# Arbre de Steiner. . .

# Un petit plan

1. Définition du problème.
2. Complexité.
3. Algorithme d'approximation.
4. Algorithme d'approximation bicritères.

# 1. Définition du problème

# Arbre de steiner.

***Problème de l'arbre de Steiner minimum (SMT)***

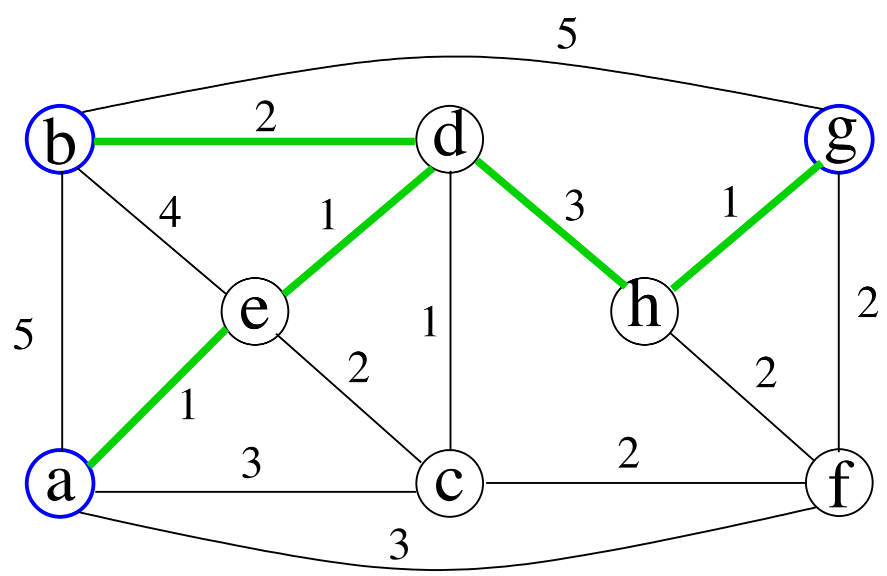
**Instance:** un graphe $G = (V, E)$, une fonction poids $c : E \to \mathbb{N}^+$, un ensemble de sommets $N \subseteq V$

**Question:** Quel est arbre couvrant $N$ de poids minimum ?

$N$ : sommets bleus.

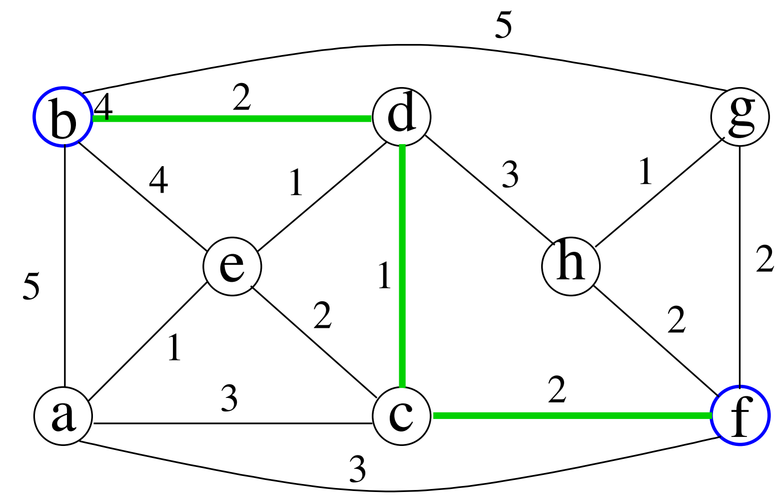
Arbre : arêtes vertes.

cout= 8 .

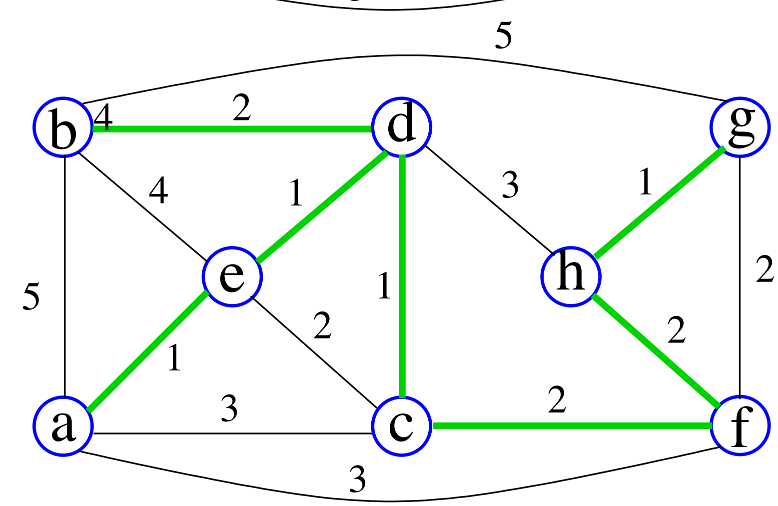


# Petites remarques.

1. Cas où $|N| = 2$,
   Problème SMT = problème du + court chemin.




2. Cas où $N = V$,
   Problème SMT = problème du arbre couvrant de poids minimum,



# Complexité du problème

***Problème de l'arbre de Steiner minimum (SMT)***

**Instance:** un graphe $G = (V, E)$, une fonction poids $c : E \to \mathbb{N}^+$, un ensemble de sommets $N \subseteq V$

**Question:** Quel est arbre couvrant $N$ de poids minimum ?

– Problème polynômial pour les cas où $|N| = 2$, $N = V$.

