M2 MPRO

Bases de l'Optimisation dans les Graphes : couplages, flots et liens avec la programmation linéaire

Programmation linéaire
et problèmes d'optimisation dans les graphes

Problèmes d'optimisation dans les graphes : quelles méthodes pour les résoudre ?

- Théorie des graphes (et algorithmique de graphes) :
  - **La théorie des graphes fournit un cadre unifié pour étudier des propriétés structurelles des graphes** (bornes sur la valeur d'un paramètre), mais aussi pour résoudre certains problèmes d'optimisation dans les graphes
  - De nombreux algorithmes combinatoires issus de l'algorithmique de graphes sont ainsi basés sur ces propriétés structurelles
  - Exemple *naïf* : K_5 n'est pas planaire (propriété structurelle) ➔ trouver une clique de taille maximum dans un graphe planaire est une tâche « triviale »
- Un autre des grands domaines de la RO est la programmation mathématique, et en particulier linéaire (PL ou PLNE) :
  - Des modèles et des méthodes **génériques** (et souvent efficaces)
  - Peut-on exhiber **des liens entre ces deux domaines** ? S'il y en a, comment les exploiter ? Et comment traduire la spécificité d'un problème particulier de l'optimisation dans les graphes dans un modèle aussi général ?

ACTE IV :

Dualité en PL et matrices totalement unimodulaires ; Application au théorème de König

Les problèmes du couplage maximum et du transversal minimum (vertex cover) dans un graphe

- Deux problèmes emblématiques de la RO et de l'optimisation dans les graphes :
  - On se donne un graphe G non orienté
  - *Couplage* = ensemble d'arêtes de G sans sommets en commun
  - *Transversal* = ensemble de sommets de G qui contient au moins une extrémité de chaque arête de G
- Modèle PLNE pour le problème du couplage maximum ?
- Dual (avec variables entières) de la relaxation continue de ce PLNE = PLNE pour le transversal minimum
- La faible dualité en PL implique déjà que :

taille couplage maximum ≤ taille transversal minimum

Totale unimodularité

- Définition : une matrice à valeurs réelles est dite **totalement unimodulaire (TU)** si et seulement si le déterminant de toute sous-matrice carrée qui en est extraite vaut 0, 1 ou -1
- **Propriété 1 : toute sous-matrice d'une matrice TU est TU, et en particulier tout élément d'une matrice TU vaut 0, 1 ou -1 (sous-matrice carrée extraite ayant une ligne et une colonne)**
- **Propriété 2 : si une matrice est TU, alors la juxtaposition de cette matrice avec la matrice identité est aussi une matrice TU**
- **Propriété 3 : une matrice est TU si et seulement si sa transposée l'est**
- **Caractérisation de Ghouila-Houri** : une matrice est TU ssi tout sous-ensemble de colonnes peut être partitionné en deux parties telles que, sur chaque ligne, les sommes des éléments sur chacune de ces deux parties diffèrent d'au plus 1
- Corollaire de la propriété 3 : le théorème précédent est aussi vrai en échangeant « colonne » et « ligne »

Conséquence de la totale unimodularité

- Propriété 4 : si un PL a une matrice des contraintes TU et des seconds membres entiers, alors toute solution de base de ce PL est entière (c'est donc vrai, en particulier, pour toute solution de base optimale)
- **Corollaire de la Propriété 4 :**

Résoudre un PLNE ayant une matrice des contraintes TU et des seconds membres entiers est équivalent à résoudre sa relaxation continue (c'est-à-dire un PL) !

Une première condition suffisante de totale unimodularité, et une conséquence

- Propriété 5 : matrice d'incidence (sommets-arêtes) TU si graphe biparti
- **Corollaire de la Propriété 5 et de la dualité PL entre couplage maximum et transversal minimum :**

Théorème de König

- La « spécificité » du problème d'optimisation dans les graphes considéré est donc ici exprimée par la formulation PL/PLNE utilisée (et l'interprétation du dual), et les propriétés de la matrice des contraintes associée (matrice d'incidence d'un graphe biparti)
- Une autre conséquence de ce raisonnement (*pourquoi ?*) est que : ***les problèmes du couplage maximum et du transversal minimum sont polynomiaux dans les graphes bipartis*** !

ACTE V :

Matrices TU et dualité flot/coupe
(Théorème de Ford-Fulkerson) ;
Problèmes connexes et conséquences

Le problème du flot maximum

- Autre problème emblématique en RO/optim. dans les graphes
  - *Rappel* : on se donne un graphe orienté G connexe dont les arcs sont munis de capacités, un sommet **source** s et un sommet **puits** p
  - Problématique : on cherche, dans G, à acheminer autant d'unités de flot que possible de s vers p, sans dépasser la capacité de chaque arc
  - Modèle PLNE (modèle 1) ?
    - Le flot total circulant sur tous les chemins de s à p passant par chaque arc est limité par la capacité de l'arc
    - Maximiser la quantité de flot acheminée de s à p
    - Résolution en temps polynomial si flots continus ?
  - Modèle PLNE alternatif (modèle 2) ?
    - Conservation du flot en chaque nœud (sauf s et p)
    - Flot limité par la capacité sur chaque arc
    - Maximiser la quantité de flot acheminée de s à p

Calcul du dual

- On peut écrire les PL duaux des relaxations continues des 2 modèles précédents
  - Modèle 1 : et si les variables duales sont entières ?
    - On obtient un PL modélisant le problème de la coupe minimum
    - Les PL sont liés par une relation de dualité (dualité en PL), et donc les problèmes aussi !
  - Modèle 2
    - Interprétation du PL dual si les variables duales sont entières
    - Preuve alternative de la faible dualité flot/coupe !

Une deuxième condition suffisante de totale unimodularité, et une autre conséquence

- Propriété 6 : matrice d'incidence (sommets-arcs) TU si graphe orienté
- **Corollaire (immédiat !) du modèle 2 et des Propriétés 1 à 4 + 6 :**

Théorème de Ford-Fulkerson (flot max=coupe min)

- « Preuve alternative » basée exclusivement sur la dualité en PL et les propriétés des matrices TU
- La « spécificité » du problème d'optimisation dans les graphes considéré est ici encore exprimée par la formulation PL/PLNE utilisée (et l'interprétation de son dual), et les propriétés de la matrice des contraintes associée (matrice d'incidence d'un graphe orienté)
- Autre conséquence de ce raisonnement : polynomialité des 2 problèmes considérés (flot maximum et coupe minimum), grâce au modèle 2

Le théorème de König comme corollaire de celui de Ford-Fulkerson : un lien inattendu ?

- En réalité, on peut prouver le théorème de König à partir de celui de Ford-Fulkerson :
  - Transformer le graphe biparti G = ((L,R), E) :
    - En ajoutant une source s reliée par un arc à tout sommet de L,
    - En ajoutant un puits p auquel tout sommet de R est relié par un arc,
    - En orientant les arêtes de E de L vers R,
    - En fixant la capacité des tous les arcs à 1.
  - Calculer un flot maximum (lien avec couplage maximum ?) et une coupe minimum (lien avec transversal minimum ?) dans ce graphe orienté, et utiliser Ford-Fulkerson ==> **Th. König** (*pourquoi ?*).
- Fournit aussi un algorithme efficace pour calculer un couplage maximum et un transversal minimum dans un graphe biparti !

Flot maximum dans les graphes non orientés

- **Problème du flot maximum entre deux sommets s et p dans un graphe *non orienté* ?**
- La matrice des contraintes de la formulation PLNE la plus naïve n'est *pas TU* ==> ce problème est-il plus difficile ? Comment le résoudre ?
- Réduire le problème considéré à un problème ayant une matrice des contraintes TU, et en déduire ainsi une « bonne » formulation PLNE ? Oui : **on peut en fait réduire ce cas au cas des graphes orientés** !
- On en déduit donc une formulation PLNE ayant une matrice TU par un raisonnement *purement combinatoire*, qui ne porte que sur des transformations du graphe donné dans l'instance considérée
- Identique pour la coupe minimum ==> extension du théorème de F.-F.

Flot à coût minimum et totale unimodularité : conséquences et applications (1/2)

- On se donne un graphe G connexe, orienté ou non, dont les arcs/arêtes sont munis de capacités et de coûts unitaires, ainsi que deux sommets s (source) et p (puits) de G
- On considère le problème suivant : déterminer dans G un flot de valeur donnée (potentiellement maximum) entre s et p, et dont le coût total (somme sur tous les arcs du coût unitaire de l'arc fois la valeur du flot sur l'arc) est minimum
- ***Problème du flot à coût minimum*** : le modèle PL associé (« inspiré » de celui du flot maximum) a une **matrice TU** ==> polynomial, même sans l'algo. de Busacker-Gowen !

