) : illustration



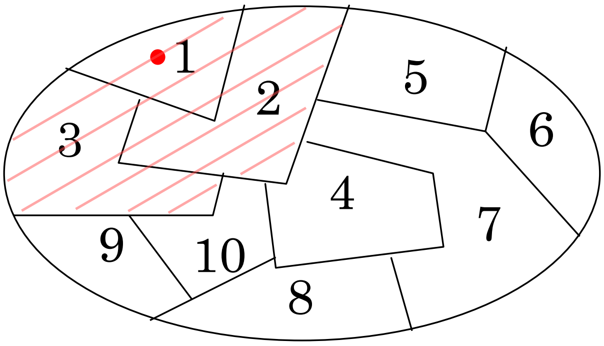
Un restaurant ouvert en **1**, **3**, **8** satisfait aussi le slogan pour les régions :

$$
\begin{array}{lllllll}
x1 + \dots + x3 + \dots & \geq & 1 & \text{(région 1)} \\
x1 + \dots + x3 + \dots & \geq & 1 & \text{(région 2)} \\
x1 + \dots + x3 + \dots & \geq & 1 & \text{(région 3)} \\
\dots + x8 + \dots & \geq & 1 & \text{(région 4)} \\
\quad \vdots \qquad \vdots & & & \\
\dots + x8 + \dots & \geq & 1 & \text{(région 7)} \\
\dots + x8 + \dots & \geq & 1 & \text{(région 8)} \\
\dots + x3 + \dots + x8 + \dots & \geq & 1 & \text{(région 9)} \\
\dots + x3 + \dots + x8 + \dots & \geq & 1 & \text{(région 10)}
\end{array}
$$

Set Covering location problem (SCP) : illustration

$N_i \quad = \{ j \mid d_{ij} \leq D_c \}$: ensemble de tous les localisations candidates pour couvrir le point i

$N_1 = \{1,2,3\}$
$N_2 = \{1,2,3,4,5,7,10\}$
$N_3 = \{1,2,3,9,10\}$
$N_4 = \{2,4,7,8,10\}$
$N_5 = \{2,5,6,7\}$
$N_6 = \{5,6,7\}$
$N_7 = \{2,4,5,6,7,8\}$
$N_8 = \{4,7,8,9,10\}$
$N_9 = \{3,8,9,10\}$
$N_{10} = \{2,3,4,8,9,10\}$




```
x1 + … + x3 + …                  ≥ 1 (région 1)
x1 + … + x3 + …                  ≥ 1 (région 2)
x1 + … + x3 + …                  ≥ 1 (région 3)
                … + x8 + …       ≥ 1 (région 4)
                  ⋮       ⋮
                … + x8 + …       ≥ 1 (région 7)
                … + x8 + …       ≥ 1 (région 8)
      … + x3 + … + x8 + …        ≥ 1 (région 9)
      … + x3 + … + x8 + …        ≥ 1 (région 10)
```


Set Covering location problem (SCP) : modélisation du problème

$$\left[\begin{array}{lll} \min z(x) & = \sum_{i=1}^{n} c_i x_i & \\ & \sum_{i=1}^{n} t_{il} x_i \geq 1 & l = 1,\ldots,k \\ & t_{il} \in \{0,1\} & \\ & x_i \in \{0,1\} & \end{array}\right]$$

```
MIN z = X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10

s/c     X1 + X2 + X3                              ≥ 1
        X1 + X2 + X3 + X4 + X5 + X7 + X10         ≥ 1
        X1 + X2 + X3 + X9 + X10                   ≥ 1
        X2 + X4 + X7 + X8 + X10                   ≥ 1
        X2 + X5 + X6 + X7                         ≥ 1
        X5 + X6 + X7                              ≥ 1
        X2 + X4 + X5 + X6 + X7 + X8               ≥ 1
        X4 + X7 + X8 + X9 + X10                   ≥ 1
        X3 + X8 + X9 + X10                        ≥ 1
        X2 + X3 + X4 + X8 + X9 + X10              ≥ 1

        X1, X2, X3, X4, X5, X6, X7, X8, X9, X10  =  (0,1)
```

Vérification de la solution trouvée :

```
           0     0     1     0     0     0     1     0     0     0

MIN z = X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10

s/c       X1 + X2 + X3                                          ≥ 1
          X1 + X2 + X3 + X4 + X5 +      X7 +            X10     ≥ 1
          X1 + X2 + X3 +                          X9 + X10      ≥ 1
               X2 +      X4 +           X7 + X8 +       X10     ≥ 1
               X2 +           X5 + X6 + X7                      ≥ 1
                              X5 + X6 + X7                      ≥ 1
               X2 +      X4 + X5 + X6 + X7 + X8                 ≥ 1
                         X4 +           X7 + X8 + X9 + X10      ≥ 1
                    X3 +                     X8 + X9 + X10      ≥ 1
               X2 + X3 + X4 +                X8 + X9 + X10      ≥ 1

          X1,  X2,  X3,  X4,  X5,  X6,  X7,  X8,  X9,  X10  =  (0,1)
```


