Université de Nantes

UNIVERSITÉ DE NANTES

{département} INFORMATIQUE

M1 Informatique – Graphes II et Réseaux
Problèmes de transbordement

Irena.Rusu@univ-nantes.fr

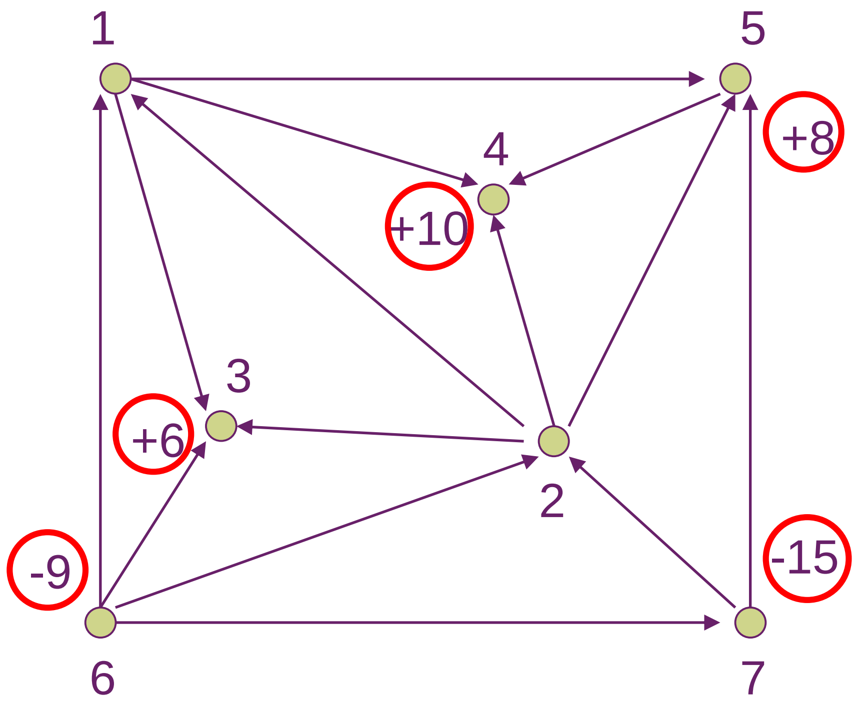
LS2N, bât. 34, bureau 303

tél. 02.51.12.58.16

- Le problème et ses solutions réalisables
- L' algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d'intégralité
- Un exemple de planification de production

- Le problème et ses solutions réalisables
- L' algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d'intégralité
- Un exemple de planification de production

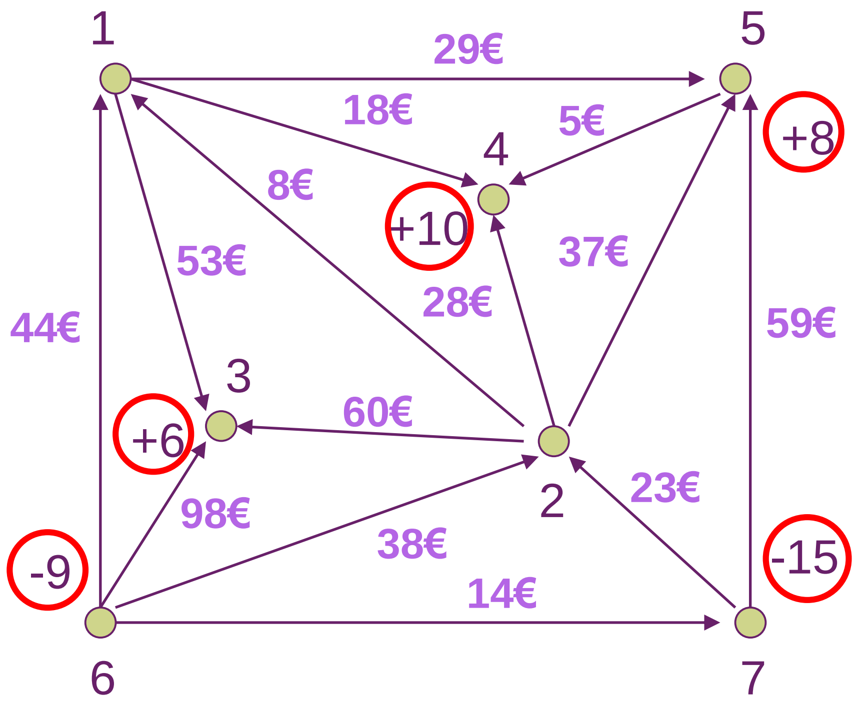
Trouver le moyen le moins cher d'acheminer des quantités indiquées d'une certaine marchandise (disons des pommes) depuis des sources spécifiées vers des destinations spécifiées utilisant un réseau de transport donné.