– Problème NP-complet, pour le cas général [Karp72]

# Rappel : Méthodologie pour prouver un problème NP-complet

1. Transformer le problème en un problème de décision $\mathcal{Z}$
2. Prouver que le problème de décision est dans NP.
3. Prouver que le problème de décision est plus difficile qu'un problème NP-complet c'est-à-dire :
   (a) Choisir un problème NP-complet $\mathcal{C}$ (ou NP-difficile) connu.
   (b) Construire une transformation POLYNÔMIALE $f : D_{\mathcal{C}} \to D_{\mathcal{Z}}$
   (c) Prouver que $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$

# Transformer le problème SMT en un problème de décision

1. **Problème de minimisation :**

   ***Problème de l'arbre de Steiner minimum (SMT)***

   **Instance:** un graphe $G = (V, E)$, une fonction poids $c : E \rightarrow \mathbb{N}^+$, un ensemble de sommets $N \subseteq V$.

   **Question:** Quel est arbre couvrant $N$ de poids minimum ?

2. **Problème de décision :**

   ***Problème de l'arbre de Steiner minimum (SMT)***

   **Instance:** un graphe $G = (V, E)$, une fonction poids $c : E \rightarrow \mathbb{N}^+$, un ensemble de sommets $N \subseteq V$, et un entier $k$.

   **Question:** Existe-t-il un arbre couvrant $N$ tel que son poids $\leq k$ ?

# Rappel : Méthodologie pour prouver un problème NP-complet

1. Transformer le problème en un problème de décision $\mathcal{Z}$
2. Prouver que le problème de décision est dans NP.
3. Prouver que le problème de décision est plus difficile qu'un problème NP-complet c'est-à-dire :
   (a) Choisir un problème NP-complet $\mathcal{C}$ (ou NP-difficile) connu.
   (b) Construire une transformation POLYNÔMIALE $f : D_{\mathcal{C}} \rightarrow D_{\mathcal{Z}}$
   (c) Prouver que $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$

# Prouver que le problème SMT est dans NP.

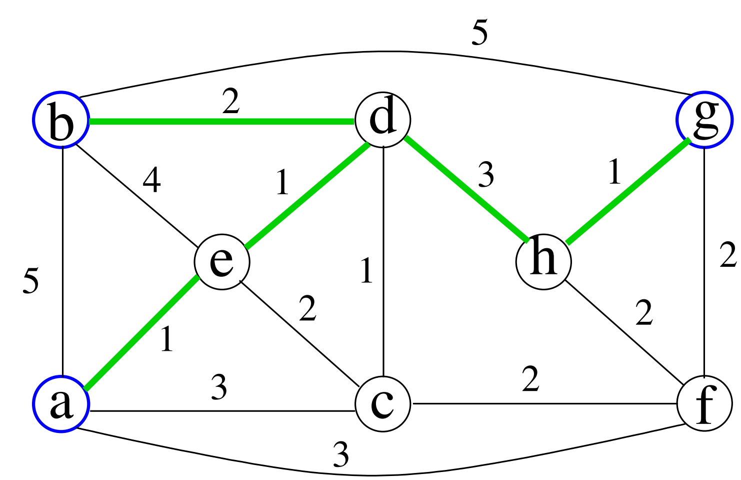
***Problème de l'arbre de Steiner minimum (SMT)***

**Instance:** un graphe $G = (V, E)$, une fonction poids $c : E \to \mathbb{N}^+$, un ensemble de sommets $N \subseteq V$, et un entier $k$.

**Question:** Existe-t-il un arbre couvrant $N$ tel que son poids $\leq k$ ?

Le problème SMT est dans NP car on peut vérifier polynomialement si un arbre donné

1. couvre l'ensemble $N$
2. a un poids $\leq k$.