Règles de réduction d'une instance de SCP

```
MIN z = X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10

s/c       X1 + X2 + X3                                          ≥ 1
          X1 + X2 + X3 + X4 + X5 +      X7 +            X10     ≥ 1
          X1 + X2 + X3 +                          X9 + X10      ≥ 1
               X2 +      X4 +           X7 + X8 +       X10     ≥ 1
               X2 +           X5 + X6 + X7                      ≥ 1
                              X5 + X6 + X7                      ≥ 1
               X2 +      X4 + X5 + X6 + X7 + X8                 ≥ 1
                         X4 +           X7 + X8 + X9 + X10      ≥ 1
                    X3 +                     X8 + X9 + X10      ≥ 1
               X2 + X3 + X4 +                X8 + X9 + X10      ≥ 1

          X1,  X2,  X3,  X4,  X5,  X6,  X7,  X8,  X9,  X10  =  (0,1)
```

Règles de réduction d'une instance de SCP

MIN z = X2 + X3 + X4 + X5 + X7 + X8 + X10

s/c X2 + X3 ≥ 1
~~X2 + X3 + X4 + X5 + X7 + X10 ≥ 1~~
~~X2 + X3 + X10 ≥ 1~~
~~X2 + X4 + X7 + X8 + X10 ≥ 1~~
~~X2 + X5 + X7 ≥ 1~~
X5 + X7 ≥ 1
~~X2 + X4 + X5 + X7 + X8 ≥ 1~~
X4 + X7 + X8 + X10 ≥ 1
X3 + X8 + X10 ≥ 1
~~X2 + X3 + X4 + X8 + X10 ≥ 1~~

X2, X3, X4, X5, X7, X8, X10 = (0,1)

Règles de réduction d'une instance de SCP

MIN z = X2 + X3 + X4 + X5 + X7 + X8 + ~~X10~~

s/c X2 + X3 ≥ 1
X5 + X7 ≥ 1
X4 + X7 + X8 + ~~X10~~ ≥ 1
X3 + X8 + ~~X10~~ ≥ 1

X2, X3, X4, X5, X7, X8, ~~X10~~ = (0,1)

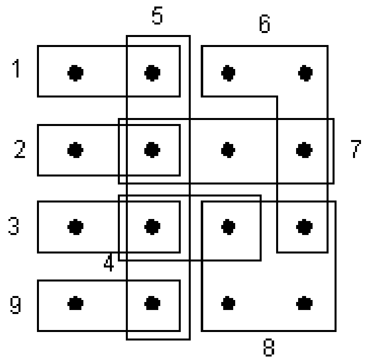
Règles de réduction d'une instance de SCP

$$
\begin{array}{llcl}
\text{MIN } z = & X2 + X3 + X4 + X5 + X7 + X8 & & \\
\text{s/c} & X2 + X3 & \geq & 1 \\
& X5 + X7 & \geq & 1 \\
& X4 + X7 + X8 & \geq & 1 \\
& X3 + X8 & \geq & 1 \\
& X2,\ X3,\ X4,\ X5,\ X7,\ X8 & = & (0,1)
\end{array}
$$

Problèmes proches : SPP

Enoncé

On considère un problème comprenant 16 éléments (ronds noir) et 9 ensembles (rectangles, carrés, L). Un élément peut appartenir à un ou plusieurs ensembles comme illustré dans la figure suivante :



Par exemple, l'ensemble 5 contient 4 éléments. On note aussi que un élément de l'ensemble 5 figure aussi dans les ensembles 2 et 7. On dira que ces trois ensembles sont incompatibles entre eux suite à cet élément en commun. A chaque ensemble correspond un profit.

Une solution optimale de ce problème sera une sélection de ces ensembles disjoints 2 à 2 qui maximise le profit.

