Recherche Opérationelle
Alternance

Moritz Mühlenthaler, Zoltán Szigeti

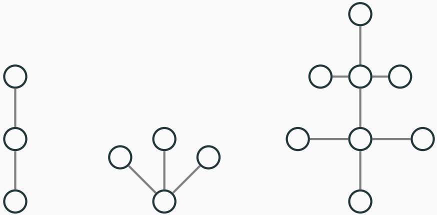
moritz.muehlenthaler@grenoble-inp.fr

G-SCOP (Sciences pour la Conception, l'Optimisation et la Production de Grenoble)
Équipe OC (Optimisation Combinatoire)

Arbres Couvrants

Arbres et Forêts

- forêt : graphe simple sans cycle
- arbre : graphe simple connexe sans cycle



Une composante connexe d'une forêt est un arbre.
Un sous-graphe partiel d'un arbre est une forêt.

Arbres : Propriétés Utiles

Soit $T = (V, E)$ un graphe d'ordre $n \geq 2$. Les propriétés suivantes sont équivalentes :

1. T est un arbre. (connexe et sans cycle)
2. T est sans cycle et admet $n - 1$ arêtes.
3. T est connexe-minimal. (pour une arête e quelconque, $T - e$ n'est pas connexe)
4. T est acyclique-maximal. (pour deux sommets non-reliés x, y, le graphe $T + xy$ contient un cycle)
5. T est sans boucle et tout couple de sommets est relié par une chaîne unique.

Arbres : Propriétés Utiles

Deux sommets quelconques d'un arbre T sont reliés par une chaîne unique.

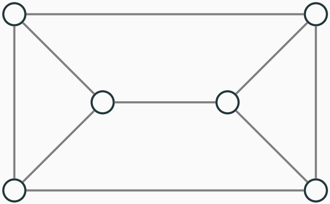
Soit T un arbre d'ordre ≥ 2. Alors

1. T possède un sommet v de degré 1 et
2. $T - v$ est un arbre.

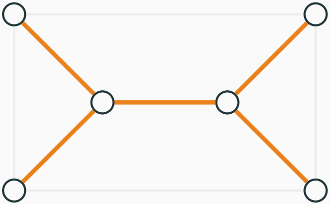
Un arbre T d'ordre n possède $n - 1$ arêtes.

Arbres Couvrants

arbre couvrant : sous-graphe partiel qui est un arbre.



Graphe G



Arbre couvrant de G

Arbres Couvrants

Un graphe G admet un arbre couvrant $\Leftrightarrow$ G est connexe.

Démonstration.

- $\Rightarrow$ Si G admet un arbre couvrant T, alors T est connexe, donc G l'est aussi.
- $\Leftarrow$ Supposons que G est connexe. Soit T un sous-graphe partiel connexe-minimal de G. Alors T est un arbre (Propriété 3).

∎

Arbres Couvrants de Coût Minimum

Motivation : Relier tout le monde à tout le monde avec un coût minimum.

- graphe pondéré : graphe $G = (V, E)$ avec poids/coûts $c : E \to \mathbb{Z}$.
- c-coût de $F \subseteq E$: $\sum_{e \in F} c(e)$

Problème : Étant donné un graphe connexe pondéré G, c, trouver un arbre couvrant de c-coût minimum.

Algorithmes : Principe

On commence avec un graphe partiel sans arête et on ajoute une arête à chaque étape jusqu'à ce qu'on on obtienne un graphe connexe.

Comment choisir la prochaine arête ?

- Ajouter la meilleure arête possible de telle sorte que le graphe obtenu reste une forêt (Algorithme de Kruskal), ou
- Ajouter la meilleure arête possible de telle sorte que le graphe obtenu reste un arbre (Algorithme de Prim)

Algorithmes : Principe

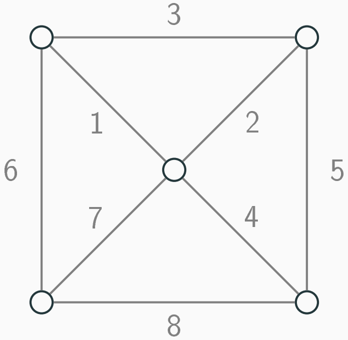
On commence avec un graphe partiel sans arête et on ajoute une arête à chaque étape jusqu'à ce qu'on on obtienne un graphe connexe.

Comment choisir la prochaine arête ?