## ATENTTION NNNNNNProuver que le problème SMT est dans NP.

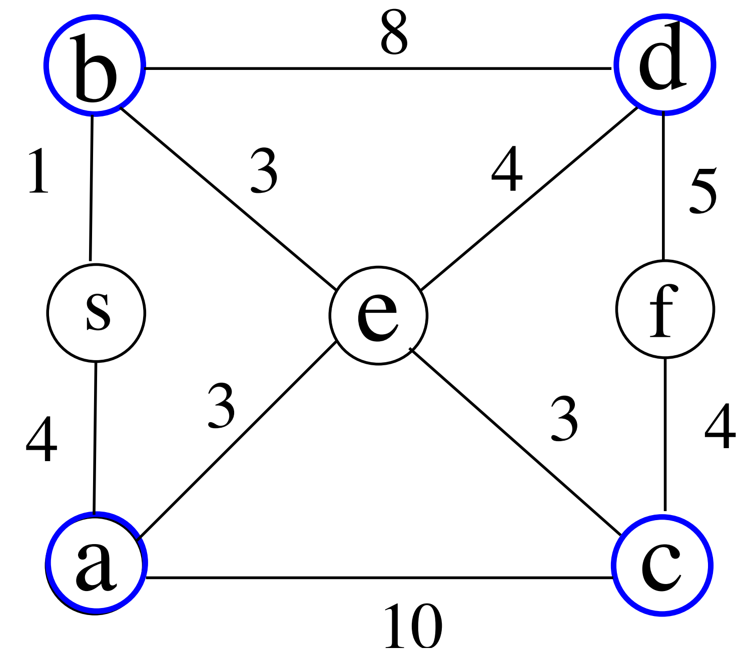
***Problème de l'arbre de Steiner minimum (SMT)***

**Instance:** un graphe $G = (V, E)$, une fonction poids $c : E \to \mathbb{N}^+$, un ensemble de sommets $N \subseteq V$, et un entier $k$.

**Question:** Existe-t-il un arbre couvrant $N$ tel que son poids $\leq k$ ?

Le problème SMT est dans NP car on peut vérifier polynomialement si un arbre donné

1. couvre l'ensemble $N$
2. a un poids $\leq k$.



# Rappel : Méthodologie pour prouver un problème NP-complet

1. Transformer le problème en un problème de décision $\mathcal{Z}$
2. Prouver que le problème de décision est dans NP.
3. Prouver que le problème de décision est plus difficile qu'un problème NP-complet c'est-à-dire :
   (a) Choisir un problème NP-complet $\mathcal{C}$ (ou NP-difficile) connu.
   (b) Construire une transformation POLYNÔMIALE $f : D_{\mathcal{C}} \rightarrow D_{\mathcal{Z}}$
   (c) Prouver que $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$

# Prouver que le problème SMT est plus difficile qu'un problème NP-complet

1. **Choisir un problème connu NP-complet $3DM$.**

***Problème du couplage à trois dimensions (3DM)***

**Instance:** Trois ensembles de même cardinalité $X$, $Y$, $Z$, un sous ensemble $M \subseteq X \times Y \times Z$

**Question:** Existe-t-il un couplage $M' \subset M$ dans $M$ ? (c'est-à-dire un élément de $X$, $Y$, $Z$ apparaît exactement une seule fois dans $M'$)

# Problème du couplage à 1 dimension (1DM)

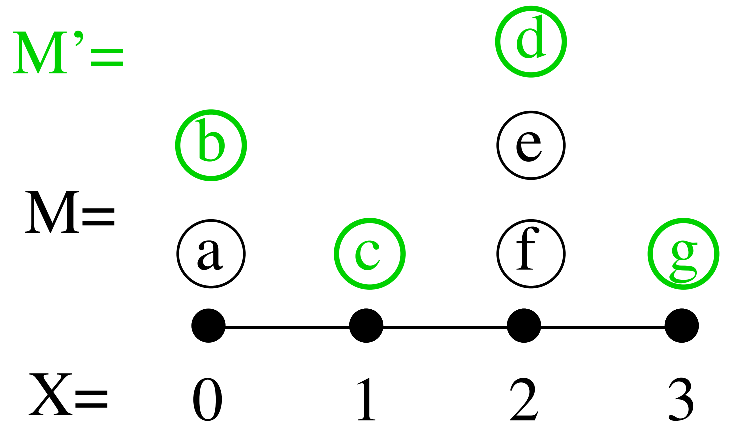
***Problème du couplage à* 1 *dimension***

**Instance:** Un ensemble $X$ et un sous ensemble $M \subseteq X$

**Question:** Existe-t-il un couplage $M' \subset M$ dans $M$ ?

**Algorithme**

- Trier les éléments de $M$ en fonction de $X$.
- Construire $M'$ en parcourant $X$.