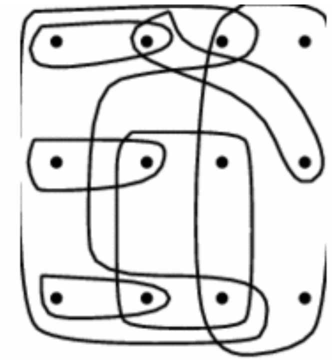
Modélisation

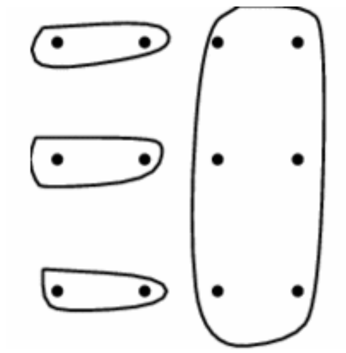
$$\left[\begin{array}{ll} \max z(x) & = \sum_{i=1}^{n} c_i x_i \\ & \sum_{i=1}^{n} t_{il} x_i \leq 1 \quad l = 1, \ldots, k \\ & t_{il} \in \{0,1\} \\ & x_i \in \{0,1\} \end{array}\right]$$

SPP : Set Packing Problem

(problème de couplage généralisé)

Autre exemple





Autre exemple

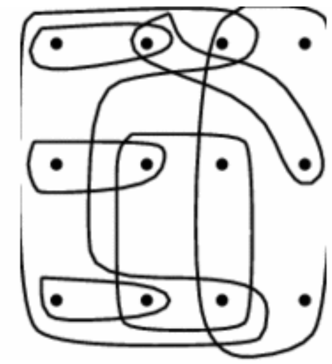
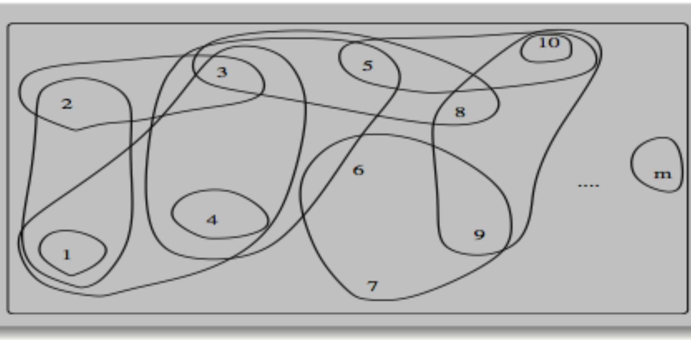


Illustration d’un SPP

Transport ferroviaire

Situation: capacité d’une infrastructure ferroviaire

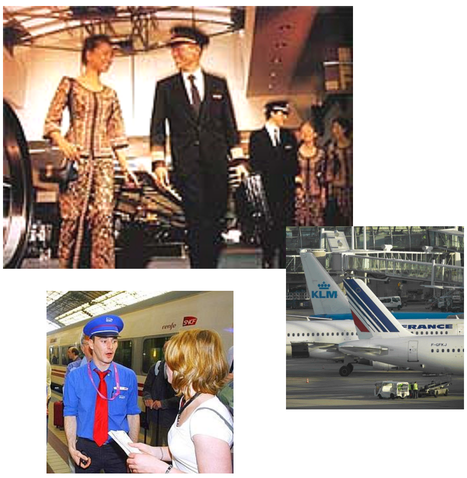
Problèmes proches :
SPA

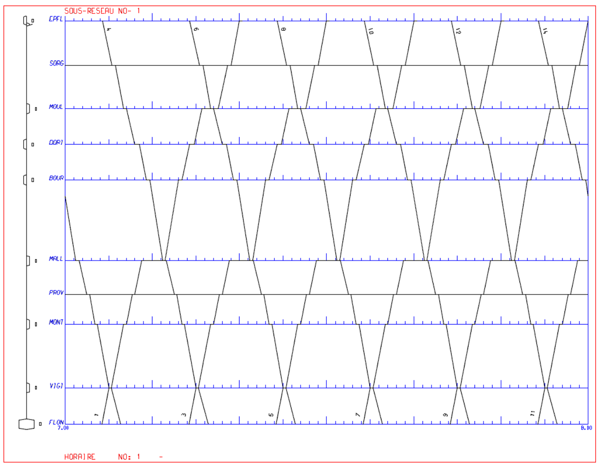


1	1	1	0	0	1	0	0	0	0	0	0		0	=	1
2	0	1	1	0	0	0	0	0	0	0	0		0	=	1
3	0	0	1	0	1	1	1	0	0	0	0		0	=	1
4	0	0	0	1	1	0	0	0	0	0	0		0	=	1
5	0	0	0	0	0	1	1	1	0	0	0		0	=	1
6	0	0	0	0	0	0	0	0	0	1	0		0	=	1
7	0	0	0	0	0	0	0	0	0	1	0	...	0	=	1
8	0	0	0	0	0	0	1	0	1	0	0		0	=	1
9	0	0	0	0	0	0	0	0	1	1	0		0	=	1
10	0	0	0	0	0	0	0	1	1	0	1		0	=	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮
m	0	0	0	0	0	0	0	0	0	0	0		1	=	1
	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}		c_n		

Optimisation combinatoire (6)

Emplois du temps et gestion de personnel





Ex: équipages de vol, plan de roulement,...