-9 → **9t** de pommes doivent partir

6 → **6t** de pommes sont attendues

Trouver le moyen le moins cher d'acheminer des quantités indiquées d'une certaine marchandise (disons des pommes) depuis des sources spécifiées vers des destinations spécifiées utilisant un réseau de transport donné.



-9 → **9t** de pommes doivent partir

6 → **6t** de pommes sont attendues

Transporter 1t de 2 à 3 coûte 60 €

L’offre totale est égale à la demande totale.

Offre : 9 + 15 = 24

Demande : 6 + 10 + 8 = 24

Si ceci n’est pas vrai en pratique, ajouter un nœud et des arcs selon les besoins.

- Un problème de transbordment est tout problème

  minimiser cx sous les contraintes $Ax=b,\ x\geq 0$

qui satisfait :

- A est la matrice d'incidence $n \times m$ d'un réseau

  (sur la colonne ij: valeur -1 sur la ligne i, valeur 1 sur la ligne j et 0 partout ailleurs),
- c est le vecteur ligne des coûts sur les arcs, et
- b est le vecteur colonne des demandes définies sur les nœuds, de sorte que les offres soient représentées comme des demandes négatives et

$$\sum_{i=1}^{n} b_i = 0$$

Solution réalisable:

- Ensemble d'arcs du réseau
- Quantités positives x_{ij} acheminées le long de chaque arc ij de l'ensemble.
- Sur les autres arcs, les quantités sont supposées nulles.
- De sorte que, pour tout noeud i :

$$\sum_{ki} x_{ki} - \sum_{ij} x_{ij} = b(i)$$

Offre ou
demande dans ***i,***
ou 0 dans le cas contraire

Solution-arbre réalisable :

- Solution réalisable
- Les arcs induisent, lorsque leur orientation est oubliée, un arbre recouvrant du réseau.

Proposition (Rappel)

Un graphe non-orienté admet un arbre recouvrant si et seulement si il est connexe.

Proposition

Un arbre recouvrant du réseau (version non-orientée) induit soit une solution-arbre unique, soit aucune solution-arbre.

- Le problème et ses solutions réalisables
- L' algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d'intégralité
- Un exemple de planification de production

Pas initial. Commencer par un arbre recouvrant et la solution-arbre correspondante.

Boucle. Tant que la solution n'est pas optimale faire

- Choisir un arc entrant (un pivot)
- Choisir un arc sortant
- Calculer le nouvel arbre recouvrant
- Calculer sa solution-arbre réalisable

Pas initial. Commencer par un dictionnaire réalisable et sa solution de base réalisable.

Boucle. Tant que la solution n'est pas optimale faire

- Choisir une variable entrante (une variable pivot)
- Choisir une variable sortante
- Calculer le nouveau dictionnaire réalisable
- Calculer sa solution de base réalisable.

Réseau

$$G=(V(G), A(G)) \quad \text{avec} \quad V(G)=\{1, 2, \ldots, n\}$$

Arbre recouvrant *T* avec

$V(T)$ = tous les sommets du réseau (c.à.d. *1, 2, …, n*).