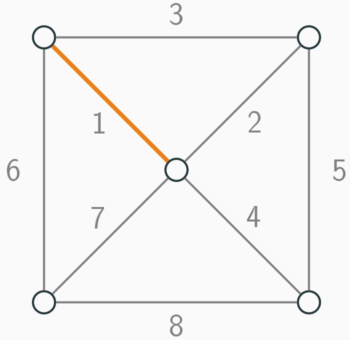
- Ajouter la meilleure arête possible de telle sorte que le graphe obtenu reste une forêt (Algorithme de Kruskal), ou
- Ajouter la meilleure arête possible de telle sorte que le graphe obtenu reste un arbre (Algorithme de Prim)

Alternative : l'algorithme pessimiste supprime la pire arête possible de telle sorte que le graphe reste connexe.

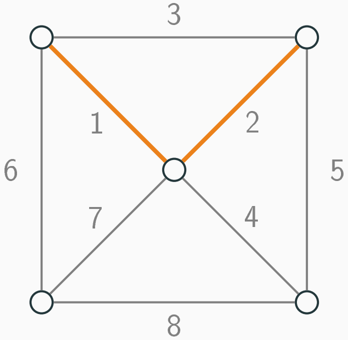
Algorithme de Kruskal : Exemple

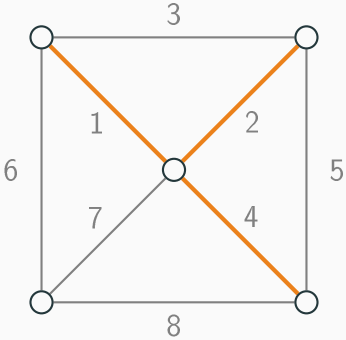


Algorithme de Kruskal : Exemple

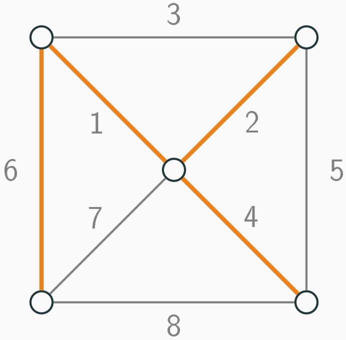


Algorithme de Kruskal : Exemple




3
1
2
6
5
7
4
8

Algorithme de Kruskal : Exemple



Algorithme de Kruskal

Algorithme 1 : Algo de KRUSKAL

Entrée : Graphe connexe $G = (V, E)$, poids $c : E \to \mathbb{Z}$

Sortie : Arbre couvrant H de G

trier les arêtes de G par ordre de coût croissant
$c(e_1) \leq c(e_2) \leq \cdots \leq c(e_m)$
$F_0 \leftarrow \emptyset$
pour $1 \leq i \leq m$ **faire**
    si $F_{i-1} + e_i$ *est une forêt* **alors** $F_i := F_{i-1} + e_i$
    sinon $F_i := F_{i-1}$
$H \leftarrow (V, F_m)$; **fin**.

Algorithme de Kruskal : Justification (1)

On note : $H_i := (V, F_i)$

Le graphe H est un arbre couvrant de G.

À chaque étape i, il existe un arbre couvrant de poids minimum qui contient H_i.

Algorithme de Prim

Algorithme 2 : Algo de PRIM

Entrée : Graphe connexe $G = (V, E)$, poids $c : E \to Z$

Sortie : Arbre couvrant H de G

$S_0 \leftarrow \{u\}$, u est un sommet quelconque

$F_0 \leftarrow \emptyset$

pour $1 \leq i < n$ **faire**

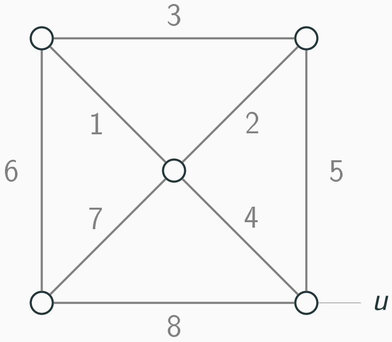
  $vw \leftarrow$ arête de poids minimum dans $\delta(S_{i-1})$

  $S_i \leftarrow S_{i-1} \cup \{v, w\}$

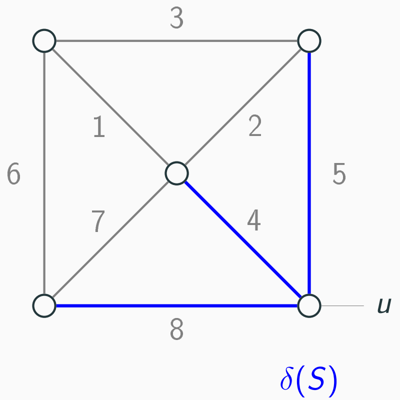
  $F_i \leftarrow F_{i-1} \cup \{vw\}$

$H \leftarrow (V, F_{n-1})$; **fin**.