Exercice (pairing)

FLP:

Facility Location problems

Facility Location:

Par « facilités », on entend des

- **Centres de secours (18)** (Schilling et al. 1980)
- **Dépôts (de brasseries)** (Gelders et al. 1987)
- **Écoles** (Tewari et Sidheswar, 1987)
- **Arrêts de bus** (Gleason, 1975)
- **Centres de santé** (Abernathy et Hershey, 1972)
- **Routeurs** (Pirkul, 1987)
- **Orbites de satellites** (Drezner, 1988)
- **Restaurants fast-foods** (Min, 1987)
- **Etc.**

à « disposer » géographiquement.

Facility Location:

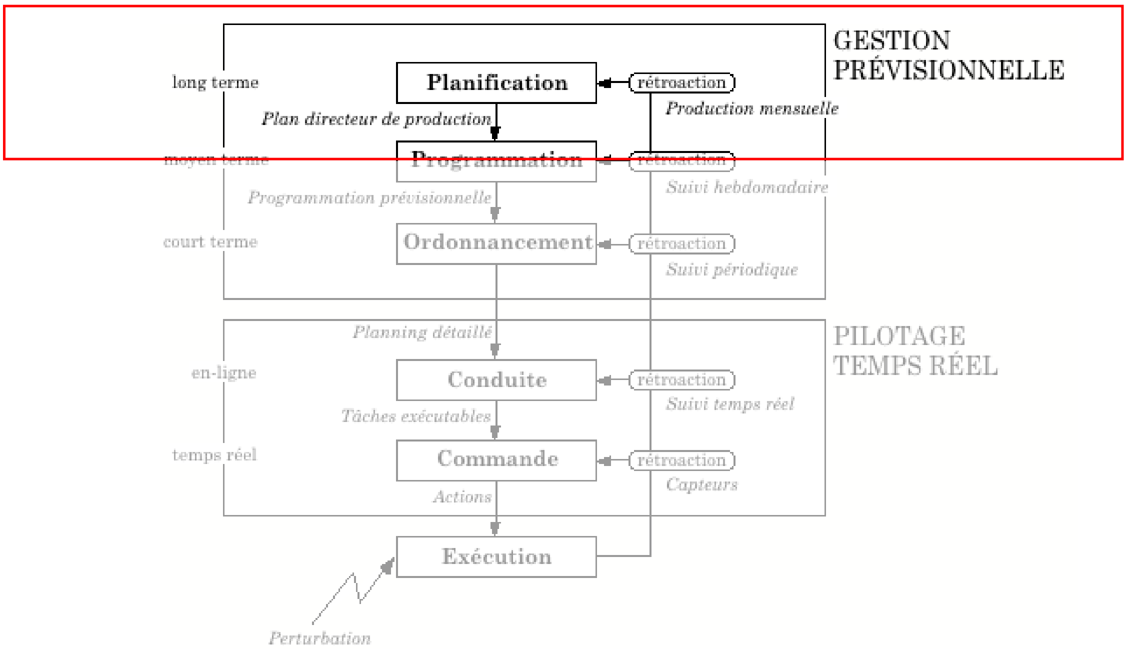
Des problèmes d'optimisations (1/2)

1) **Rencontrés à tous les niveaux des organisations**
   - Depuis la personne isolée
   - Jusqu'à la multinationale, gouvernements,…

2) **Concernent des décisions souvent de nature stratégiques**

3) **Imposent des considérations économiques externes**

Facility Location:



Facility Location:

Des problèmes d'optimisations (1/2)

1) **Rencontrés à tous les niveaux des organisations**
   - Depuis la personne isolée
   - Jusqu'à la multinationale, gouvernements,…

2) **Concernent des décisions souvent de nature stratégiques**
   - Mobilisent de gros capitaux
   - Impacts sur le long terme

3) **Imposent des considérations économiques externes**
   - Pollution
   - Congestion
   - Développement économique
   - Etc.