$A(T)$ = les arcs de *T* (avec leurs orientations)

Solution-arbre réalisable :

$$X_{ij}>0, \text{ pour tous } ij \in A(T)$$

- Soit y_i le prix unitaire légitime pour notre marchandise au sommet i.

- Les prix unitaires dépendent les uns des autres comme suit :

  $y_i + c_{ij} = y_j$ pour tous $ij \in A(T)$

- Un prix unitaire (choisi arbitrairement) fixe tous les autres prix unitaires.

- Convention: $y_n = 0$

Choisir un arc entrant (II): recherche d'un meilleur acheminement

- Est-ce qu'il est possible d'avoir un prix unitaire plus petit à un certain sommet i, si un arc qui n'est pas dans T (mais qui est dans le réseau) est utilisé ?

- Trouver un arc ij n'appartenant pas à T tel que

$$y_i + c_{ij} < y_j$$

- Meilleur choix intuitif : ij qui atteint la valeur

$$\max_{kl \in A(G) - A(T)} \{y_l - (y_k + c_{kl})\}$$

(c'est la règle de pivotage du plus grand mérite)

- L'arc entrant s'appelle aussi le pivot.

Choisir un arc sortant (I): ajuster les acheminements

- Soit t = nombre d'unités acheminées par l'arc entrant ij.
- Comment ajuster les unités acheminées le long des arcs de T de sorte à avoir une solution-arbre résultante qui soit réalisable ?
- Le long du cycle créé par ij dans T, modifier les quantités sur chaque arc kl comme suit :
  - Si kl, ij ont la même direction : ajouter t unités
  - Sinon : soustraire t unités

- Quelle est la valeur maximum possible pour t qui assure des valeurs positives ou *0* le long du cycle ?

- Pour chaque arc, énoncer la contrainte le concernant.

- Définir l'arc sortant kl comme l'un de ceux qui imposent la contrainte la plus forte pour t, et attribuer à t la valeur indiquée par cette contrainte.

Note. Si aucune contrainte n'est trouvée (ce qui arrive si tous les arcs sont dans la même direction) alors le problème est non-borné.

- Mettre à jour les arcs de T.
- Mettre à jour les quantités x_{ij} sur les arcs.
- Mettre à jour les prix y_i.

Test d'optimalité :

$$y_i + c_{ij} \geq y_j \quad \textit{pour tous} \quad ij \in A(T)$$

(il n'y a plus d'arc entrant)

- Le problème et ses solutions réalisables
- L’ algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d’intégralité
- Un exemple de planification de production

- Plusieurs candidats pour l'arc sortant impliquent au moins une valeur *0* dans la nouvelle solution-arbre.
- $t=0$ est donc possible à l'itération suivante (l'arc entrant – c'est-à-dire le pivot – est alors appelé dégénéré)
- La méthode ne réalise aucun progrès pendant plusieurs (voir beaucoup) d'itérations (l'arbre change mais pas x).

- A cause de la dégénérescence, une configuration précédente (même arbre recouvrant et même solution réalisable) réapparaît.
- La même configuration suivie des mêmes itérations produisent une boucle infinie.
- Il y a des méthodes pour éviter les boucles infinies.

