Fiche Méthodes - SM602

Un recueil de méthodes pour accompagner les révisions de Recherche Opérationnelle

DorYan Denis

2025-2026

Table des matières

1 Problèmes de transports **2**
- 1.1 Trouver une proposition initiale 2
  - 1.1.1 Méthode du coin Nord-Ouest 2
  - 1.1.2 Méthode de Balas-Hammer 5
  - 1.1.3 Un cas particulier... 7
- 1.2 Optimiser la proposition initiale 7
  - 1.2.1 Graphe associé et proposition non-dégénérée 7
  - 1.2.2 Calcul des potentiels 9
  - 1.2.3 Matrices des coûts potentiels et marginaux 10
  - 1.2.4 Fin ou bouclage 11

2 Programmation linéaire et la méthode du simplexe **14**
- 2.1 Traduire un problème en un programme linéaire 14
- 2.2 Résoudre graphiquement un programme linéaire 16
- 2.3 Utiliser la méthode du simplexe 17
  - 2.3.1 Passage à la forme standard 18
  - 2.3.2 Construction du tableau et choix du pivot 19
  - 2.3.3 Exécution du pivot de Gauss 21
  - 2.3.4 Objectif non borné 23
- 2.4 Initialiser l'algorithme du simplexe 24

3 La dualité **28**
- 3.1 Écrire le dual d'un programme linéaire 28
- 3.2 Résoudre le primal à l'aide du dual géométriquement 30
- 3.3 Réaliser une analyse en stratégies pures (jeu à somme nulle 2×2) 32
- 3.4 Résoudre des problèmes en stratégies mixtes (jeu à somme nulle 2×2) 34
- 3.5 Jeux matriciels de taille $2 \times p$ 36

4 Problèmes de flots **40**
- 4.1 Méthode de Ford-Fulkerson 40
- 4.2 Algorithme Pousser-Réétiqueter 43
- 4.3 Problème de flot à coût minimal 48

1 Problèmes de transports

Ce premier chapitre est consacré aux problèmes de transports. Ils consistent à trouver le plan le moins cher (ou le plus rapide) pour transporter des marchandises de plusieurs points de départ vers plusieurs points d'arrivée. C'est un type particulier de programmation linéaire (nous y reviendrons au chapitre 2) qui cherche à répondre à la question : "Qui doit livrer qui pour minimiser les frais ?"

Pour modéliser ce problème, nous avons besoin de trois composantes :

- Les **Sources** (L'Offre) : Ce sont les lieux qui possèdent les provisions en marchandise (ex : usines, entrepôts). Chaque source a une capacité limitée (elle ne peut pas livrer plus qu'elle ne possède).
- Les **Destinations** (La Demande) : Ce sont les lieux qui commandent de la marchandise (ex : magasins, clients). Chaque destination a une demande précise à satisfaire.
- Les **Coûts de Transport** : C'est le prix pour déplacer une unité d'une source spécifique vers une destination spécifique (ex : livrer 1 tonne de Paris à Lyon coûte 50€).

Le but est de déterminer exactement combien d'unités chaque source doit envoyer à chaque destination pour que toutes les demandes des clients soient satisfaites, que les capacités des sources ne soient pas dépassées et que le coût total de transport soit le plus bas possible.

On représente ce problème sous la forme d'un tableau comme suit :

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60
S_2	17	16	15	18	30
S_3	19	21	20	22	90
Commande	50	75	30	25	180

1.1 Trouver une proposition initiale

Avant de vouloir optimiser tout ça, il faut débuter avec une solution de départ qui n'est pas forcément la meilleure (c'est le but de l'optimisation d'améliorer cette proposition initiale !). Pour cela nous avons 2 méthodes que l'on va détailler maintenant.

1.1.1 Méthode du coin Nord-Ouest

Méthode. Principe et Algorithme

Cette première méthode est la technique la plus simple pour trouver une première solution réalisable à un problème de transport. Son principe est purement géographique : on commence par la case tout en haut à gauche (le Nord-Ouest) et on descend en escalier vers le bas à droite (le Sud-Est), en remplissant les cases au maximum sans se soucier des coûts.