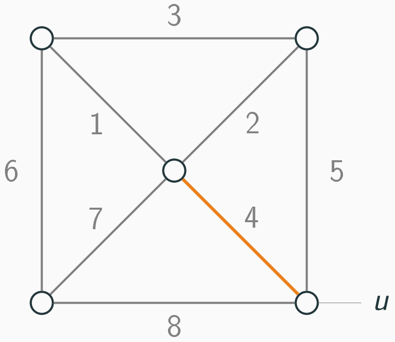
Algorithme de Prim : Exemple



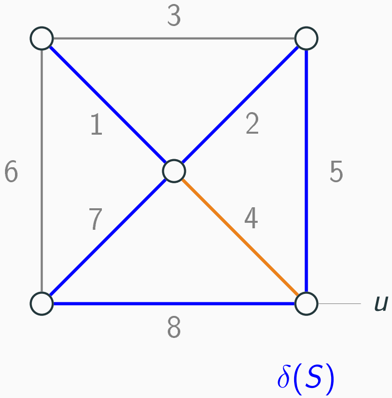
Algorithme de Prim : Exemple



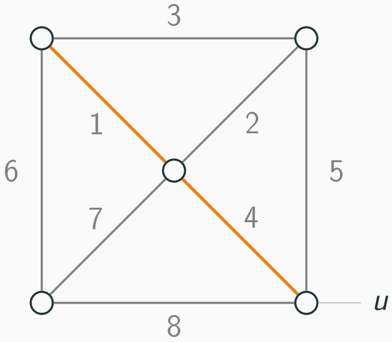
Algorithme de Prim : Exemple



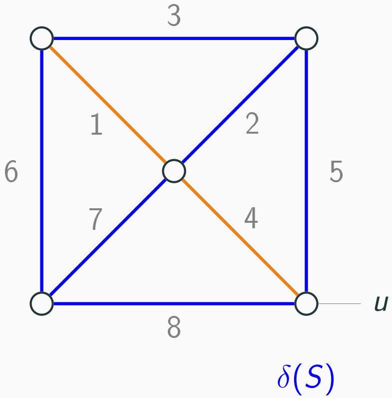
Algorithme de Prim : Exemple



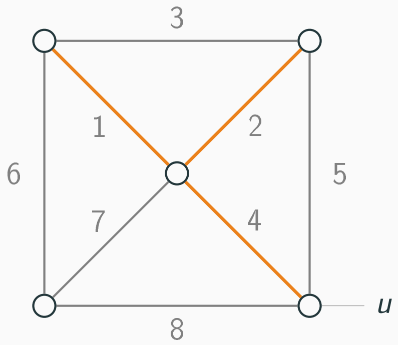
Algorithme de Prim : Exemple



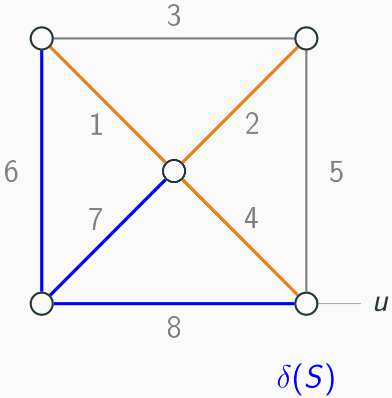
Algorithme de Prim : Exemple



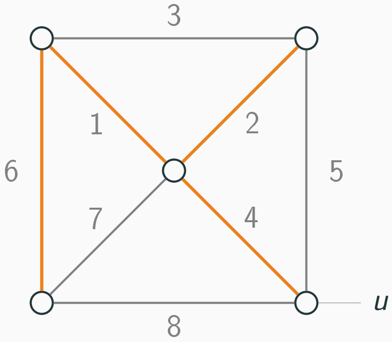
Algorithme de Prim : Exemple



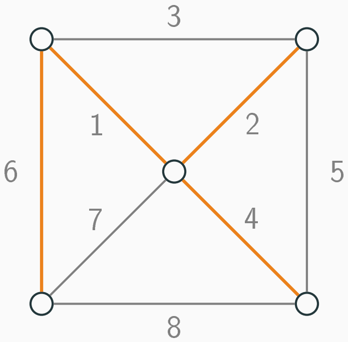
Algorithme de Prim : Exemple



Algorithme de Prim : Exemple

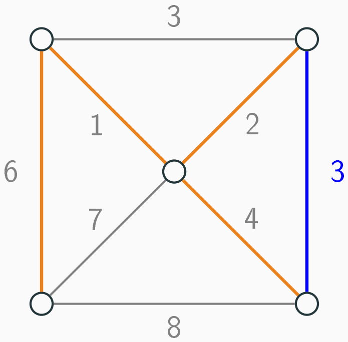


Coûts Dynamiques



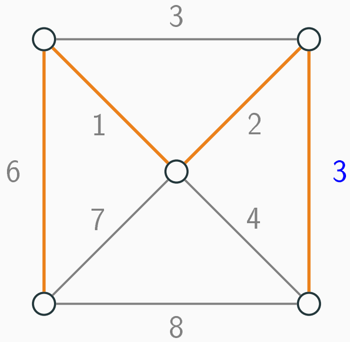
Supposons que le coût d'une arête diminue de 5 à 3. . .