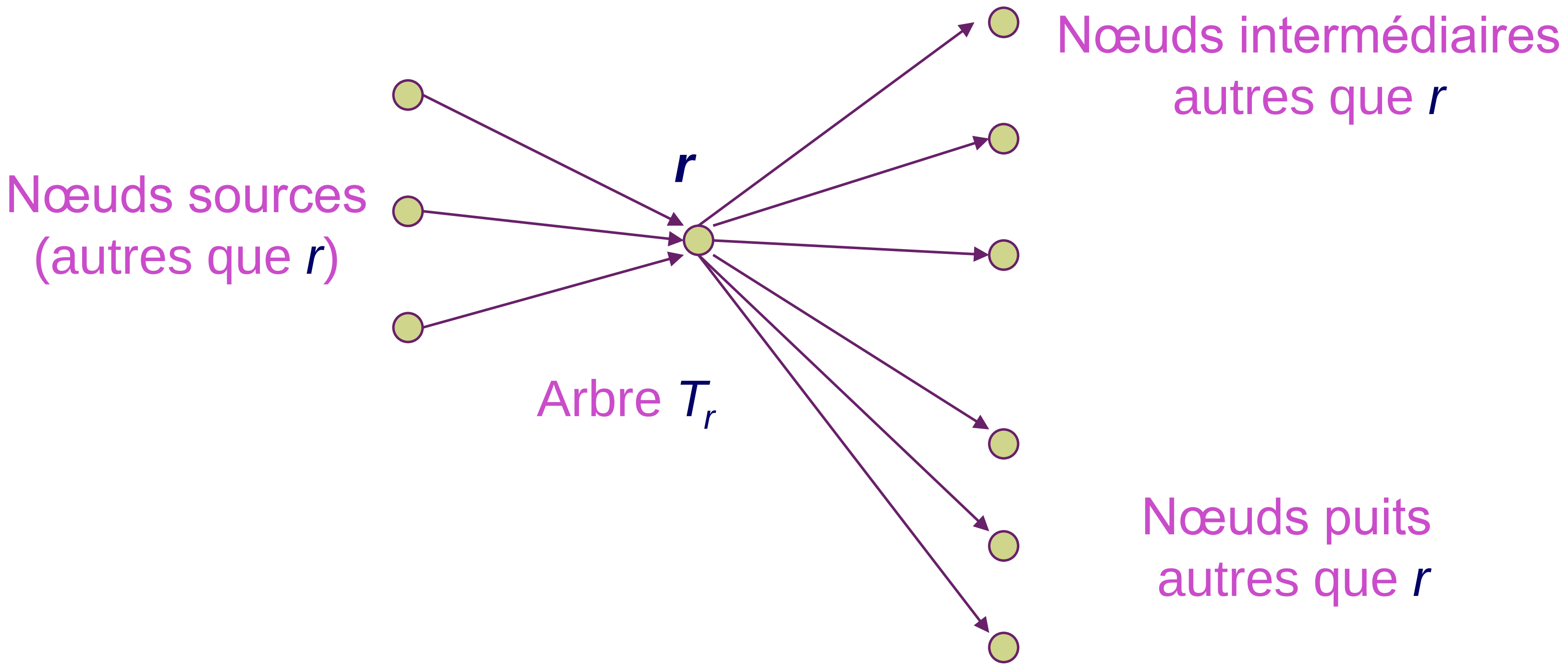
- Si une solution-arbre réalisable initiale est trouvée et que l'algorithme ne boucle pas, alors la méthode simplexe réseau termine après un nombre fini d'itérations et produit une solution optimale.

Combien d'itérations ?

- Dépend de la règle de choix du pivot.
- Les règles du plus grand mérite et celle du premier candidat (définies intuitivement) sont bonnes en pratique. Toutefois, des exemples avec un nombre exponentiel d'itérations existent pour la première.
- En pratique, la méthode est très rapide.

- Le problème et ses solutions réalisables
- L' algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d'intégralité
- Un exemple de planification de production