Flot à coût minimum et totale unimodularité : conséquences et applications (2/2)

- **Plus court chemin (longueurs ≥ 0)** : déterminer un plus court chemin entre deux sommets s et p dans un graphe <==> calculer un flot de valeur 1 à coût minimum
- **Affectation linéaire** : n tâches à affecter à n agents (chaque tâche affectée à un et un seul agent), chaque affectation induisant un coût fixe connu

==> minimiser la somme totale des coûts (couplage parfait biparti à coût minimum) = flot (de valeur n) à coût minimum

- En général, de telles propriétés impliquent aussi l'existence d'algorithmes combinatoires efficaces pour les problèmes considérés : Busacker-Gowen, Dijkstra... Et la méthode dite « hongroise » pour l'affectation linéaire ==> cf la suite !

ACTE VI :

Totale unimodularité et algorithmes combinatoires : les approches primales-duales

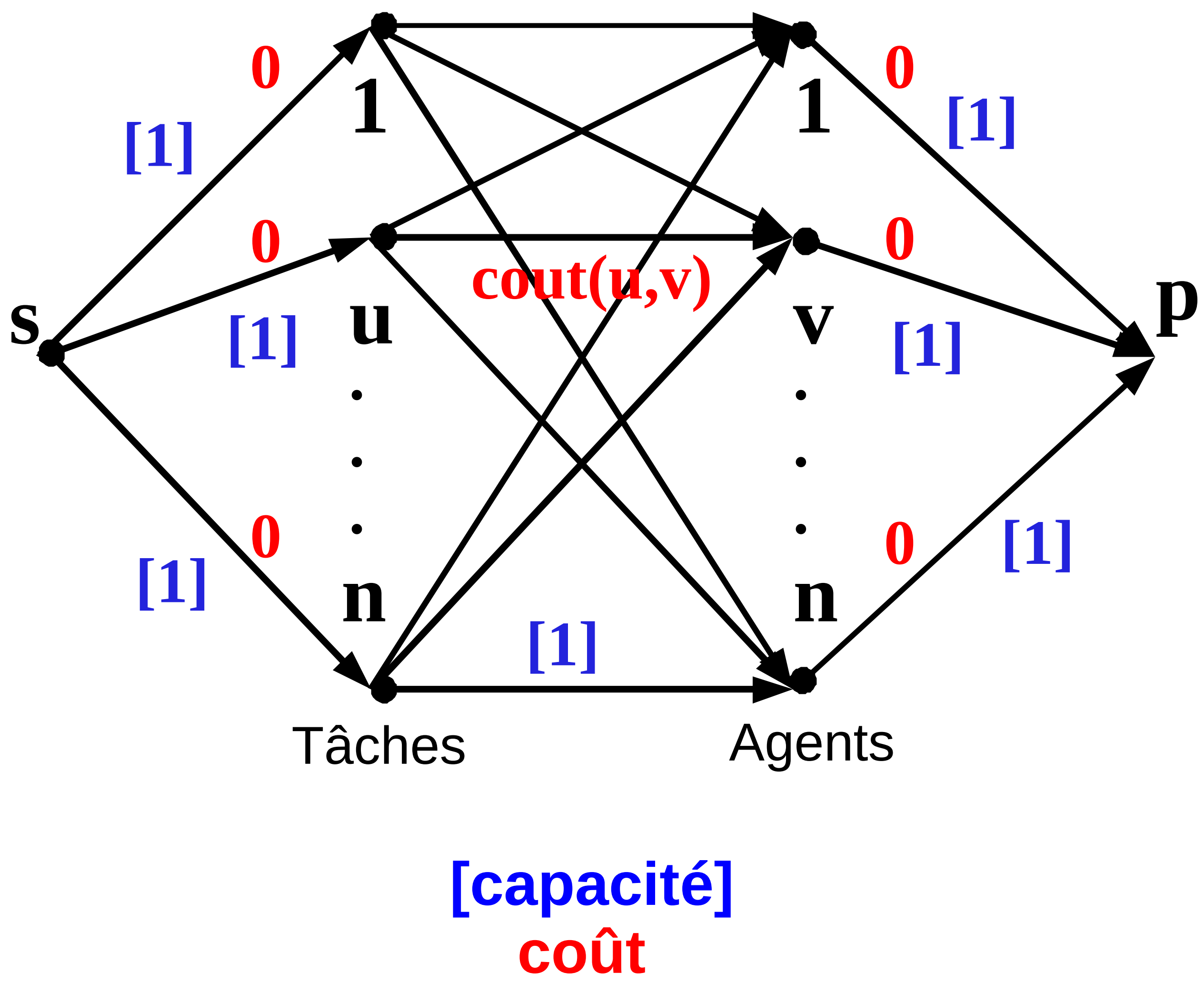
Affectation linéaire (rappels)

- Contexte : n tâches à affecter à n agents (exactement une tâche par agent)
- Affecter une tâche u donnée à un agent v donné a un coût (dit « *coût d'affectation* » ; par exemple, une durée en minutes)
- Ainsi, on notera cout(u,v) le coût (constant) d'affecter la tâche u à l'agent v
- Comment déterminer une affectation (des tâches aux agents) de coût total minimum ?

Agents	Tâche 1	Tâche 2	Tâche 3
1	10 min	15 min	40 min
2	5 min	20 min	1 heure
3	10 min	3 heures	3 heures

Affectation linéaire et couplages parfaits bipartis

- Rechercher le coût d'une affectation optimale revient en fait à rechercher le coût minimum d'un couplage parfait dans le graphe biparti complet composé de sommets de type tâches et agents (coûts = cout(u,v))
- Ceci est aussi équivalent à calculer le coût minimum d'un flot de valeur n dans un graphe orienté obtenu après ajout de s et p !



Calcul de couplages maximums à coût minimum dans les graphes bipartis (non nécessairement complets)

- On considère un graphe biparti G quelconque, dont chaque arête uv est pondérée par un coût **entier**, noté *cout(u,v)*.
- Si G n'est pas complet, alors on peut ajouter les arêtes manquantes (avec un coût *1+somme des cout(u,v)* sur chacune), et, si G n'a pas autant de sommets à gauche et à droite (par exemple, si |L| > |R|), alors on peut ajouter |L| - |R| > 0 sommets à droite, en reliant chacun d'eux à tous les sommets de L par une arête de coût 1, pour obtenir un graphe biparti **complet** G' avec **autant** de sommets à gauche et à droite.
- On a alors : calculer un couplage maximum de coût minimum dans G <==> calculer un couplage parfait de coût minimum dans G'.
- En d'autres termes, calculer un couplage maximum de coût minimum dans G <==> calculer une affectation de coût minimum en utilisant les coûts décrits précédemment, avec L = {Tâches} et R = {Agents}.

PLNE pour le calcul de couplages parfaits à coût minimum dans les graphes bipartis complets

- On s'intéresse donc dans la suite au calcul d'un couplage parfait à coût minimum dans un graphe biparti complet (avec $|L| = |R|$).
- PLNE ?
  - Variables (0-1) déjà vues : $x_{\mathbf{uv}} = 1$ ssi arête uv sélectionnée.
  - Fonction objectif : minimiser le coût total des arêtes sélectionnées.
  - Contraintes (un seul type) : cf couplage maximum, avec « = » et non « $\leq$ ».
  - Matrice des contraintes TU ! (*Pourquoi ?*)
- Dual de la relaxation continue de ce PLNE ?
  - Fonction objectif en maximisation.
  - Un seul type de variables duales (signe ?), qu'on appellera des *étiquettes*.
  - Un seul type de contraintes duales (sens ?).

Une méthode combinatoire basée sur les relations d'exclusion en PL pour calculer un couplage parfait à coût minimum dans un graphe biparti complet

- Les 2 PL précédents ont des matrices TU + des 2nds membres entiers ==> résolution en temps polynomial par la PL (*Pourquoi ?*).
- Pour faire mieux ? Méthode combinatoire (dite « *Méthode Hongroise* ») basée sur les relations d'exclusion en PL, et le calcul d'une paire de solutions admissibles (l'une pour le primal, et l'autre pour le dual) :
  - Rappel : dans une solution (de base) optimale, **contrainte primale (duale)** saturée OU variable duale (primale) associée nulle
  - De façon équivalente : dans une solution (de base) optimale, contrainte primale (duale) non saturée ==> variable duale (primale) associée nulle
  - Ici, contraintes primales saturées + un seul type de contraintes duales :
    - **Arête sélectionnée dans couplage ==> contrainte duale saturée (*)**
  - Il « suffit » donc de montrer que les 2 solutions construites :
    - Sont admissibles (pour le primal et le dual, respectivement),
    - Respectent les relations d'exclusion (cf (*)).

Aperçu de la méthode

- On construit au fur et à mesure un couplage parfait de coût minimum, en partant d'un couplage vide, et en augmentant sa taille progressivement.
- On ne s'autorise à utiliser, pour ce faire, que des arêtes pour lesquelles la contrainte duale est saturée, pour garantir la relation d'exclusion (*).
- Tant que le couplage n'est pas parfait, on tient à jour une *arborescence alternée*, qui est un arbre enraciné en un sommet (à gauche) non saturé par le couplage, et où *chaque chemin vers une feuille est alterné* :
  - Soit on peut mettre à jour la solution duale (la solution duale initiale étant choisie pour être une solution trivialement admissible), *créant ainsi* ***au moins*** *une nouvelle arête pour laquelle la contrainte duale est saturée*.
  - Soit un sommet u (à gauche) de l'arbre a au moins un voisin v (à droite) qui n'est pas dans l'arbre et tel que la contrainte duale est saturée pour l'arête uv :
    - Si v est saturé, alors on ajoute l'arête uv (orientée de u vers v) et l'arête vw du couplage saturant v (orientée de v vers w) dans l'arbre.
    - Sinon, on obtient une chaîne augmentante en ajoutant uv dans l'arbre.