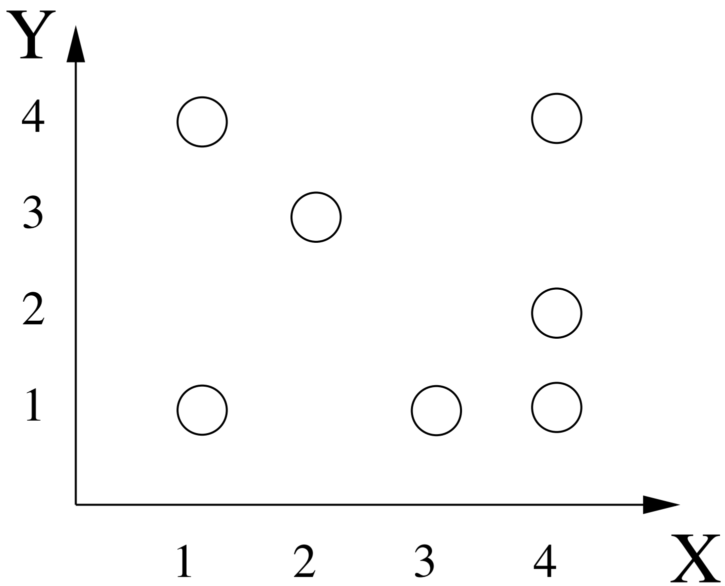


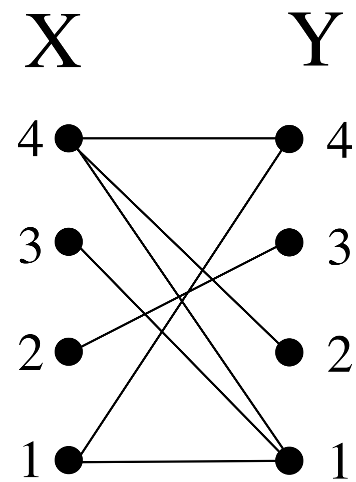
# Problème du couplage à 2 dimension (2DM)

***Problème du couplage à 2 dimension***

**Instance:** 2 ensembles de même cardinalité $X$, $Y$ et $M \subseteq X \times Y$

**Question:** Existe-t-il un couplage $M' \subset M$ dans $M$ ?




Problème équivalent au problème de couplage dans un graphe biparti.

# Prouver que le problème SMT est plus difficile qu'un problème NP-complet

1. **Choisir un problème connu NP-complet $3DM$.**

***Problème du couplage à trois dimensions (3DM)***

**Instance:** Trois ensembles de même cardinalité $X$, $Y$, $Z$, un sous ensemble $M \subseteq X \times Y \times Z$

**Question:** Existe-t-il un couplage $M' \subset M$ dans $M$ ? (c'est-à-dire un élément de $X$, $Y$, $Z$ apparaît exactement une seule fois dans $M'$)

## Prouver que le problème SMT est NP-difficile

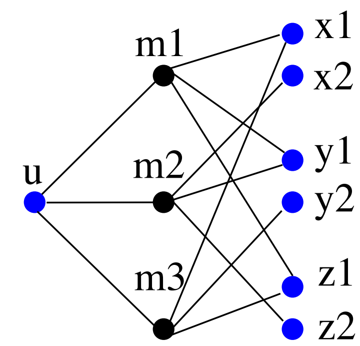
2. Construction une transformation polynômiale $f : D_{3DM} \to D_{\mathcal{Z}}$

$$f : (X, Y, Z, M) \to (G = (V, E), c : E \to \mathbb{N}^+, N \subset V, k)$$

### Algorithme

1. $V := \{u\} \cup X \cup Y \cup Z \cup M$
2. $N = \{u\} \cup X \cup Y \cup Z$
3. $k = 3|X|^5 + |X|$
4. $\forall m = (x, y, z) \in M$,
   - $(m, x) \in E$, $c(m, x) = |X|^2$,
   - $(m, y) \in E$, $c(m, y) = |X|^2$,
   - $(m, z) \in E$, $c(m, z) = |X|^2$
   - $(m, u) \in E$, $c(m, u) = 1$

$X = \{x1, x2\}$ $Y = \{y1, y2\}$
$Z = \{z1, z2\}$
$(x(1), y(1), z(1)) \in M$
$(x(2), y(1), z(2)) \in M$ ,
$(x(1), y(2), z(1)) \in M$



# Prouver que le problème SMT est NP-difficile

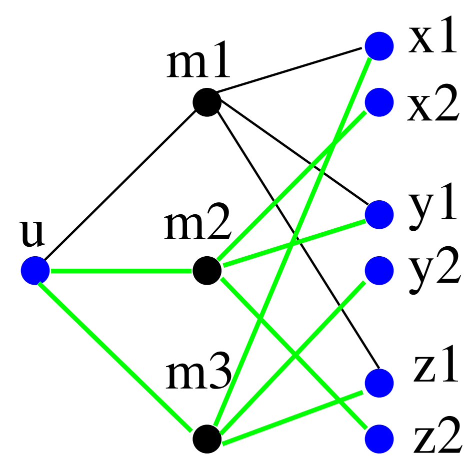
**3. Prouver que** $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$