Facility Location:

Des problèmes d'optimisations (2/2) :

4) Souvent extrèmement difficiles à résoudre exactement
- Complexité calculatoire : résoudre que des petits problèmes

5) Souvent intimement liés à un problème spécifique
- Objectifs, contraintes, variables : déterminés par le problème traité

⇒ **Il n'existe pas un modèle de localisation général approprié pour toutes les applications potentielles ou existantes**

Facility Location problems

Modèles

Facility Location; Applications and Theory
Zvi Drezner and Horst W. Hamacher Edts
Springer. 2004

Modèles de Facility Location

Huit modèles discrets de base :

- **Set Covering**
- **Maximal Covering**
- **[vertex] *p*-center**
- ***p*-dispersion**
- ***p*-median**
- **Fixed charge**
- **Hub**
- **Maxisum**

Points communs :

- Le réseau sous-jacent est donné
- La localisation des demandes par les facilités
- La localisation des facilités existantes

Problème général :

Localiser de nouvelles facilités au regard de critères (temps de parcours, coûts, satisfaction de la demande, etc) càd souvent une mesure de « distance »

Modèles de Facility Location: fondés sur la distance maximale

2) Maximal Covering (MCP) location problem

L'hypothèse du SCP : toutes les demandes doivent être couvertes, pas de contrainte de budjet → introduire une telle contrainte.

Une ville souhaiterai avoir une école élémentaire dans chaque quartier accessible en marche à pied pour tous les élèves devant se rendre à l'école. Mais ce souhait risque d'exiger la construction de plus d'écoles qu'il n'a été envisagé de construire par la ville.

Le Maximal Covering location problem :

- traite des situations où **une limite supérieure existe** sur le nombre de facilités à ouvrir;
- suppose qu'il n'y a **pas assez de facilités** pour couvrir l'ensemble des demandes;
- a pour objectif de localiser un nombre prédéterminé de facilités, *p*, de façon à **maximiser la demande qui est couverte**.

Modèles de Facility Location:
fondés sur la distance maximale

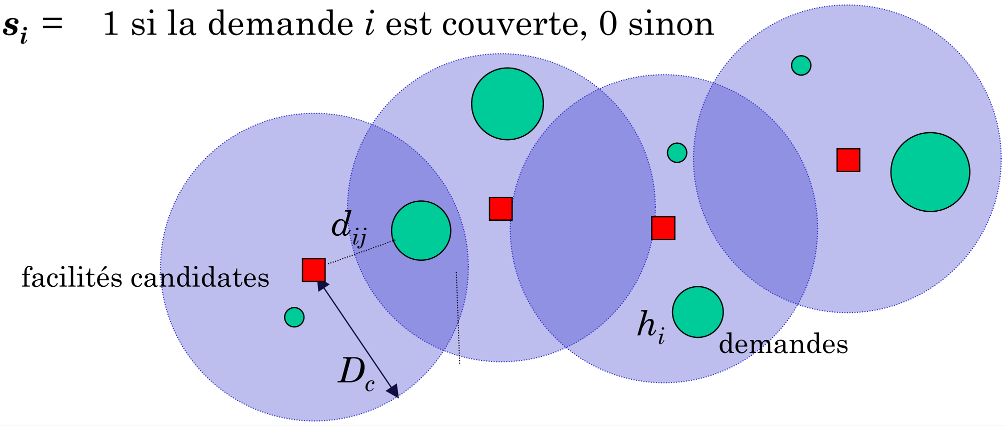
2) Maximal Covering (MCP) location problem

<u>Notations du SCP complétées</u> :

$\boldsymbol{h_i}$ = la demande au point i

$\boldsymbol{p}$ = le nombre maximum de facilités à localiser

$\boldsymbol{s_i}$ = 1 si la demande i est couverte, 0 sinon



Modèles de Facility Location:
fondés sur la distance maximale

2) Maximal Covering (MCP) location problem

$$
\left[
\begin{array}{lllr}
\textbf{max } z(x) = \sum_{i \in I} h_i s_i & & & (1.4) \\
\text{s/c} & & & \\
\sum_{j \in N_i} x_j - s_i \geq 0 & \forall i \in I & & (1.5) \\
\sum_{j \in J} x_j = p & & & (1.6) \\
x_j \in \{0,1\} & \forall j \in J & & (1.7) \\
s_i \in \{0,1\} & \forall i \in I & & (1.8)
\end{array}
\right]
$$

(1.4) : maximise la demande totale couverte

(1.5) : assure que la demande en i n'est pas comptée comme couverte tant qu'une facilité candidate n'est pas localisé pour couvrir cette demande

(1.6) : limite le nombre de facilités à ouvrir

Modèles de Facility Location:
fondés sur la distance maximale

Le Set Covering (SCP) location problem et le Maximal Covering (MCP) location problem posent comme hypothèse que
***D_c*, la distance de couverture est fixée, prédéterminée.**

Hypothèse valide dans plusieurs situations, mais pour d'autres cas
***D_c* peut se présenter comme un but à atteindre.**