Initialisation de la méthode et définitions

- Initialalement, on choisit :
  - Couplage initial (*solution primale initiale*) : couplage vide ==> *non admissible*.
  - Etiquetage initial des sommets (*solution duale initiale*) : 0 (sommet à gauche) ou minimum des coûts des arêtes incidentes (sommet à droite) ==> *admissible*.
- Tout au long de la méthode, la solution duale sera admissible (elle est clairement admissible au départ), et la relation d'exclusion sera vérifiée.
- En revanche, la solution primale M (un couplage du début à la fin) ne sera admissible qu'à la fin de la méthode (= couplage parfait).
- On note G le graphe biparti complet considéré (avec L = {sommets à gauche} et R = {sommets à droite}), $G^{=}$ le sous-graphe de G qui ne contient que les arêtes pour lesquelles la contrainte duale est saturée, T l'arbre enraciné utilisé par la méthode, et S(T) = {sommets de T} :
  - Toute arête de T non saturée par M est orientée de $S(T)\cap L$ vers $S(T)\cap R$,
  - Toute arête de T saturée par M est orientée de $S(T)\cap R$ vers $S(T)\cap L$.
- On testera si $S(T)\cap R$ et le voisinage de $S(T)\cap L$ dans $G^{=}$ sont égaux ou non.

Mise à jour de la solution duale

- Si $S(T) \cap R$ est égal au voisinage de $S(T) \cap L$ dans $G^=$:
  - On met à jour la solution duale y ==> nouvelle solution duale admissible y'.
  - Cela crée (au moins) une nouvelle arête uv dans $G^=$ telle que u est dans $S(T) \cap L$ et v est dans $R \backslash (S(T) \cap R)$ ==> après mise à jour, on n'est plus dans ce cas !
- Détail de la mise à jour ?
  - On note $q_{min} = \min_{u \in S(T) \cap L,\, v \in R \backslash (S(T) \cap R)} (cout(u,v) - y_u - y_v)$, où on rappelle que, pour tout sommet w, y_w est la valeur de la solution duale en w.
  - On pose alors $y'_w = y_w + q_{min}$ si w est dans $S(T) \cap L$, $y'_w = y_w - q_{min}$ si w est dans $S(T) \cap R$, et $y'_w = y_w$ sinon.
  - Par définition, la solution duale y' ainsi obtenue est bien admissible.
  - De même, par construction, au moins une nouvelle arête uv avec u dans $S(T) \cap L$ et v dans $R \backslash (S(T) \cap R)$ vérifie $y'_u + y'_v = cout(u,v)$ (alors que $y_u + y_v < cout(u,v)$)
- On observe aussi que toute arête uv dans $G^=$ telle que u est dans $S(T) \cap L$ et v est dans $S(T) \cap R$ vérifie $y'_u + y'_v = (y_u + q_{min}) + (y_v - q_{min}) = cout(u,v)$!

Mise à jour de l'arbre enraciné ou du couplage

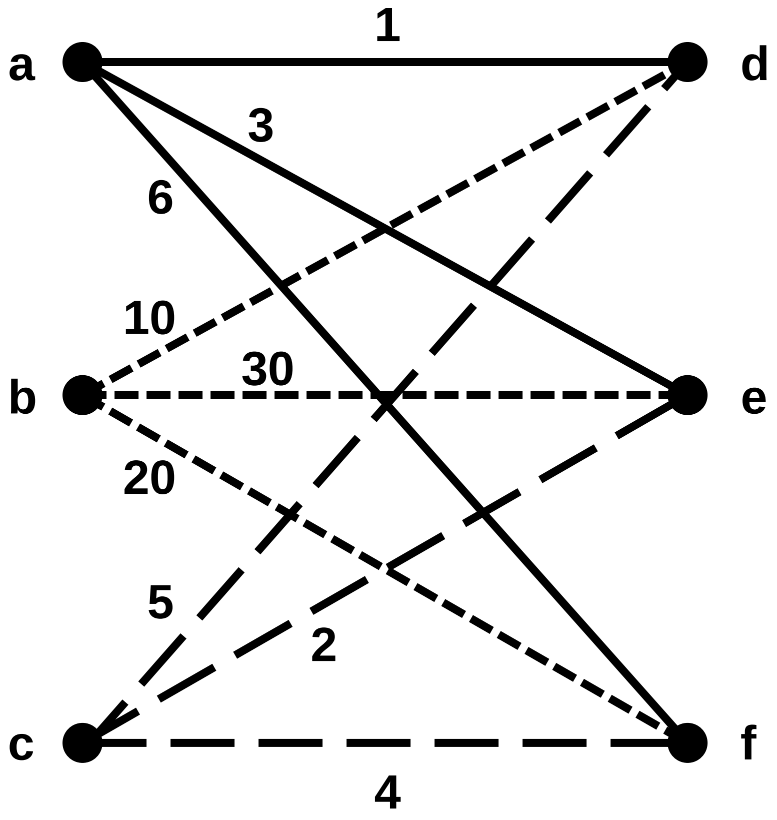
- Si S(T)∩R n'est pas égal au voisinage de S(T)∩L dans $G^=$, alors il existe une arête uv dans $G^=$ telle que u est dans S(T)∩L et v est dans R\(S(T)∩R) :
  - Si v est saturé par le couplage M, alors on note vw l'arête de M incidente à v : on ajoute le sommet v à S(T)∩R et le sommet w à S(T)∩L, et les arêtes uv et vw à l'arbre enraciné T (dans le sens u vers v et v vers w, respectivement).
  - Sinon, le chemin (alterné) dans l'arbre enraciné T entre la racine de cet arbre et le sommet u suivi de l'arête uv est une *chaîne augmentante* pour M ==> on met à jour le couplage M, et on augmente ainsi sa taille de 1. (T devient vide !)
- Propriétés de cette mise à jour (de l'arbre ou de la solution primale) ?
  - Toute arête uv du couplage M vérifie : soit u est dans S(T)∩L et v est dans S(T)∩R, soit u est dans L\(S(T)∩L) et v est dans R\(S(T)∩R).
  - A part la racine de l'arbre enraciné T, tout sommet de S(T) est saturé par M.
  - Si v n'est pas saturé par M, on obtient bien une chaîne augmentante, car le chemin dans l'arbre enraciné T entre la racine r de cet arbre et le sommet u est bien une chaîne alternée (par récurrence sur la distance à r ; c'est trivialement vrai pour r, et cela implique aussi en particulier que *u est différent de w* !).

Description synthétique de la méthode

- Soit M := couplage vide, et y := solution duale admissible.
- Tant que M n'est pas un couplage parfait faire
  - Choisir un sommet r de L non saturé par M, et poser S(T) = {r}.
  - Tant que |M| n'a pas augmenté faire
    - Si $S(T) \cap R$ est égal au voisinage de $S(T) \cap L$ dans $G^=$, alors mettre à jour la solution duale (*ainsi*, $S(T) \cap R \neq$ *voisinage de* $S(T) \cap L$ *dans* $G^=$).
    - Soit uv une arête dans $G^=$ avec u dans $S(T) \cap L$ et v dans $R \setminus (S(T) \cap R)$.
    - Si v n'est pas saturé par M, alors il y a une chaîne augmentante dans T+{uv} ==> on met à jour M, et |M| augmente donc de 1.
    - Sinon, v est incident à une arête vw de M ==> ajouter uv et vw dans T, et ajouter v et w dans S(T).