Supposons qu'il existe un couplage $M'$ dans $M$.
Prouvons qu'il existe un arbre couvrant $T$ de $N$ dans $G$ de poids $\leq k$ :

$$\forall m = (x, y, z) \in M', (m, x),\ (m, y),\ (m, z), (m, u) \in\in T$$

$X = \{x1, x2\}$, $Y = \{y1, y2\}$, $Z = \{z1, z2\}$
$M = \{m1, m2, m3\}$ $M' = \{m2, m3\}$

$T$ =arêtes vertes



# Prouver que le problème SMT est NP-difficile

**3. Prouver que** $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$

Supposons qu'il existe un arbre couvrant $T$ de $N$ dans $G$ de poids $\leq k$.
Prouvons qu'il existe aussi un couplage $M'$ dans $M$.

**Structure de** $T$ **:**

– $3|X|$ arêtes de poids $|X|^4$

1. au moins $3X$ arêtes de poids $|X|^4$ car il faut connecter les sommets de $X \cup Y \cup Z$.
2. au plus 3 $X$ arêtes de poids $|X|^4$ à cause de la valeur de $k(= 3|X|^5 + |X|)$

– $|X|$ arêtes de poids $1$.

1. au plus $X$ arêtes de poids $1$ à cause de la valeur de $k$
2. au moins $X$ arêtes de poids $1$ car il faut connecter avec les "bouts" d'arbres.

## Prouver que le problème SMT est NP-difficile

**3. Prouver que** $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$

Supposons qu'il existe un arbre couvrant $T$ de $N$ dans $G$ de poids $\leq k$.

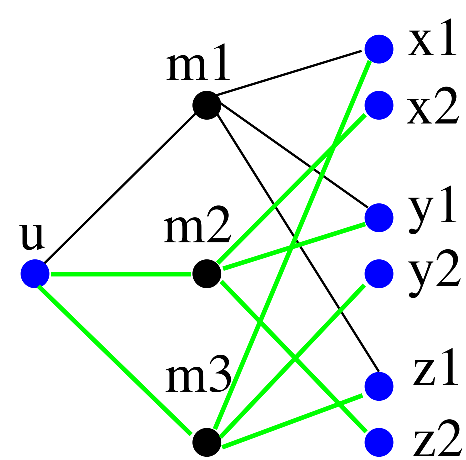
Prouvons qu'il existe aussi un couplage $M'$ dans $M$.

**Structure de $T$ :** $3|X|$ arêtes de poids $|X|^4 + |X|$ arêtes de poids $1$.

**Extraction de $M'$ de $M$ :** $m \in M'$ si et seulement si $m$ couvert par $T$

$M'$ **couplage parfait de** $M$ **:** oui.

1. $|M'| = |X|$ (# arêtes de poids $|X|^4$)
2. $\forall s \in X \cup Y \cup Z$, $s$ apparait 1 fois dans $M'$.




$M' = \{m2, m3\}$

## En résumé :

1. Transformation du problème de minmisation en un problème de décision $SMT$.
2. Le problème $SMT$ de décision est dans NP.
3. Le problème $SMT$ de décision est NP-difficile, car...
   (a) Choix d'un problème NP-complet : problème du 3DM.
   (b) Construction d'une transformation polynômiale $f : D_{3DM} \rightarrow D_{SMT}$
   (c) Prouver que $\forall I \in D_{\mathcal{C}}, I \in Y_{\mathcal{C}} \Leftrightarrow f(I) \in Y_{\mathcal{Z}}$ c'est-à-dire :

| | | |
|---|---|---|
| Pour 1 instance $I = (X, Y, Z, M)$,<br><br>il existe un couplage $M'$ dans $M$ | $\Leftrightarrow$ | Pour l'instance $f(I) = (G, c, N, k)$,<br>il existe un arbre couvrant $T$ de $N$ dans $G$ de poids $\leq k$ |

## En conclusion :

**Théorème 1.** *Le* **problème de l'arbre de Steiner minimum (SMT)**

**Instance:** *un graphe* $G = (V, E)$*, une fonction poids* $c : E \to \mathbb{N}^+$*, un ensemble de sommets* $N \subseteq V$*, et un entier* $k$*.*

**Question:** *Existe-t-il un arbre couvrant* $N$ *tel que son poids* $\leq k$ *?*

*est NP-complet.*

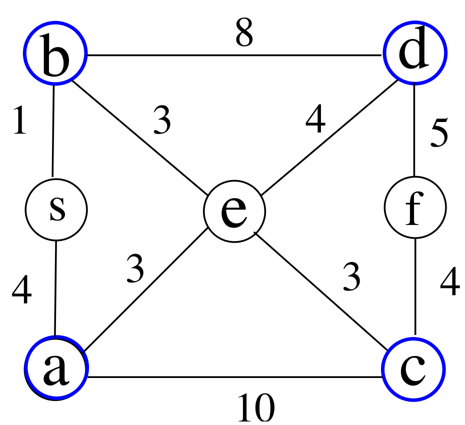
**Remarque :** A-t-on prouvé si le problème est NP-complet si les poids des arêtes sont identiques ?