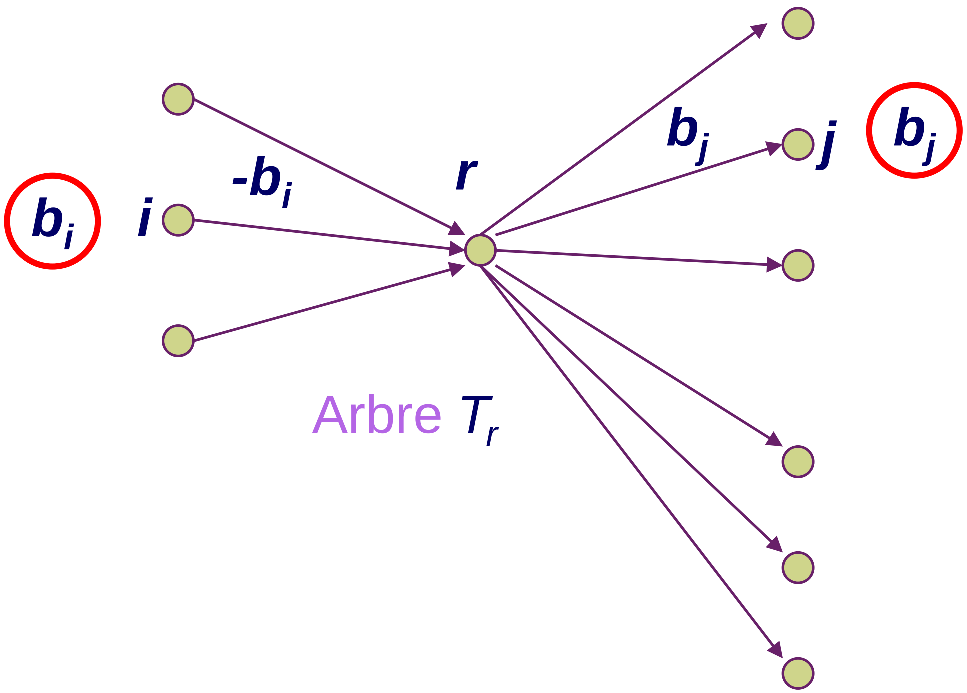
- Cas facile : il existe un nœud *r* dans l'arbre tel que