Coûts Dynamiques



Supposons que le coût d'une arête diminue de 5 à 3. . .

Coûts Dynamiques



Supposons que le coût d'une arête diminue de 5 à 3...

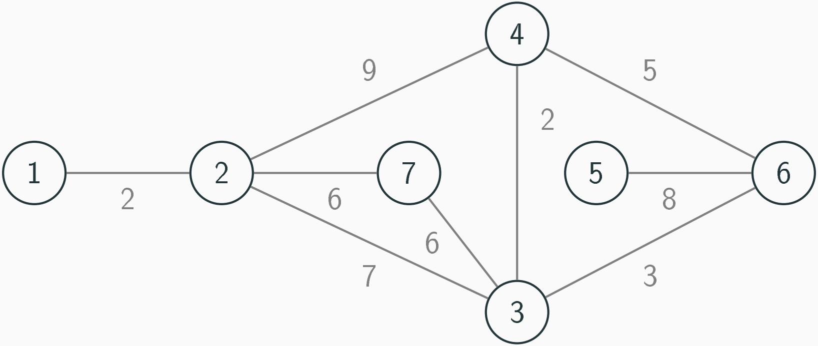
Exercice

Un réseau informatique est composé de 7 stations de travail en 7 centres différents. Certains centres sont reliés par des lignes de communication. Les équipements étant vétustes, il a été décidé de les remplacer par un matériel plus moderne. Ce remplacement peut être effectué selon les coûts suivants :

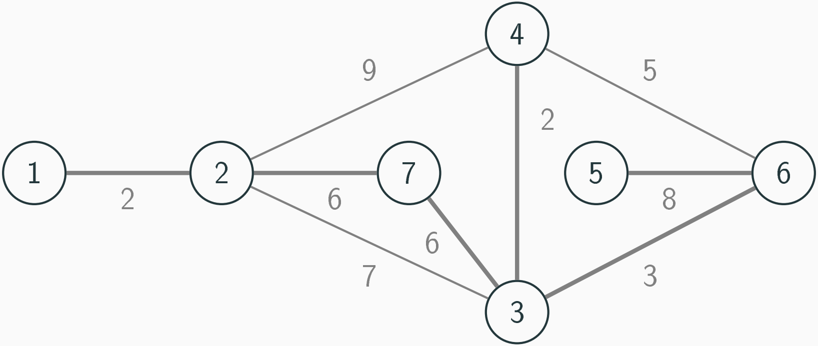
	2	3	4	5	6	7
1	2	c	c	c	c	c
2		7	9	c	c	6
3			2	c	3	6
4				c	5	c
5					8	c
6						c

La valeur c veut dire qu'il n'existe pas de ligne reliant directement les deux centres correspondants, mais on peut la construire pour un coût $c = 10$.