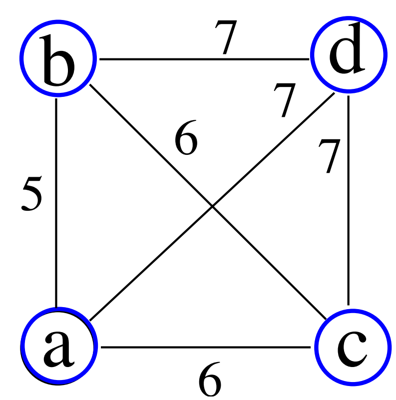
## Algorithme d'approximation du problème SMT

**Entrée :** un graphe $G = (V, E)$ et sous ensemble $N$

**Sortie :** un arbre $T$ couvrant de $N$

1. Extraire $K$, graphe complet : poids= distance entre les sommets de $M$.



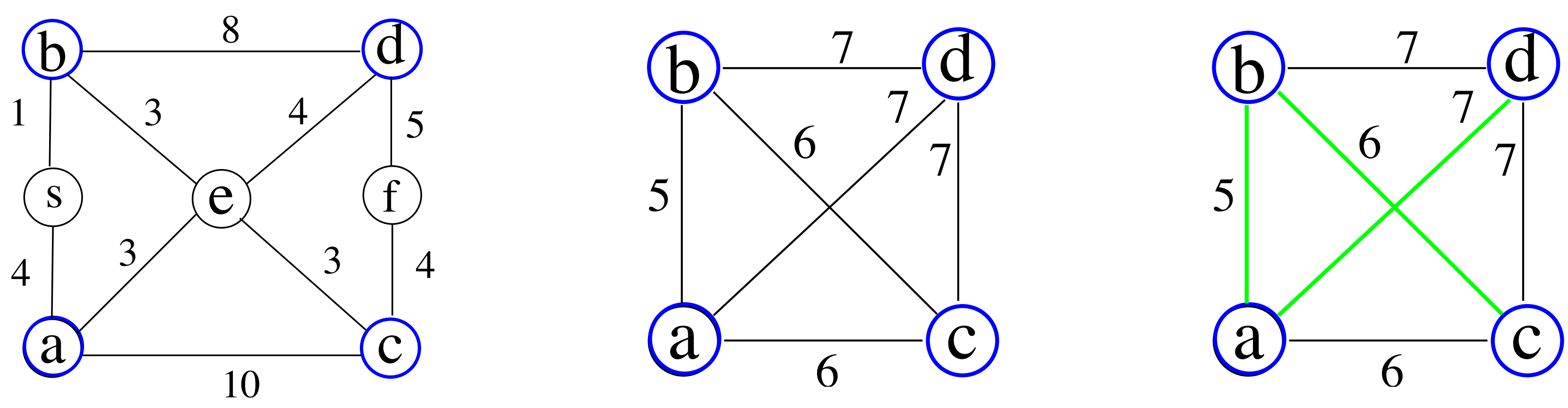



## Algorithme d'approximation du problème SMT

**Entrée :** un graphe $G = (V, E)$ et sous ensemble $N$

**Sortie :** un arbre $T$ couvrant de $N$

1. Extraire $K$, graphe complet : poids= distance entre les sommets de $M$.
2. Calculer $TC$, l'arbre couvrant de poids minimal de $K$

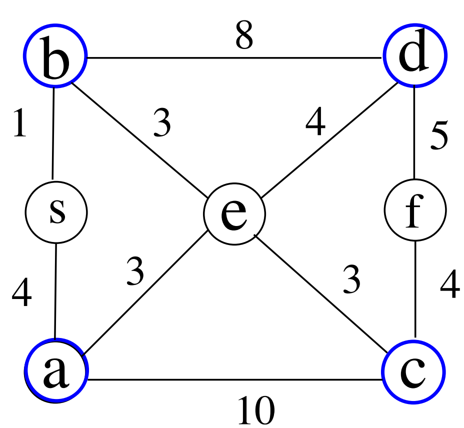


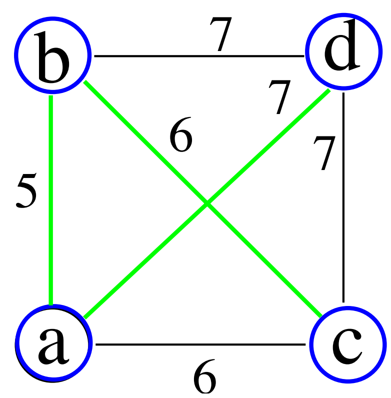
# Algorithme d'approximation du problème SMT