D'autres arcs peuvent exister dans le réseau

- Cas facile: solution-arbre réalisable pour T_r

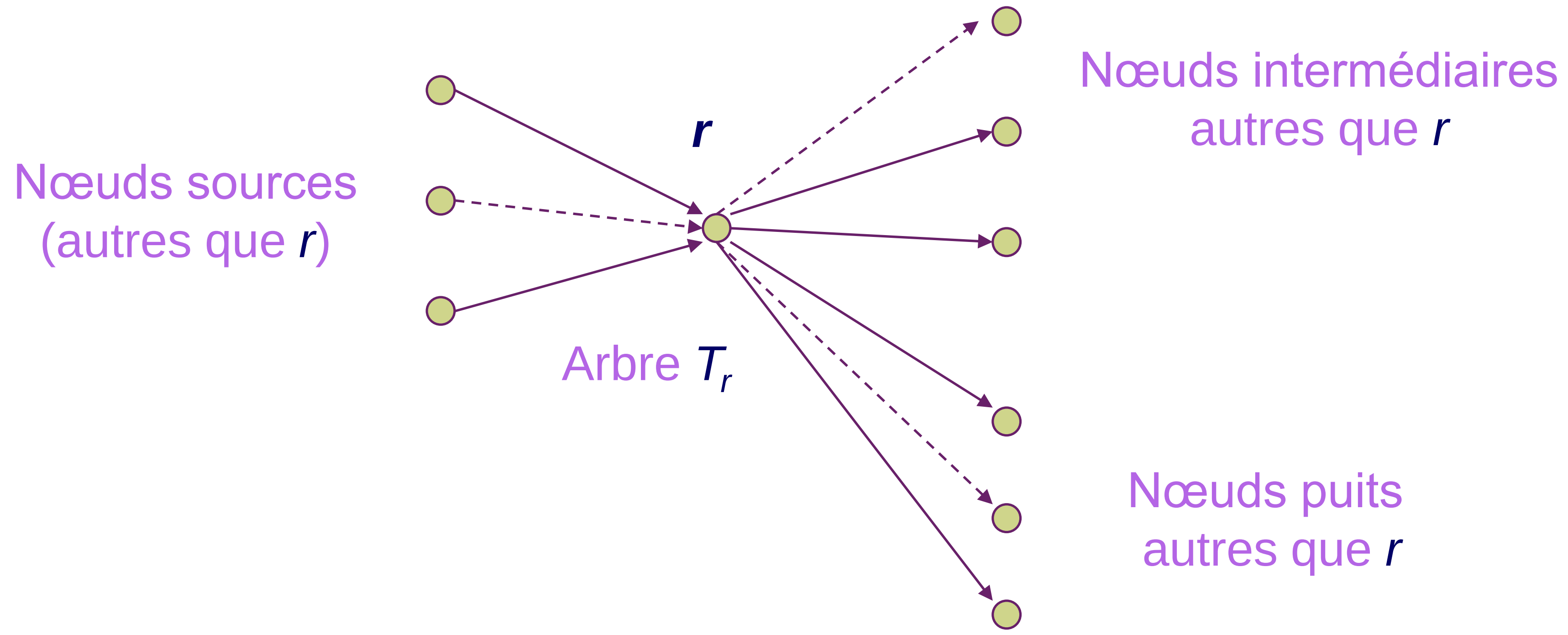
x_{ir}=-b_i si b_i<0



x_{rj}=b_j si b_j≥0

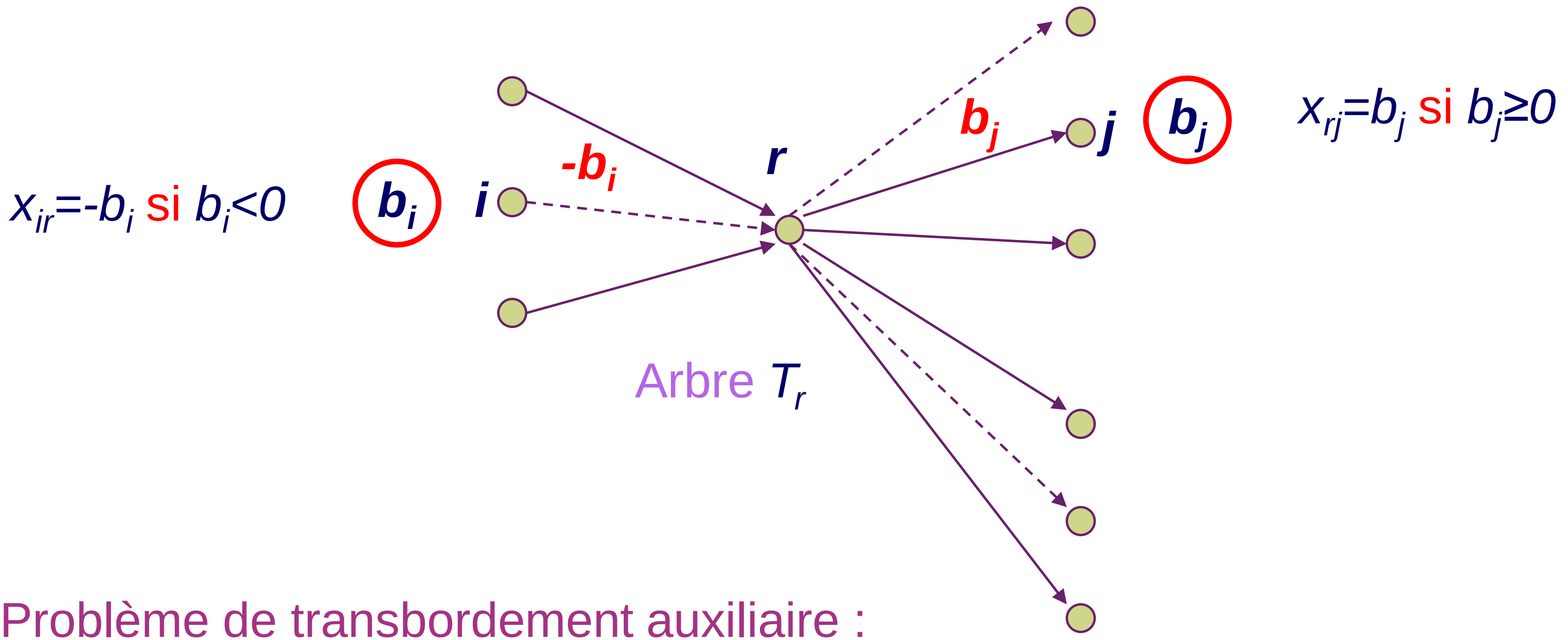
D'autres arcs peuvent exister dans le réseau

- Cas difficile: choisir arbitrairement un nœud *r* dans le réseau et ajouter des arcs artificiels au réseau de sorte que :



D'autres arcs peuvent exister dans le réseau

- Cas difficile: solution-arbre réalisable pour T_r



Problème de transbordement auxiliaire :

minimiser px sous les conditions $Ax=b, x\geq 0$
où le nouveau coût p est égal à 1 pour les arcs artificiels et à 0 pour les arcs réels.