Un exemple d'exécution de la Méthode Hongroise

- $|L| = |R| = 3$
- Sur chaque arête uv on fait apparaître son coût (= cout(u,v))
- Arêtes incidentes à a en traits pleins
- Arêtes incidentes à b en pointillés
- Arêtes incidentes à c en pointillés « longs »

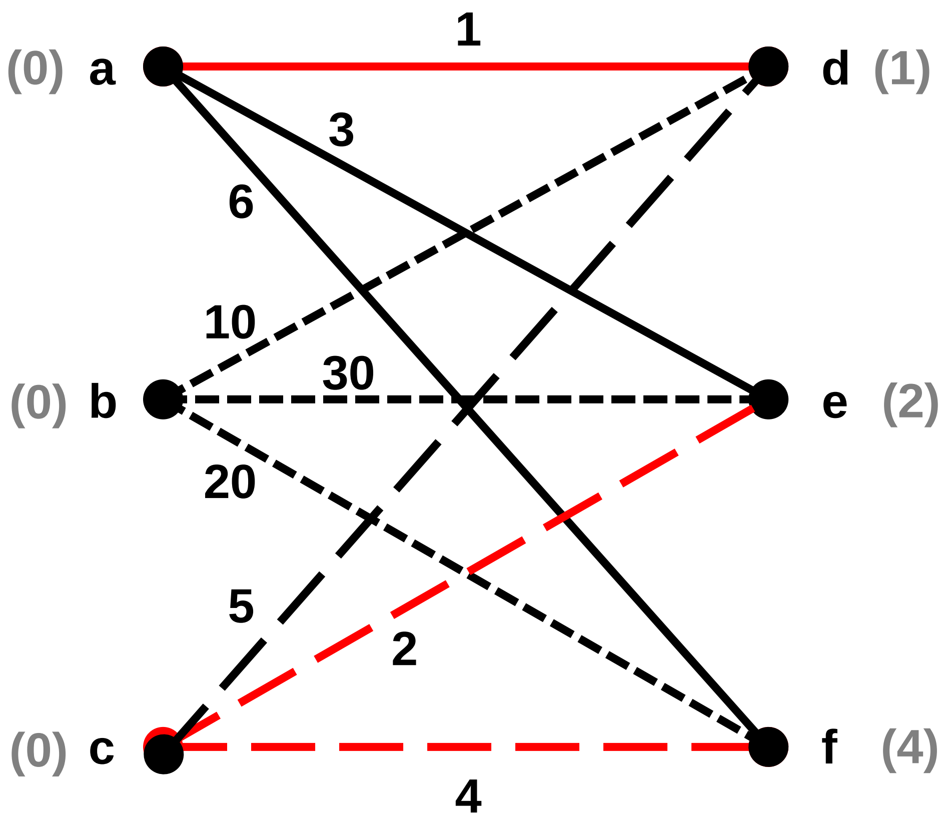


Un exemple d'exécution de la Méthode Hongroise

- Solution duale initiale (entre parenthèses et en gris) : 0 si sommet à gauche, et coût minimum des arêtes incidentes si sommet à droite

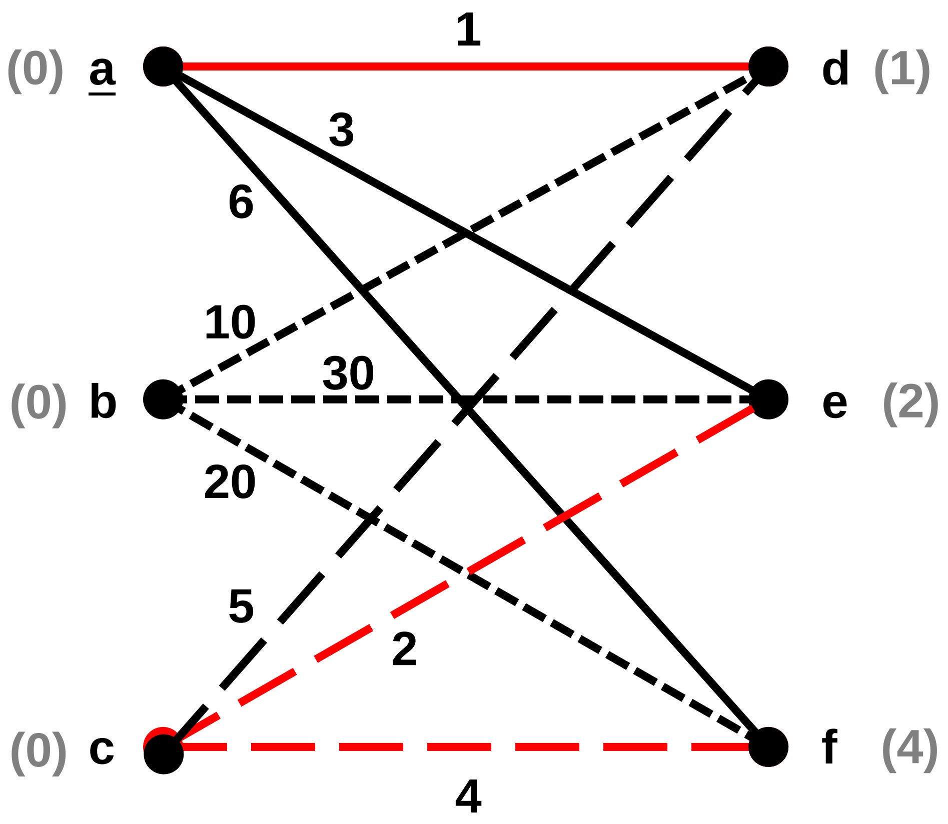
- Dans cette solution (qui est admissible), la contrainte duale est saturée pour certaines arêtes (en rouge)

- Couplage M vide : |M| = 0



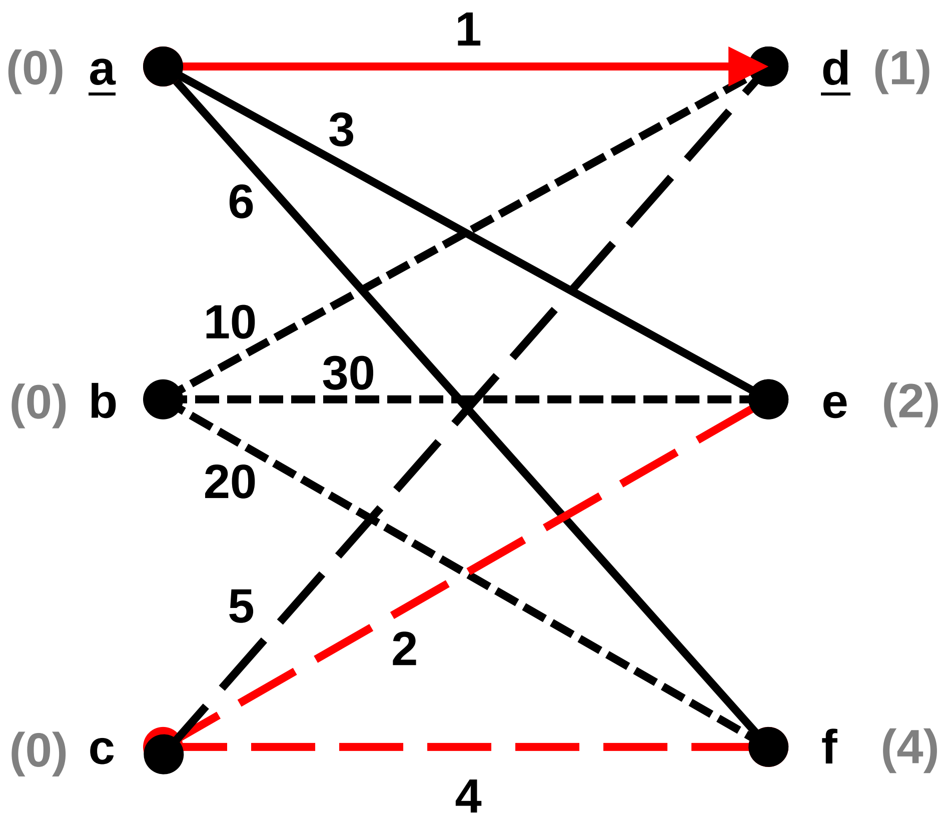
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arêtes de $G^=$ en rouge
- Initialisation de l'arborescence alternée T : S(T) = {a}
- La racine de T est donc a
- Sommets soulignés = S(T)



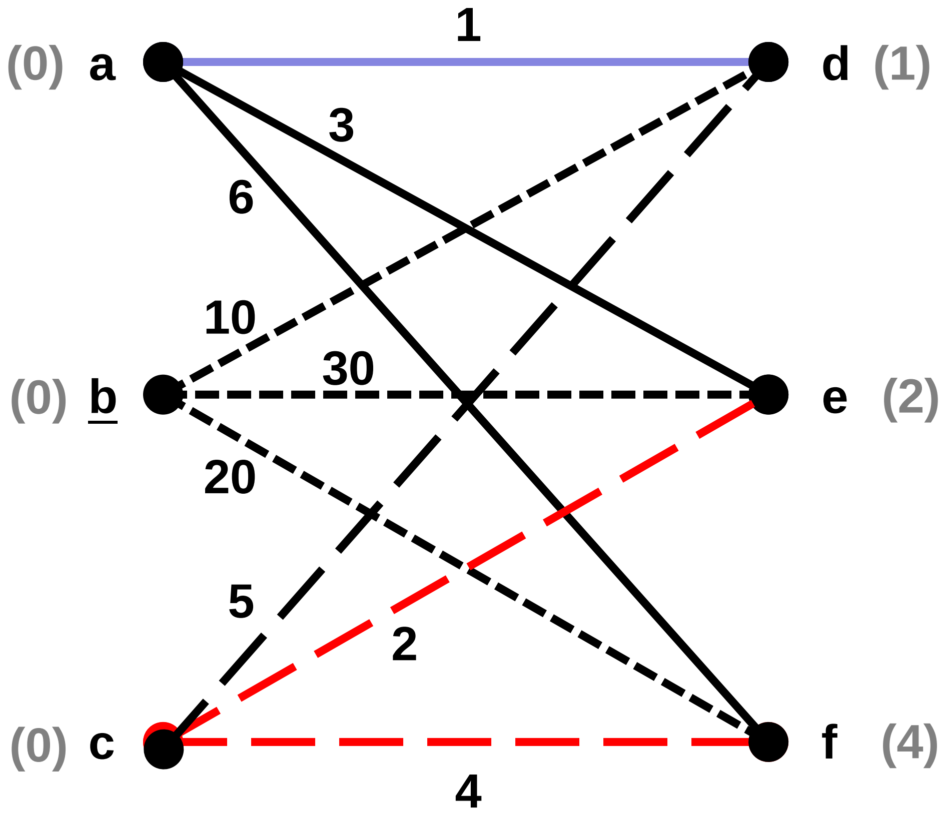
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arête ad = arête dans $G^=$ (rouge) avec a dans $S(T)\cap L$ et d dans $R\backslash(S(T)\cap R)$
- Sommet d ajouté à S(T) (sommets soulignés = S(T))
- d non saturé par M (vide) ==> ad est une chaîne augmentante pour M