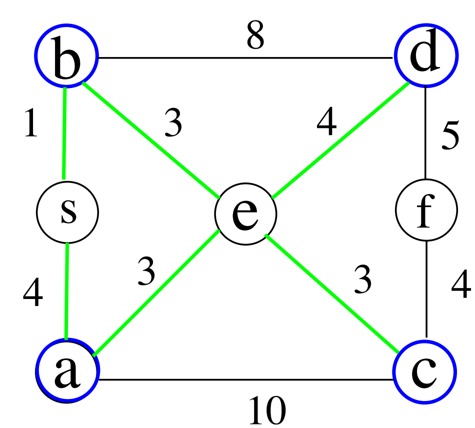
**Entrée :** un graphe $G = (V, E)$ et sous ensemble $N$

**Sortie :** un arbre $T$ couvrant de $N$

1. Extraire $K$, graphe complet : poids= distance entre les sommets de $M$.
2. Calculer $TC$, l'arbre couvrant de poids minimal de $K$
3. Déduire $G_3$, un sous graphe de $G$, où chaque arête de $TC$=un + court chemin.



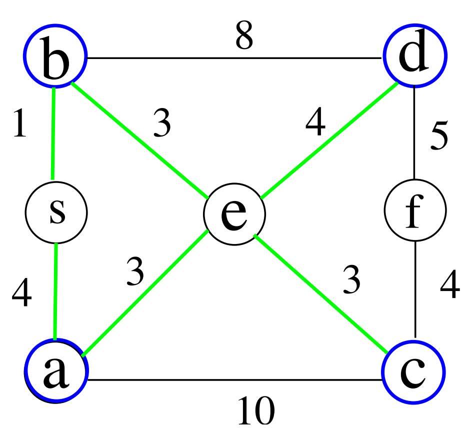



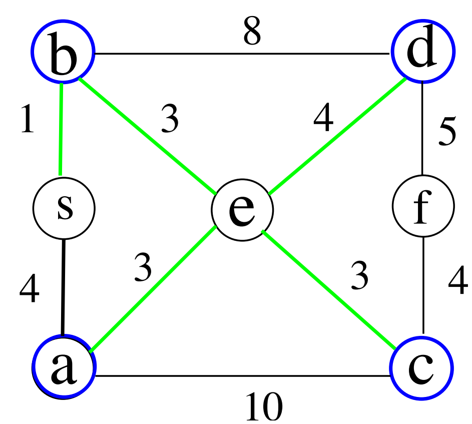
## Algorithme d'approximation du problème SMT

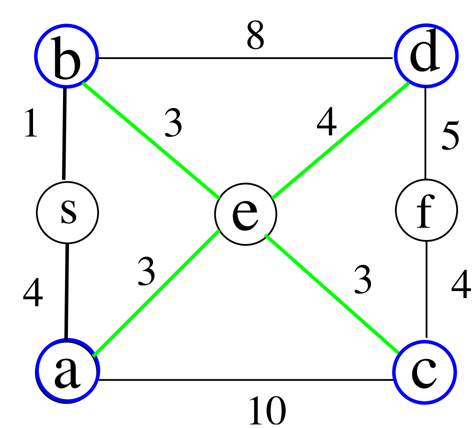
**Entrée :** un graphe $G = (V, E)$ et sous ensemble $N$

**Sortie :** un arbre $T$ couvrant de $N$

3. $G_3$, un sous graphe de $G$, où chaque arête de $TC$= un + court chemin.
4. Calculer $G_4$, l'arbre couvrant de poids minimal de $G_3$
5. Calculer $T$, suppression d'arêtes inutiles.



# Algorithme d'approximation du problème SMT

**Entrée :** un graphe $G = (V, E)$ et sous ensemble $N$

**Sortie :** un arbre $T$ couvrant de $N$

1. Extraire $K$, le sous-graphe des plus courts chemins dans $M$.
2. Calculer $TC$, l'arbre couvrant de poids minimal de $K$ (Prim/Kruskal)
3. Déduire $G_3$, un sous graphe de $G$, où chaque arête de $TC$ est représentée par un plus court chemin.
4. Calculer $G_4$, l'arbre couvrant de poids minimal de $G_3$
5. Calculer $T$, suppression d'arêtes inutiles.