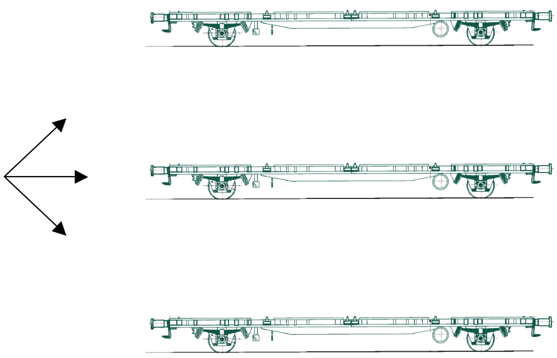
> *Concernant l'ouverture de bibliothèques publiques, il est souhaitable par mesure d'équité de minimiser la distance maximale qu'aura a parcourir un citoyen pour se rendre à la bibliothèque*

$\Rightarrow$ ***p*-center problem** (pondéré -ie importance de la demande- ou pas) :

Minimiser ou maximiser la distance d'une demande par rapport à des facilités proches données à ouvrir parmi un ensemble prédéfini de facilités

Problème intermédiaire:
formulation d'un problème min-max

Enoncé :



Trois wagons de charge utile limitée à
100 quintaux sont mobilisés pour transporter 16 containers (de poids donnés).
Comment affecter les containers aux wagons de façon à respecter les charges utiles maximales et à minimiser la charge du wagon le plus chargé?

Problème intermédiaire: **formulation d'un problème min-max**

Modèle :

$$\left[\begin{array}{lll} \min & W & \\ \text{s/c} & \sum_{j=1}^{m} x_{ij} = 1 & \forall i = 1,\dots,n \\ & \sum_{i=1}^{n} p_i x_{ij} \leq W & \forall j = 1,\dots,m \\ & x_{ij} \in \{0,1\} & \forall i = 1,\dots,n,\ j = 1,\dots,m \\ & W \geq 0 & \\ & W \leq W_{max} & \end{array}\right]$$

W : charge des wagons – W_{max} : charge maximale des wagons
i : les containers – j : les wagons – p_i : poids du container i
x_{ij} : vaut 1 si le container i est affecté au wagon j, 0 sinon

Contrainte 1 : chaque container est affecté à un seul wagon
Contrainte 2 : majore la charge de chaque wagon par W

Modèles de Facility Location: fondés sur la distance à optimiser

3) [vertex] *p*-center problem

<u>Notations précédentes complétées</u> :

W = la distance maximale entre une demande et la facilité qui lui sera assignée

y_{ij} = 1 si la demande i est assignée à la facilité j, 0 sinon

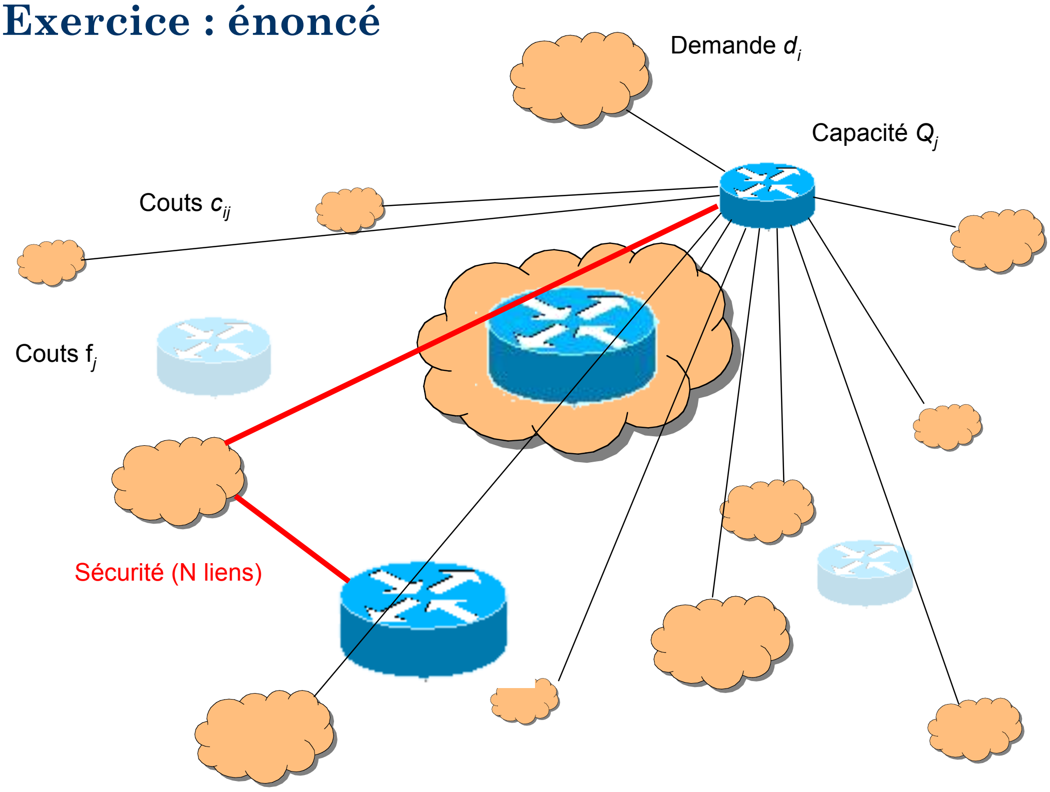
Modèles de Facility Location:
fondés sur la distance à optimiser