- Cas difficile : résoudre le

  Problème de transbordement auxiliaire :

  minimiser px sous les conditions $Ax=b, x\geq 0$

  où le nouveau coût p est égal à 1 pour les arcs artificiels et à 0 pour les arcs réels.

- Et soient T et x' l'arbre final et la solution-arbre optimale.

  Trois cas :

  - T contient un arc artificiel uv tel que $x'_{uv} > 0$
  - T ne contient pas d'arc artificiel
  - T contient au moins un arcs artificiel, mais tout arc artificiel ij satisfait $x'_{ij}=0$

- Cas 1: T contient un arc artificiel uv tel que $x'_{uv} > 0$

  Le problème initial n'a pas de solution réalisable.

- Cas 2: T ne contient pas d'arc artificiel.

  La méthode simplexe réseau sur le problème initial est initialisée avec T et x'.

- Cas 3: T contient au moins un arc artificiel, mais tout arc artificiel ij satisfait $x'_{ij}=0$.

  Le problème initial a une solution réalisable (donnée par les valeurs de x' sur les arcs du réseau), mais peut ne pas avoir de solution-arbre réalisable → décomposition en sous-problèmes.

- Le problème et ses solutions réalisables
- L' algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d'intégralité
- Un exemple de planification de production

Théorème.

Le problème de transbordement

minimiser *cx* sous les conditions *Ax=b, x≥0*

tel que tous les composantes de *b* sont des entiers a les propriétés :