Exercice

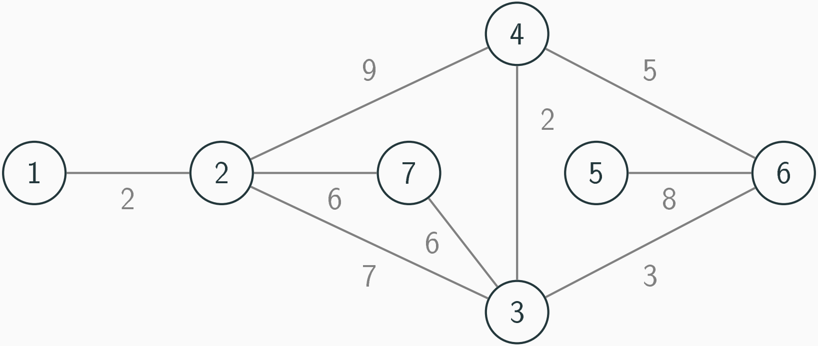


Exercice



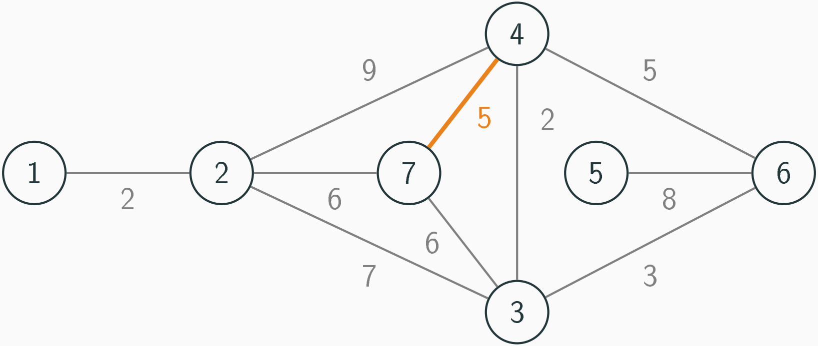
En appliquant l'algorithme de Kruskal on trouve un arbre couvrant de coût 27.

Exercice



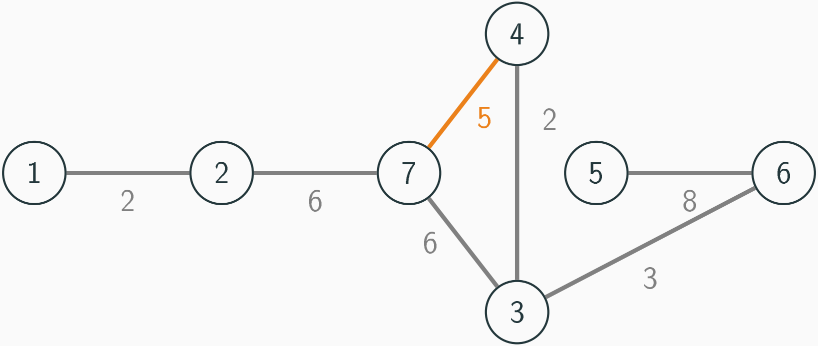
Que peut-on dire si le coût c de construction de la ligne $(4, 7)$ diminue et devient 5 ?

Exercice



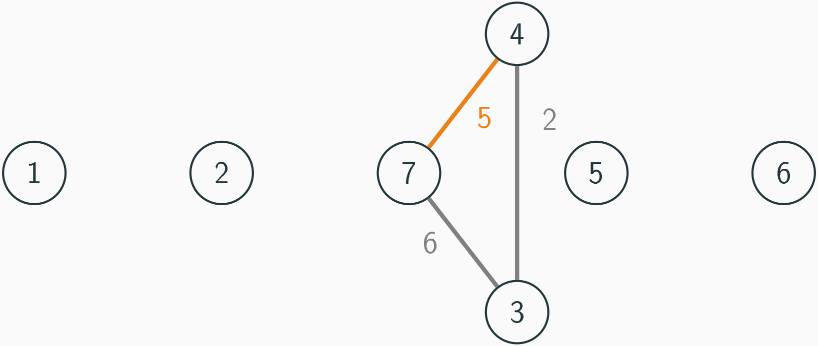
Que peut-on dire si le coût c de construction de la ligne $(4, 7)$ diminue et devient 5 ?