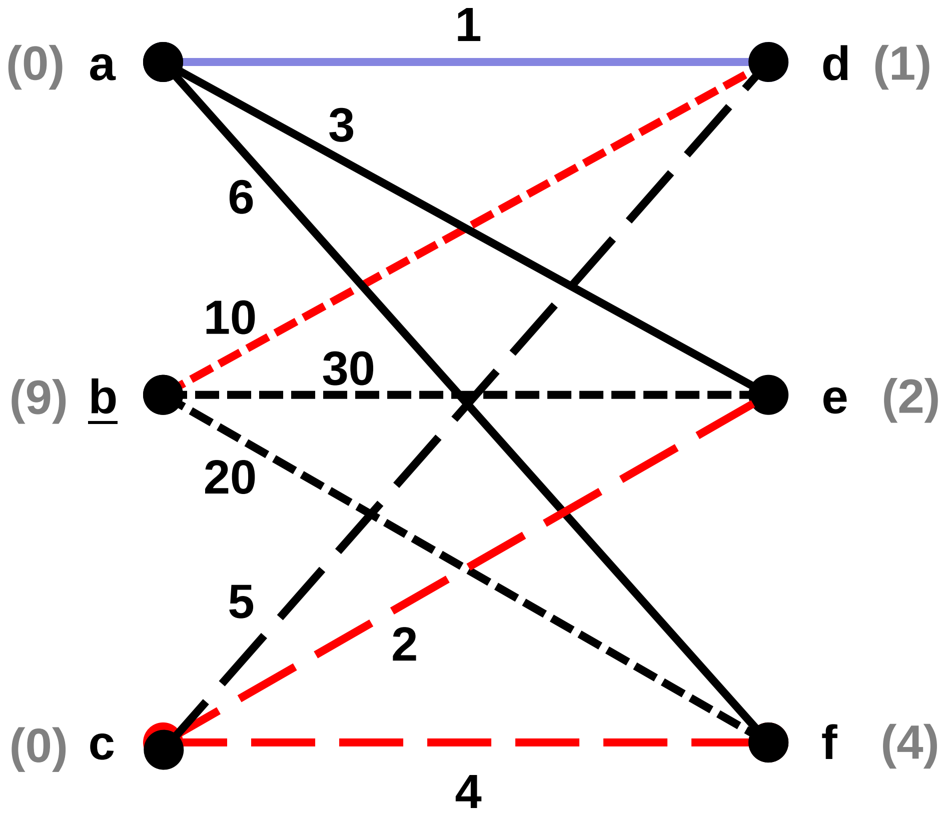
Un exemple d'exécution de la Méthode Hongroise

- M mis à jour par ajout de l'arête ad (en bleu) : |M| = 1
- Solution duale en gris
- Arêtes de $G^=$ en rouge
- Réinitialisation de l'arbre enraciné T : S(T) = {b}
- La racine de T est donc b
- Sommets soulignés = S(T)
- Voisinage dans $G^=$ de $S(T) \cap L = S(T) \cap R$ (vide)



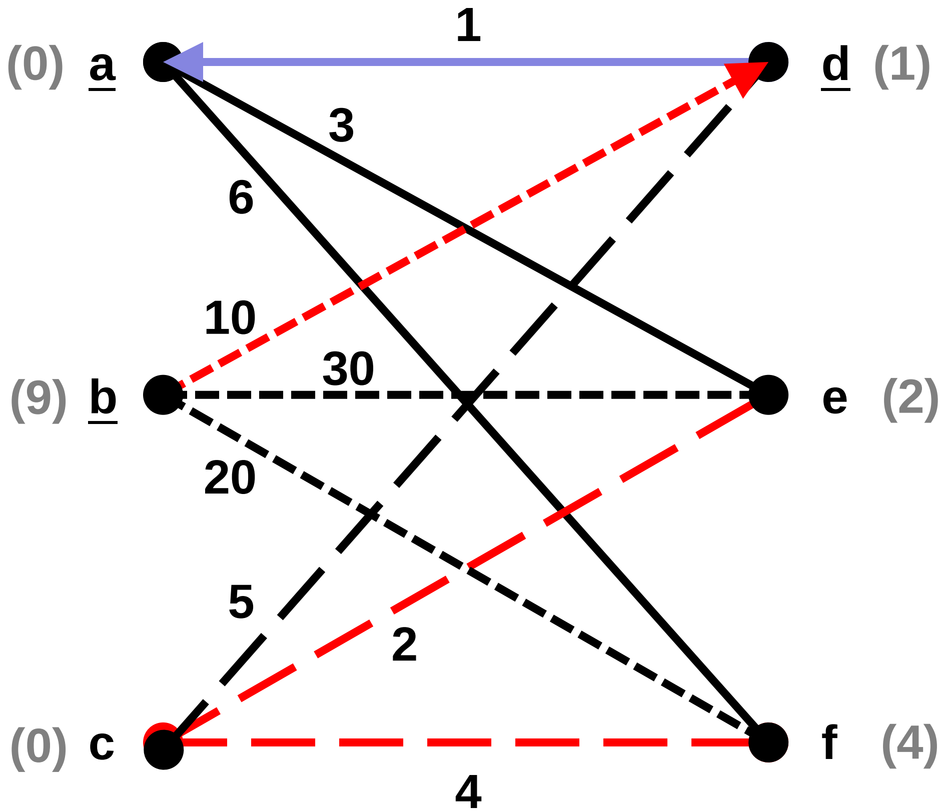
Un exemple d'exécution de la Méthode Hongroise

- Arêtes de $G^=$ en rouge
- Sommets soulignés = S(T)
- Voisinage dans $G^=$ de $S(T) \cap L = S(T) \cap R$ (vide) ==> mise à jour de la solution duale
- $q_{\mathbf{min}} = \min(10\text{-}1, 30\text{-}2, 20\text{-}4) = 9$
- $y_{\mathbf{b}} = y_{\mathbf{b}} + 9 = 0 + 9 = 9$
- Nouvelle arête dans $G^=$: bd



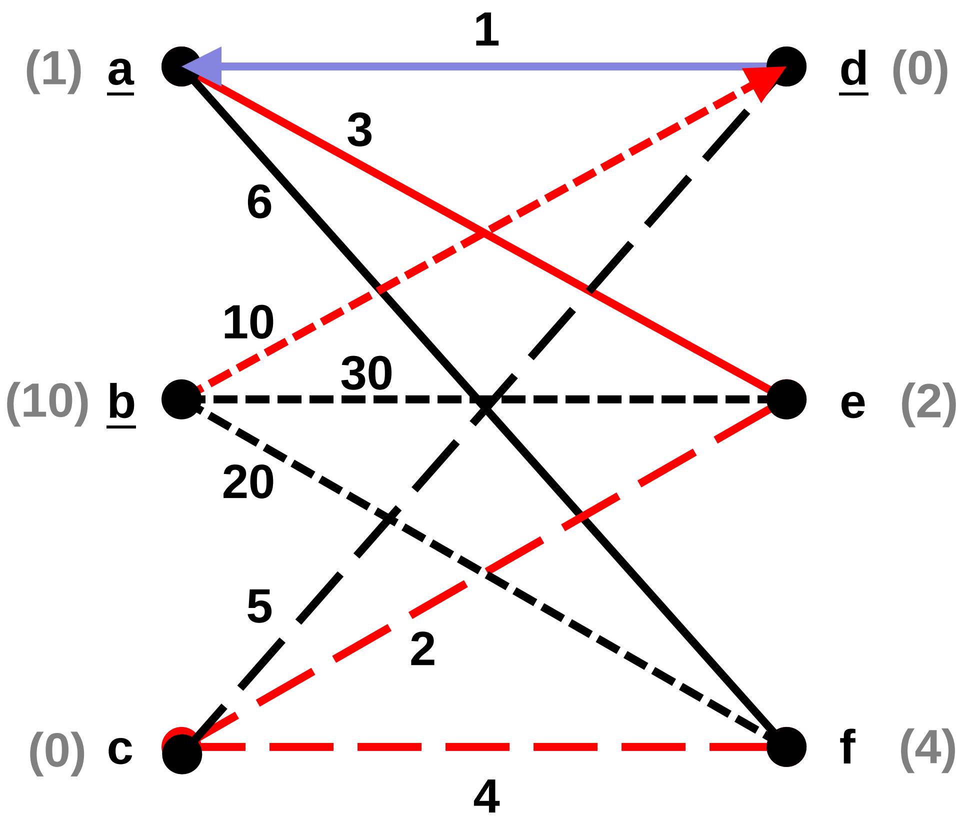
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arête bd = arête dans $G^=$ (rouge) avec b dans S(T)∩L et d dans R\(S(T)∩R)
- Sommet d ajouté à S(T) (sommets soulignés = S(T))
- Arête ad = arête dans M (en bleu) ==> d saturé par M
- On ajoute bd et da dans T, et le sommet a dans S(T)