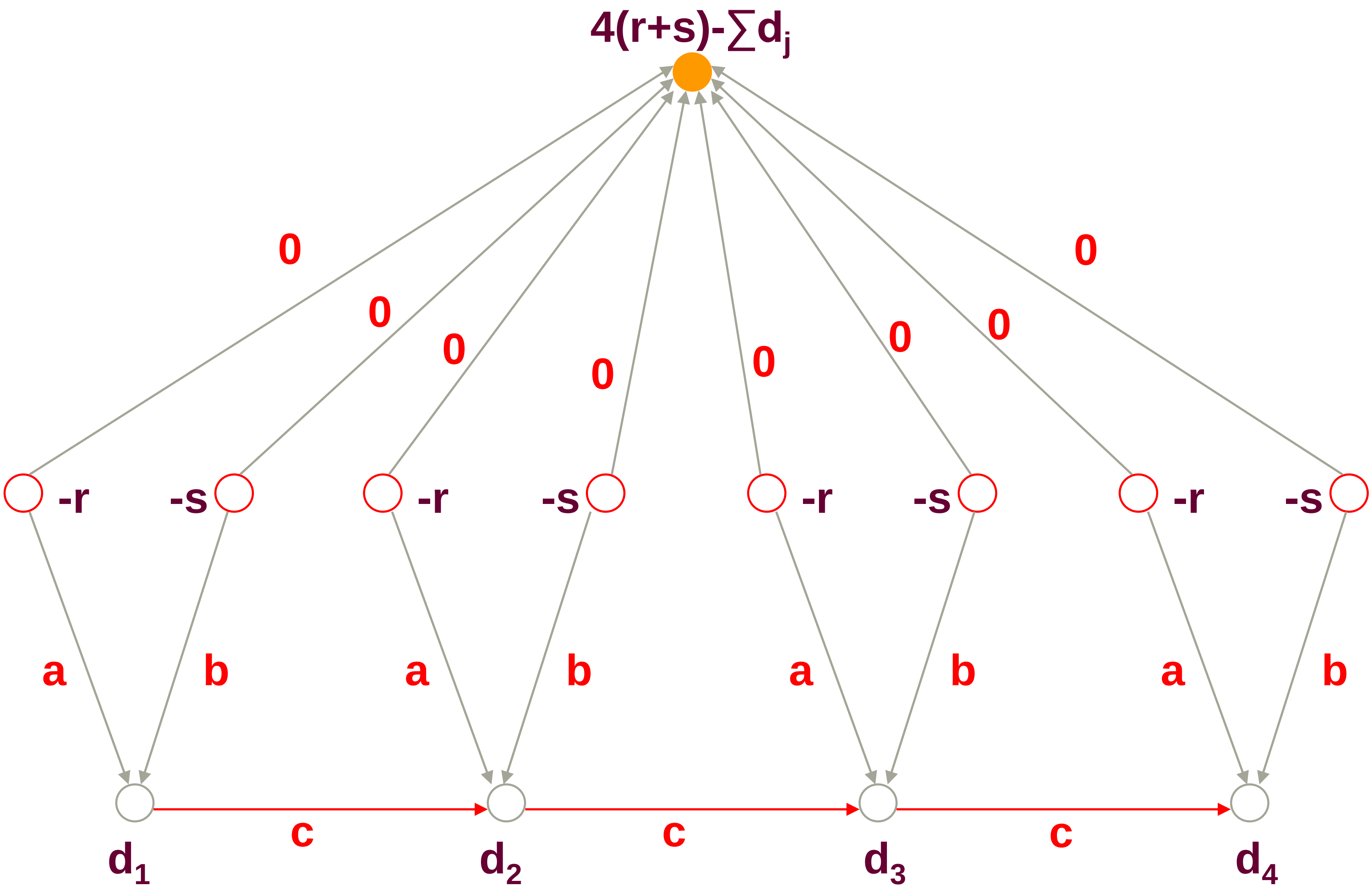
- Si le problème a une solution réalisable, alors il a une solution réalisable a valeurs entières.
- Si le problème a une solution optimale, alors il a une solution optimale à valeurs entières.

- Le problème et ses solutions réalisables
- L’ algorithme simplexe réseau: … lignes générales
- … et détails:
  - Dégénerescence, boucle infinie, terminaison
  - La solution initiale réalisable
- Le théorème d’intégralité
- Un exemple de planification de production

- Une usine fabrique des produits dont la demande change sur une période de n mois ; chaque mois j, d_j unités sont nécessaires.

- Il existe trois moyens d'ajuster le volume de produits sortis de l'usine :
  - Changer le niveau de production régulière ; il n'y a pas de pénalités pour ce changement, mais le niveau est limité à au plus r unités par mois.
  - Utiliser de la production en heures supplémentaires pour couvrir les besoins ponctuels ; le niveau de production en heures supplémentaires est limité à au plus s unités par mois ; de plus, la production en heures supplémentaires est de b euros par unité, et dépasse le coût de production régulière qui est de a euros par unité.
  - Stocker l'excédent à un moment donné pour couvrir les besoins futurs ; il n'y a pas de limite sur le volume stocké, mais il coûte c euros pour stocker une unité, par mois.

- Problème: répondre correctement à la demande, à un prix minimum.


4(r+s)-∑dj
0
0
0
0
0
0
0
0
-r
-s
-r
-s
-r
-s
-r
-s
a
b
a
b
a
b
a
b
c
c
c
d1
d2
d3
d4

- Cas particulier du problème du flot de valeur fixée et de coût minimum (qui n'est pas bien différent de celui du flot maximum de coût minimum), mais l'algorithme du simplexe réseau est généralisable pour résoudre aussi ce problème.

- Bibliographie

  - V. Chvatal – Linear Programming, W.H. Freeman and Company, 1999
  - I. Charon, A. Germa, O. Hudry – Méthodes d'optimisation combinatoire, Masson, 1996.