Exercice



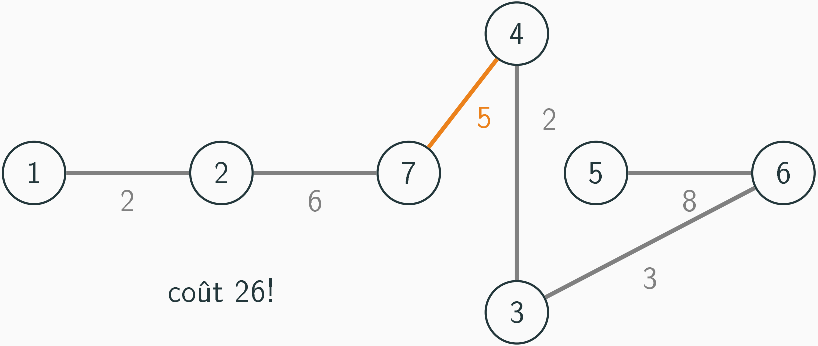
Que peut-on dire si le coût c de construction de la ligne $(4, 7)$ diminue et devient 5 ?

Exercice



Que peut-on dire si le coût c de construction de la ligne $(4, 7)$ diminue et devient 5 ?

Exercice

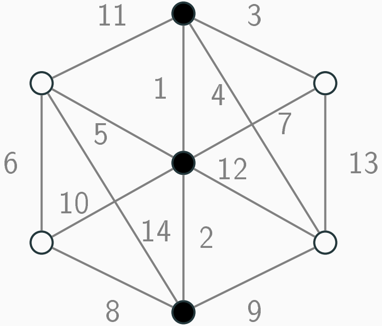


coût 26!

Que peut-on dire si le coût c de construction de la ligne $(4, 7)$ diminue et devient 5 ?

Exercice

Nous avons 4 ordinateurs (sommets blancs) et trois serveurs (sommets noirs). Le croquis présente les coûts du câblage direct entre deux postes. Chaque ordinateur doit être connecté (directement ou indirectement) à au moins un serveur. Le problème consiste à réaliser le réseau d'interconnexion au coût minimum.



Exercice (cont.)

Voici une modélisation formelle : Soit $G = (V, E)$ un graphe connexe, $\emptyset \subsetneq S \subsetneq V$ et $p : E \to \mathbb{R}_{>0}$ une fonction. On cherche un graphe partiel $G' = (V, F)$ de poids minimum, qui possède la propriété suivante :

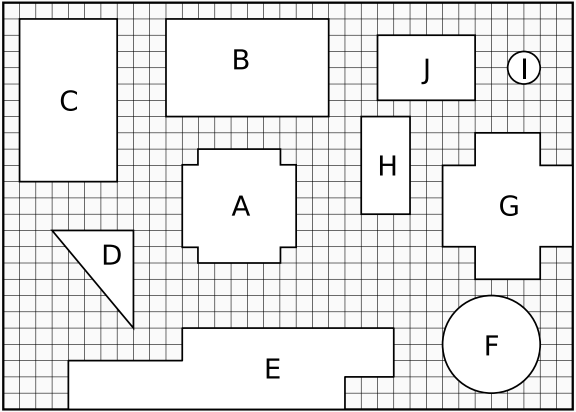
$$\forall v \in V \setminus S : \exists \text{ une } v\text{-}S \text{ chaîne dans } G'$$

Soit $G' = (V, F)$ une solution optimale.

1. Montrer que G' est une forêt.
2. Montrer que chaque composante connexe de G' contient *exactement* un sommet de S
3. Proposer un algorithme qui trouve le graphe G'.
4. Traiter l'exemple numérique et présenter la forêt de poids minimum.

Exercice

A l'aide d'une scie à découper les courbes on doit découper les 10 profils placés sur un morceau rectangulaire 35×25 de contre-plaqué comme l'indiqué le schéma ci-dessous.



Exercice (cont.)

On veut trouver le tracé qui minimise la longueur totale de découpe réellement effectuée (les passages en arrière, les déplacements répétés par une ligne ou un trou déjà découpé ne comptent pas). Pour découper un morceau à l'intérieur de la planche il faut commencer le déplacement de la scie à partir du bord de la planche.

1. Modéliser le problème général comme un modèle classique de la théorie des graphes et justifier cette modélisation.
2. Traiter l'exemple et proposer un plan optimal de découpe. Ce plan est-il unique ? (justifier)
3. Peut-on avoir en plusieurs morceaux la partie restante (quadrillée) du contre-plaqué à l'issue d'une découpe optimale ?