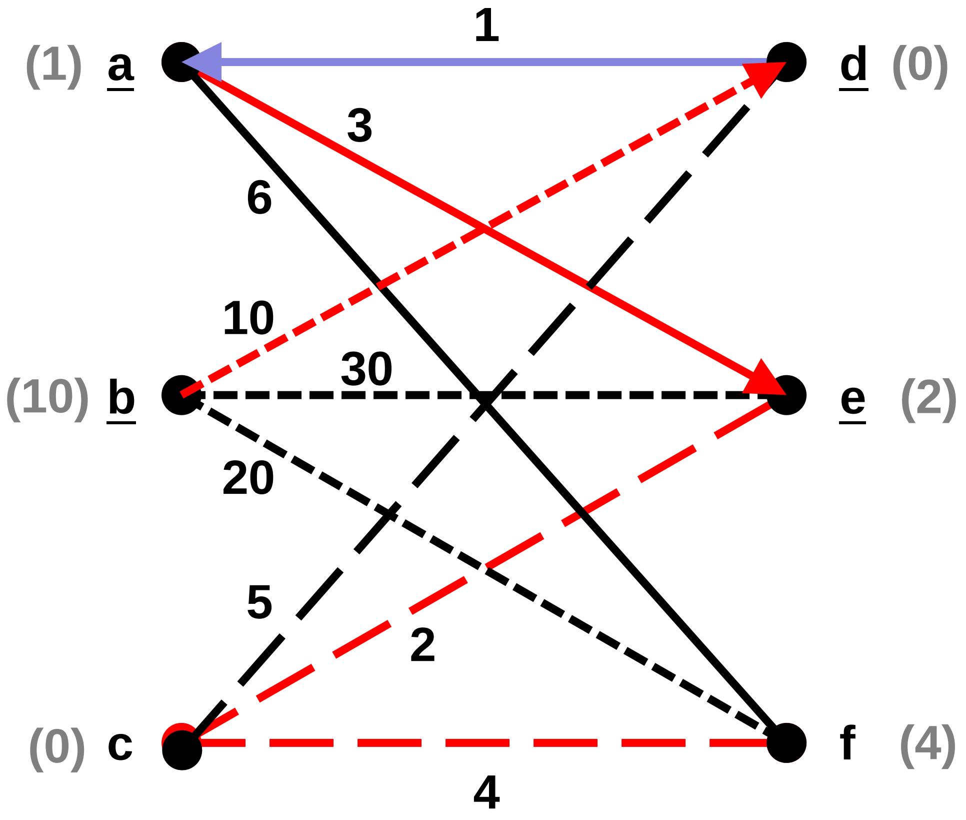
Un exemple d'exécution de la Méthode Hongroise

- Arêtes de $G^=$ en rouge
- Sommets soulignés = S(T)
- Voisinage dans $G^=$ de $S(T) \cap L = S(T) \cap R = \{d\}$ ==> mise à jour de la solution duale
- q_{min} = min(min(3-2-0, 6-4-0), min(30-9-2, 20-9-4)) = 1
- $y_b = y_b + 1 = 9 + 1 = 10$, $y_a = y_a + 1 = 0 + 1 = 1$, et $y_d = y_d - 1 = 1 - 1 = 0$
- Nouvelle arête dans $G^=$: ae



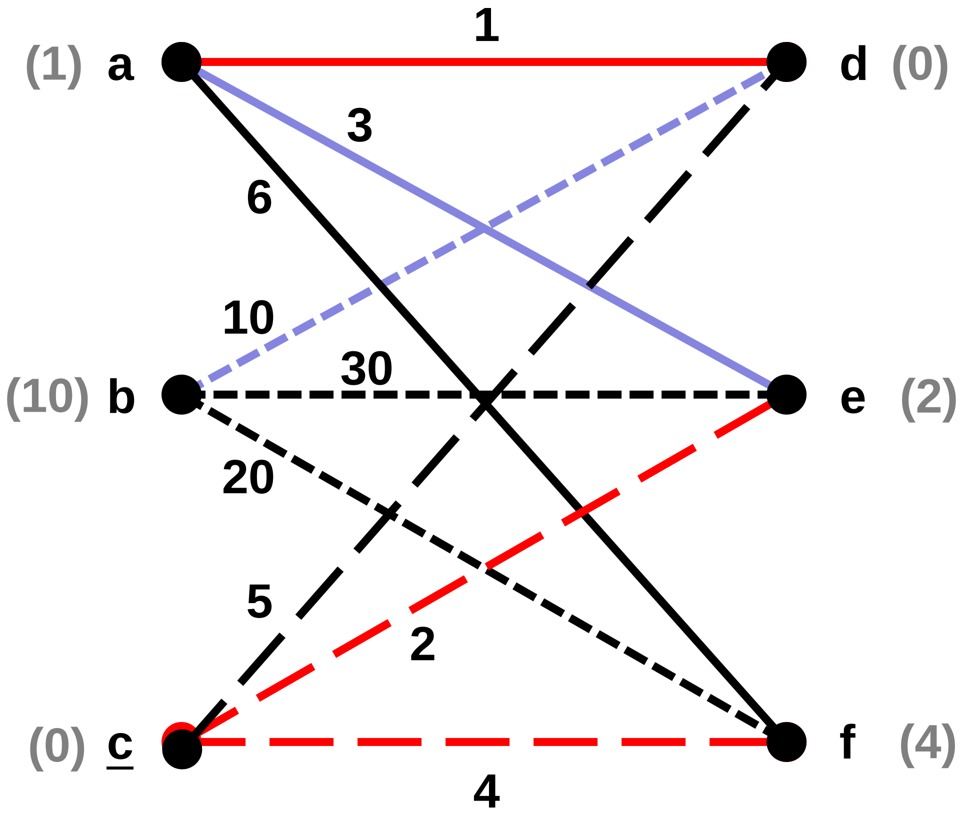
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arête ae = arête dans $G^=$ (rouge) avec a dans $S(T) \cap L$ et e dans $R \setminus (S(T) \cap R)$
- Sommet e ajouté à S(T) (sommets soulignés = S(T))
- e non saturé par M (arêtes en bleu) ==> bd+da+ae est une chaîne augmentante pour M



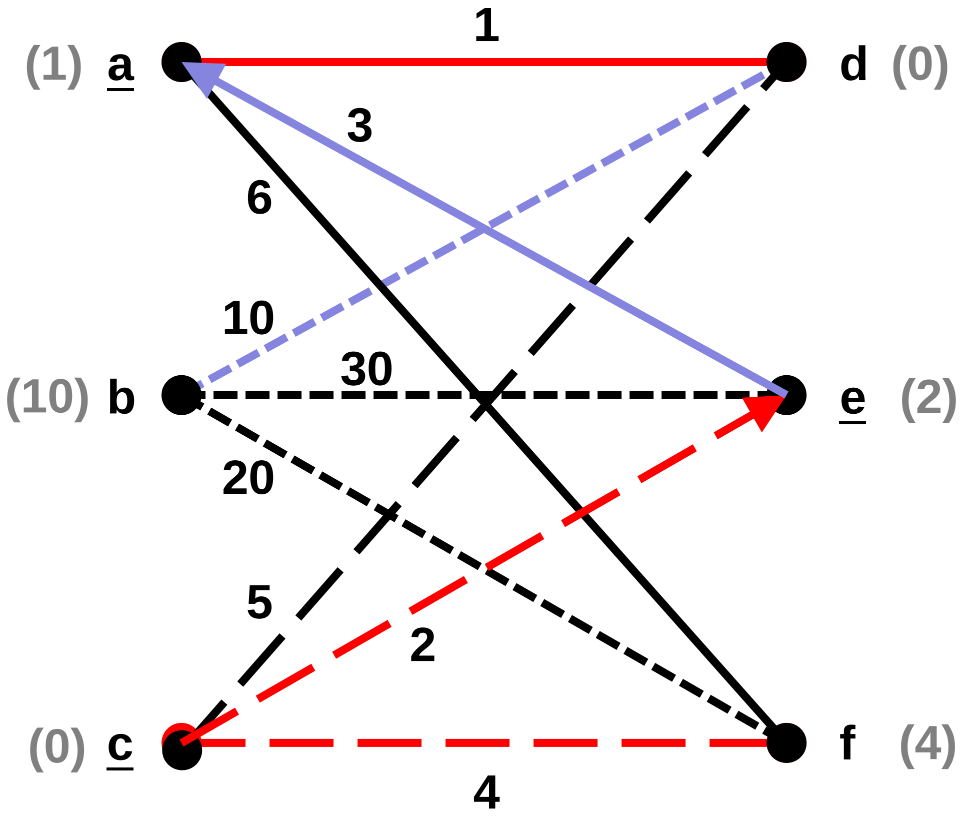
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arêtes de $G^=$ en rouge
- M mis à jour par ajout des arêtes bd et ae (en bleu), qui remplacent ad : $|M| = 2$
- Réinitialisation de l'arbre enraciné T : S(T) = {c}
- La racine de T est donc c
- Sommets soulignés = S(T)
- $S(T) \cap R$ est alors vide, alors que voisinage dans $G^=$ de $S(T) \cap L$ = {e, f}