3) [vertex] *p*-center problem

$$
\left[
\begin{array}{llll}
\min z(x) & = W & & (1.10) \\
\text{s/c} & \sum_{j \in J} x_j = p & & (1.11) \\
& \sum_{j \in J} y_{ij} = 1 & \forall i \in I & (1.12) \\
& y_{ij} - x_j \leq 0 & \forall i \in I,\, j \in J & (1.13) \\
& W - \sum_{j \in J} h_i d_{ij} y_{ij} \geq 0 & \forall i \in I & (1.14) \\
& x_j \in \{0,1\} & \forall j \in J & (1.15) \\
& y_{ij} \in \{0,1\} & \forall i \in I,\, j \in J & (1.16)
\end{array}
\right]
$$

(1.10) : minimise la demande pondérée maximale entre chaque demande et sa facilité la plus proche
(1.12) : chaque demande doit être affecté exactement à une facilité
(1.13) : l'affectation d'une demande ne peut se faire qu'aux facilités ouvertes
(1.14) : borne inférieure sur la distance de la demande pondérée maximale

Exercice : énoncé



Exercice : modèle mathématique

$$\left[\begin{array}{lll} \min \text{Cost} = \sum_{j=1}^{J} f_j y_j + \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij} x_{ij} & & \\ \text{s/c} \quad \sum_{j=1}^{J} x_{ij} = N & & \forall i \in \{1 \ldots I\} \\ \sum_{i=1}^{I} \frac{d_i}{N} x_{ij} \leq y_j Q_j & & \forall j \in \{1 \ldots J\} \\ y_j \in \{0,1\}, x_{ij} \in \{0,1\} & & \end{array}\right]$$

Exercice : le modèle sous MPL :

```
TITLE        FacilityLocationProblem;

INDEX        zones   := (z1, z2, z3, z4, z5, z6);
             sites   := (s1, s2, s3, s4);

DATA         CoutOuverture[sites]           := ( 1000,  750,  500,  500 );
             CapaciteSite[sites]            := (  200,  200,  100,  100 );
             CoutRaccordement[zones, sites] := (   10,   50,  100,  140,  140,  120,
                                                  150,  100,   50,   90,  250,  170,
                                                  330,  380,  330,  370,  470,  210,
                                                  360,  370,  320,  280,  220,  440 );
             demandeZone[sites]             := (   50,   20,   60,   20,   35,   20 );
             NbrConnexion                   := 2;

VARIABLES    Ouverture[sites]             -> Ouve;
             Raccordement[zones, sites]   -> Racc;

MACROS       TotalInstallation   := SUM(sites:  CoutOuverture * Ouverture);
             TotalRaccordement   := SUM(zones, sites:  CoutRaccordement * Raccordement);

MODEL        MIN Cout  =  TotalInstallation + TotalRaccordement;

SUBJECT TO   Raccordements[zones] -> CRacc:
               SUM(sites:  Raccordement)  = NbrConnexion;

             Consommation[sites] -> CCons:
               SUM(zones: demandeZone[sites]/NbrConnexion * Raccordement[zones, sites] )
                                            <= Ouverture[sites] * demandeZone[sites]   ;
INTEGER      Ouverture;
             Raccordement;
END
```

Problèmes de packing
- bin packing
- 2D packing

Bin-packing : stockage de mp3

2-Dimensional packing problem (1)

Input:

A set of rectangles $I = \{1,2,\ldots,n\}$
each i has several modes.