## Rapport d'approximation

**Théorème 2.** *Le rapport d'approximation* $R = \frac{cout(T)}{cout(T^*)} \leq 2 - \frac{2}{|N|}$ *où* $T^*$ *est la solution optimale. [Takahashi]*

**Preuve :** Soient $S_1, S_2, \ldots, S_{|N|}$ les sommets de $N$ dans l'ordre correspondant à un parcours en profondeur de $T^*$

- $d_K(S_1, S_2) \leq d_{T^*}(S_1, S_2)$, $d_K(S_2, S_3) \leq d_{T^*}(S_2, S_3)$,. . ..
- Soit $A_1$ arbre couvrant dans $K$

  $A_1 = \{(S_1, S_2), \ldots, (S_{|N|-1}, S_{|N|})\}$

  $cout(A_1) = \sum_{i=1}^{|N|-1} d_K(S_i, S_{i+1}) \leq \sum_{i=1}^{|N|-1} d_{T^*}(S_i, S_{i+1})$

  $cout(A_1) \leq 2cout(T^*) - d_{T^*}(S_{|M|}, S_1)$
- Soit $A_2$ arbre couvrant dans $K$

  $A_2 = \{(S_2, S_3), \ldots, (S_{|N|}, S_1)\}$

  $cout(A_2) \leq 2cout(T^*) - d_{T^*}(S_1, S_2)$

## Suite de la preuve

- $A_3 = \{(S_3, S_4), \ldots, (S_{|N|}, S_1), (S_1, S_2)\}$
- . . ..
- $A_{|N|} = \{(S_{|N|}, S_1), \ldots, (S_{|N|-3}, S_{|N|-2}), (S_{|N|-2}, S_{|N|-1})\}$

Donc

$$\sum_{i=1}^{|N|} cout(A_i) \leq 2|N|cout(T^*) - \sum_{i=1}^{|N|} d_{T^*}(S_i, S_{i+1})$$

$$\sum_{i=1}^{|N|} cout(A_i) \leq 2|N|cout(T^*) - 2cout(T^*)$$

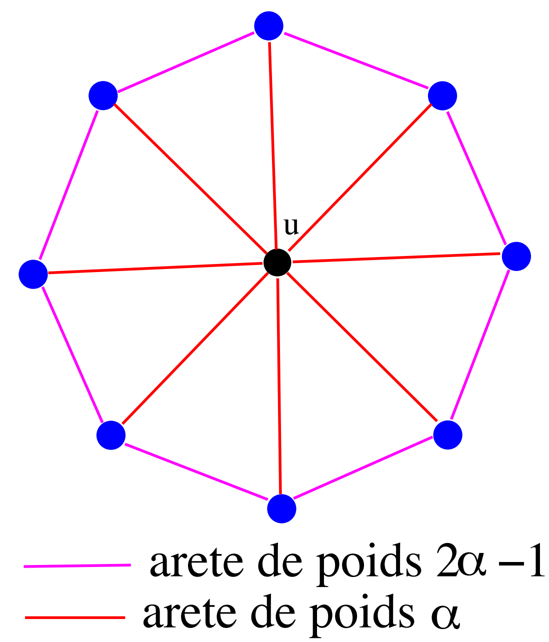
- Par définition, on a $\forall i, cout(TC) \leq cout(A_i)$.
- $|N| \times cout(TC) \leq 2|N|cout(T^*) - 2cout(T^*)$
- $cout(T) \leq cout(TC) \leq 2 - \frac{2}{|N|}$

# Instances atteignant le rapport d'approx : la Roue

**1. La roue de $n$ sommets :**

– le poids des aretes du cycle $= 2\alpha - 1$

– le poids des aretes radiales $= \alpha$



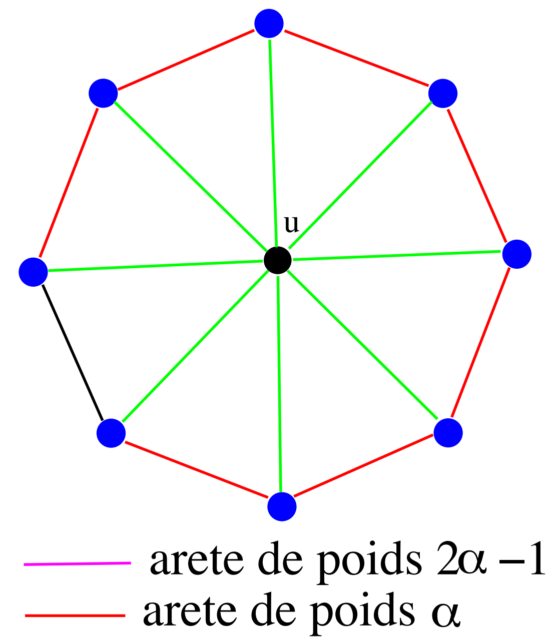

**2.** $N$ **:** les sommets du cycle

## Instances atteignant le rapport d'approx : la Roue

**Instance**
La roue de $n$ sommets :
$N$ : les sommets du cycle

- L'arbre de Steiner optimal :
  - étoile centrée en $u$
  - cout $= \alpha(n-1)$
- L'arbre de l'algo :
  - cycle - une arete
  - cout $= (2\alpha - 1) \times (n-1)$

- Rapport : $\lim_{\alpha \to} R = 2 - \frac{2}{|N|}$



# Fin du cours sur les arbres de Steiner