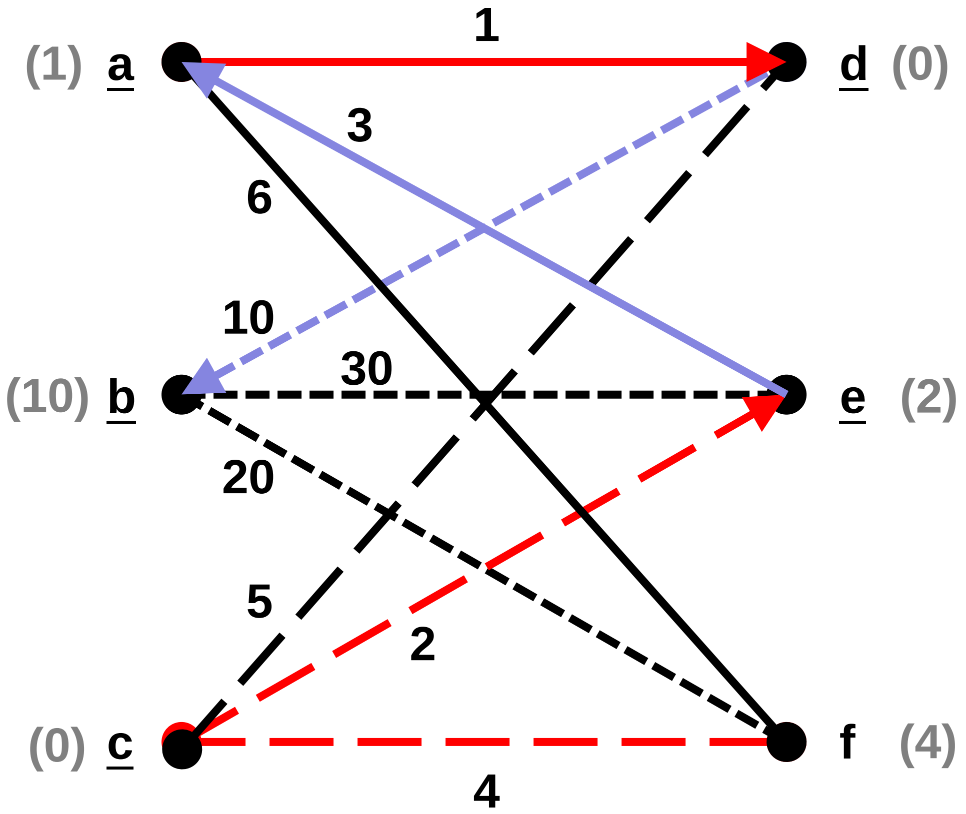
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arête ce = arête dans $G^=$ (rouge) avec c dans $S(T) \cap L$ et e dans $R \backslash (S(T) \cap R)$
- Sommet e ajouté à S(T) (sommets soulignés = S(T))
- Arête ea = arête dans M (en bleu) ==> e saturé par M
- On ajoute ce et ea dans T, et le sommet a dans S(T)



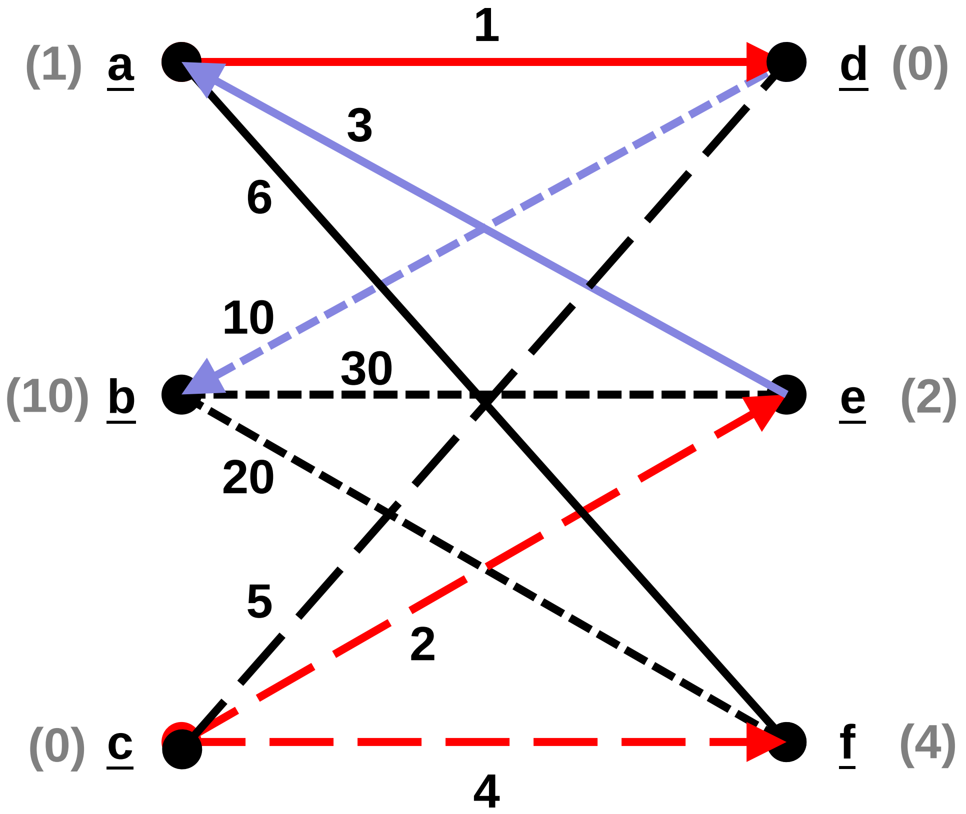
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arête ad = arête dans $G^=$ (rouge) avec a dans $S(T) \cap L$ et d dans $R \backslash (S(T) \cap R)$
- Sommet d ajouté à S(T) (sommets soulignés = S(T))
- Arête db = arête dans M (en bleu) ==> d saturé par M
- On ajoute ad et db dans T, et le sommet b dans S(T)



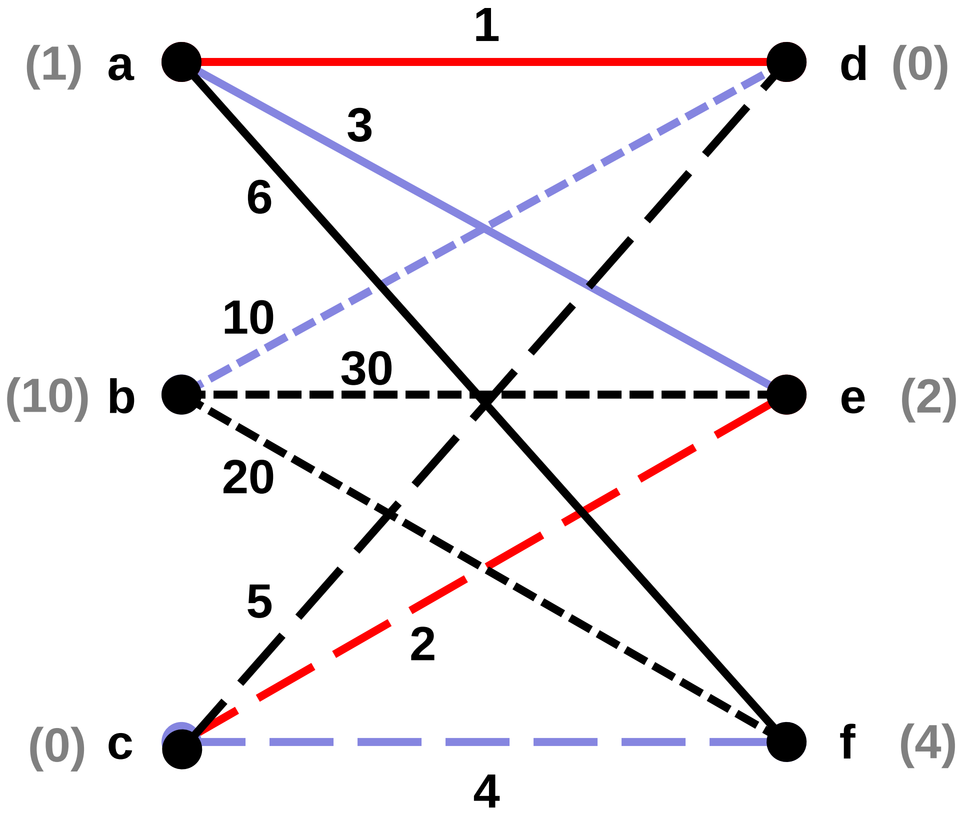
Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arête cf = arête dans $G^=$ (rouge) avec c dans S(T)∩L et f dans R\(S(T)∩R)
- Sommet f ajouté à S(T) (sommets soulignés = S(T))
- f non saturé par M (arêtes en bleu) ==> cf est une chaîne augmentante pour M