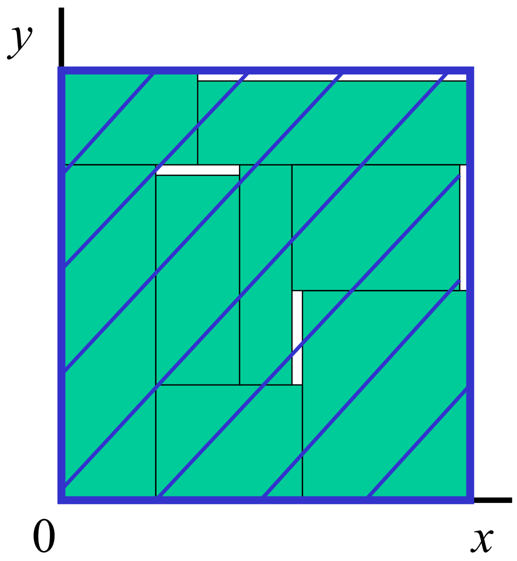
mode: (width, height, spatial cost functions)

Output:

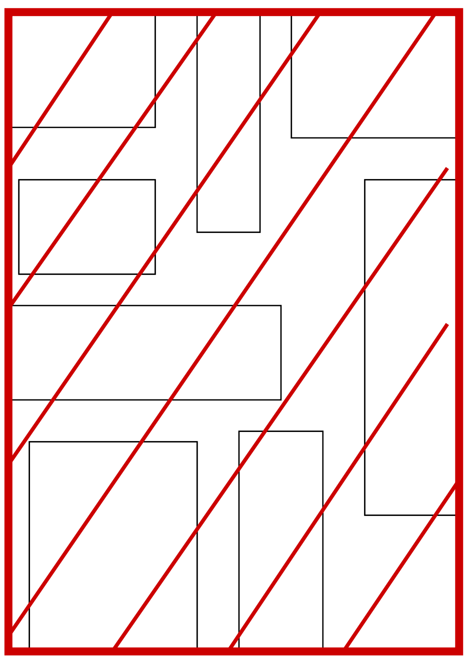
Modes and x, y coordinates of all rectangles.

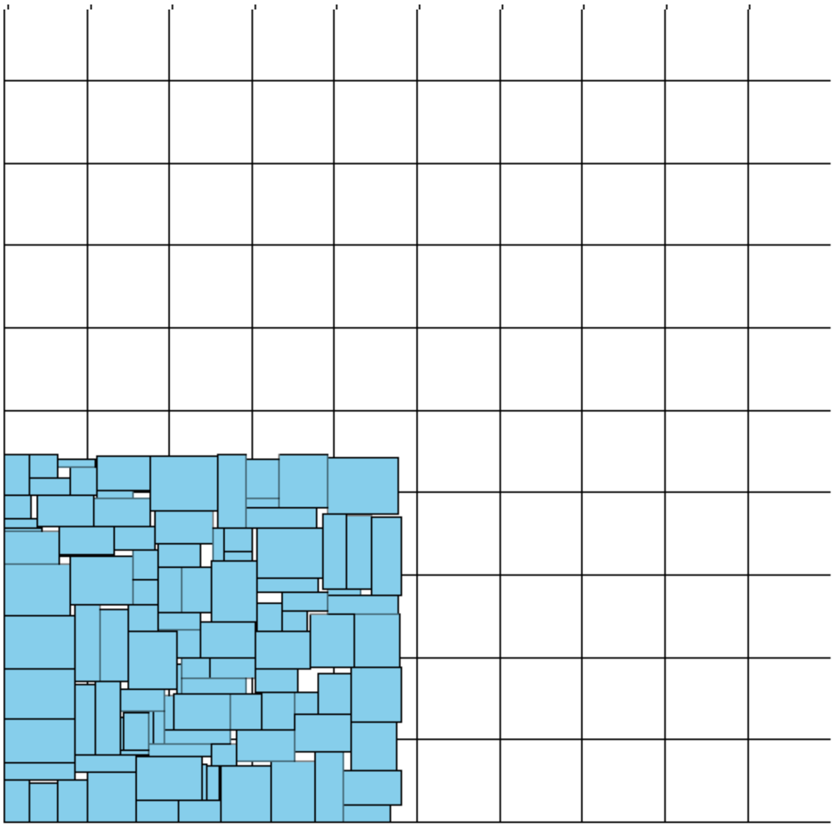
It is asked to place all rectangles in the plane without overlap so that the objective function is minimized.

2-Dimensional packing problem (2)



smallest area packing





2-Dimensional packing problem (3)

Ibaraki et al, 2002

Une modélisation possible sous la forme d'un
Problème de partitionnement d'ensembles :

P.C. Gilmore, R.E. Gomory,
Multistage cutting problems of two and more dimensions,
Operations Research, 13 (1965) 94−119.

Idées :
- principe d'énumération de l'ensemble des possibles (lourd!)
- algorithme de génération de colonnes pour le résoudre

Mosaïque récapitulative