Un exemple d'exécution de la Méthode Hongroise

- Solution duale en gris
- Arêtes de $G^=$ en rouge
- M mis à jour par ajout de l'arête cf (en bleu) : on a alors |M| = 3 = |L| ==> M est un couplage parfait !
- Fin de l'exécution de la méthode : couplage parfait M obtenu = {ae, bd, cf}, de coût 3+10+4 = 17



Description synthétique de la méthode (rappel)

- Soit M := couplage vide, et y := solution duale admissible.
- Tant que M n'est pas un couplage parfait faire
  - Choisir un sommet r de L non saturé par M, et poser S(T) = {r}.
  - Tant que |M| n'a pas augmenté faire
    - Si $S(T) \cap R$ est égal au voisinage de $S(T) \cap L$ dans $G^=$, alors mettre à jour la solution duale (*ainsi*, $S(T) \cap R \neq$ *voisinage de* $S(T) \cap L$ *dans* $G^=$).
    - Soit uv une arête dans $G^=$ avec u dans $S(T) \cap L$ et v dans $R \setminus (S(T) \cap R)$.
    - Si v n'est pas saturé par M, alors il y a une chaîne augmentante dans T+{uv} ==> on met à jour M, et |M| augmente donc de 1.
    - Sinon, v est incident à une arête vw de M ==> ajouter uv et vw dans T, et ajouter v et w dans S(T).

Preuve de la méthode (1/2)

- Rappel : on veut obtenir une paire de solutions admissibles (l'une pour le primal, *donc un couplage parfait*, et l'autre pour le dual), qui vérifient la relation d'exclusion (*).
- Initialement, la solution duale est clairement admissible, et, par les règles de mise à jour, elle reste admissible tout du long.
- Il reste à montrer qu'à la fin on obtient toujours un couplage M parfait, qui en outre n'utilise que des arêtes dans $G^{=}$ (cf (*)).
  - Lors de la mise à jour de la solution duale, toute arête uv dans $G^{=}$ avec u et v dans S(T), ou ni u ni v dans S(T), reste dans $G^{=}$.
  - Tout arc de l'arborescence *alternée* T est soit dans M, soit dans $G^{=}$: ==> Après chaque augmentation, toute arête de M est dans $G^{=}$.
  - Tout du long, toute arête uv de M est donc dans $G^{=}$ (car soit u et v sont dans S(T), soit ni u ni v n'est dans S(T)) ==> vrai aussi pour T.

Preuve de la méthode (2/2)

- Après chaque mise à jour de la solution duale, $S(T) \cap R$ n'est plus égal au voisinage de $S(T) \cap L$ dans $G^=$:
  - En particulier, $S(T) \cap R$ n'est pas égal à R avant la mise à jour, car, à part la racine de T, tout sommet de S(T) est saturé par M.
  - Si $S(T) \cap R$ n'est pas égal au voisinage de $S(T) \cap L$ dans $G^=$, alors il y a dans $G^=$ une arête uv avec u dans $S(T) \cap L$ et v dans $R \setminus (S(T) \cap R)$, car, comme toute arête de T est dans $G^=$, tout sommet de $S(T) \cap R$ est forcément voisin dans $G^=$ d'un sommet de $S(T) \cap L$.
- Au pire, S(T) contient tous les sommets (ceux de L et ceux de R), et donc au bout d'un temps fini (?) on pourra augmenter |M|.
  - On finira donc par avoir $|M| = |L| = |R| \Longleftrightarrow M$ = couplage parfait.

Description synthétique de la méthode (2e rappel)

- Soit M := couplage vide, et y := solution duale admissible.
- Tant que M n'est pas un couplage parfait faire
  - Choisir un sommet r de L non saturé par M, et poser S(T) = {r}.
  - Tant que |M| n'a pas augmenté faire
    - Si S(T)∩R est égal au voisinage de S(T)∩L dans $G^=$, alors mettre à jour la solution duale (*ainsi, S(T)∩R ≠ voisinage de S(T)∩L dans* $G^=$).
    - Soit uv une arête dans $G^=$ avec u dans S(T)∩L et v dans R\(S(T)∩R).
    - Si v n'est pas saturé par M, alors il y a une chaîne augmentante dans T+{uv} ==> on met à jour M, et |M| augmente donc de 1.
    - Sinon, v est incident à une arête vw de M ==> ajouter uv et vw dans T, et ajouter v et w dans S(T).

Complexité (cubique) de la méthode (1/3)

- A chaque passage dans le « Tant que » externe, |M| augmente de 1 ==> O(|L|) passages en tout
- A chaque passage dans le « Tant que » interne, soit |M| augmente (fin de la boucle), soit |S(T)| augmente de 2 ==> O(|L|) passages
- Structures de données utilisées :
  - 2 tableaux de booléens de taille O(|R|) = O(|L|) pour représenter S(T)∩R et le voisinage de S(T)∩L dans $G^=$ (= *vrai* ssi le sommet considéré est dans l'ensemble), et 1 tableau de taille O(|L|) pour y
  - Une variable égale à |M|, et initialisée à 0
  - 1 tableau de booléens de taille O(|L|) indiquant, pour chaque sommet x, s'il est (et reste) saturé par M (= *vrai* ssi M sature x)
  - Un tableau d'entiers appelé *écart* : on aura, pour tout sommet v dans R\(S(T)∩R), écart[v] = $\min_{\mathbf{u\ dans\ S(T)\cap L}}$ (cout(u,v) - $y_{\mathbf{u}}$ - $y_{\mathbf{v}}$)

Complexité (cubique) de la méthode (2/3)

- Solution duale y initialisée en temps $O(|L|^2)$
- Choix de r en temps $O(|L|)$, puis $S(T)\cap R$ (qui est vide) et voisinage dans $G^=$ de $S(T)\cap L = \{r\}$ initialisés en temps $O(|L|)$
- Tableau *écart* initialisé en temps $O(|L|^2)$
- Test de l'égalité entre $S(T)\cap R$ et le voisinage de $S(T)\cap L$ dans $G^=$ en temps $O(|L|)$, grâce aux 2 tableaux adéquats
- Mise à jour de la solution duale en temps $O(|L|)$:
  - Calcul de $q_{\mathbf{min}} = \min_{\mathbf{v\ dans\ R\backslash(S(T)\cap R)}}$ (écart[v]), en temps $O(|L|)$
  - Mise à jour de la solution duale y et de écart[v] := écart[v] – $q_{\mathbf{min}}$, pour tout sommet v dans $R\backslash(S(T)\cap R)$, en temps $O(|L|)$
  - Calcul du nouveau voisinage de $S(T)\cap L$ dans $G^=$ en temps $O(|L|)$ (= sommets v dans $R\backslash(S(T)\cap R)$ tels que écart[v] = 0 désormais)

Complexité (cubique) de la méthode (3/3)

- Choix de l'arête uv appartenant à $G^=$ avec u dans $S(T)\cap L$ et v dans $R\setminus(S(T)\cap R)$ en temps $O(|L|)$:
  - Sommet v pas dans $S(T)\cap R$ tel que écart[v]=0, puis recherche de u
- Mise à jour du couplage M (solution primale) en temps $O(|L|)$:
  - Calcul de la chaîne augmentante dans T en temps $O(T) = O(|L|)$
  - Mise à jour de M, $|M|$ et des sommets saturés par M en temps $O(|L|)$
- Mise à jour de l'arborescence alternée T en temps $O(|L|)$:
  - Arête vw de M saturant le sommet v déterminée en temps $O(|L|)$
  - Mise à jour de T et de S(T) en temps O(1), puis de $S(T)\cap R$, du voisinage de $S(T)\cap L$ dans $G^=$, et enfin des valeurs du tableau *écart*, en temps $O(|L|)$, car v est alors dans $S(T)\cap R$ et w dans $S(T)\cap L$
- Bilan : complexité globale en temps $O(|L|^3) = O(|R|^3)$!

Bilan sur les approches primales-duales

- Les propriétés d'une bonne formulation PL peuvent parfois permettre d'établir la polynomialité du problème étudié
- Souvent, cela n'est qu'une première étape dans la résolution pratique du problème, et **signifie en fait qu'il existe un algorithme combinatoire efficace** pour résoudre ce problème sans passer par la PL
- **Un algorithme primal-dual est un bon candidat** dans ce cas-là, puisque **ses « choix » sont guidés par les conditions des écarts complémentaires** (relations d'exclusion) du PL associé
- MAIS : dans certains cas, cela ne marche pas si bien !
- Enfin, ce genre d'approches peut parfois être utilisé également pour obtenir des solutions approchées à des problèmes difficiles (cf CAP)