On répète ainsi la boucle suivante tant qu'il reste des provisions à distribuer :

1. **Ciblage**
   Sélectionnez toujours la case disponible située le plus en **haut à gauche** du tableau (ou de la partie du tableau qui n'a pas encore été rayée).
2. **Allocation**
   Pour cette case, comparez la **Provision restante** de la source (Ligne) et la **Commande restante** du client (Colonne).
   - Attribuez à cette case la valeur la plus petite des deux.
   - Cela garantit que vous livrez le maximum possible sans dépasser le stock ni livrer trop au client.

3. **Mise à jour et Saturation**
   Soustrayez la quantité que vous venez d'attribuer des totaux de la ligne et de la colonne.
   - **Si la Provision de la source tombe à 0 :** La source est vide. La ligne est "saturée". Vous devez changer de source.
   - **Si la Commande du client tombe à 0 :** Le client est servi. La colonne est "saturée". Vous devez changer de client.
4. **Déplacement**
   - Si vous avez rayé la **ligne** (source vide) $\rightarrow$ Descendez d'une case (vers la source suivante).
   - Si vous avez rayé la **colonne** (client servi) $\rightarrow$ Déplacez-vous à droite d'une case (vers le client suivant).

L'algorithme s'arrête lorsque toutes les offres et toutes les demandes sont à zéro.

Exemple. Trouver une proposition initiale au problème de transport suivant en appliquant la méthode du coin Nord-Ouest :

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60
S_2	17	16	15	18	30
S_3	19	21	20	22	90
Commande	50	75	30	25	180

Solution.
Itération 1 (Rouge) :

- On cible la case Nord-Ouest : (S_1, C_1).
- $\min(\text{Prov } S_1 = 60, \text{Cmd } C_1 = 50) = 50$. On affecte **50**.
- S_1 restant : $60 - 50 = 10$. C_1 restant : $50 - 50 = 0$.
- La commande C_1 est satisfaite (0), on raye la colonne et on se déplace à droite vers (S_1, C_2).

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60 (reste 10)
	50				
S_2	17	16	15	18	30
	0				
S_3	19	21	20	22	90
	0				
Commande	50 (reste 0)	75	30	25	180

Itération 2 (Vert) :

- Case courante : (S_1, C_2).
- $\min(\text{Prov } S_1 = 10, \text{Cmd } C_2 = 75) = 10$. On affecte **10**.
- S_1 restant : $10 - 10 = 0$. C_2 restant : $75 - 10 = 65$.
- La provision S_1 est épuisée (0), on raye la ligne et on descend vers (S_2, C_2).

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60 (reste 0)
	50	**10**	0	0	
S_2	17	16	15	18	30
	0				
S_3	19	21	20	22	90
	0				
Commande	50 (reste 0)	75 (reste 65)	30	25	180

Itération 3 (Violet) :

- Case courante : (S_2, C_2).
- $\min(\text{Prov } S_2 = 30, \text{Cmd } C_2 = 65) = 30$. On affecte **30**.
- S_2 restant : $30 - 30 = 0$. C_2 restant : $65 - 30 = 35$.
- La provision S_2 est épuisée (0), on raye la ligne et on descend vers (S_3, C_2).

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60 (reste 0)
	50	**10**	0	0	
S_2	17	16	15	18	30 (reste 0)
	0	**30**	0	0	
S_3	19	21	20	22	90
	0				
Commande	50 (reste 0)	75 (reste 35)	30	25	180

Itération 4 (Orange) :

- Case courante : (S_3, C_2).
- $\min(\text{Prov } S_3 = 90, \text{Cmd } C_2 = 35) = 35$. On affecte **35**.
- S_3 restant : $90 - 35 = 55$. C_2 restant : $35 - 35 = 0$.
- La commande C_2 est satisfaite (0), on raye la colonne et on va à droite vers (S_3, C_3).

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60 (reste 0)
	50	**10**	0	0	
S_2	17	16	15	18	30 (reste 0)
	0	**30**	0	0	
S_3	19	21	20	22	90 (reste 55)
	0	**35**			
Commande	50 (reste 0)	75 (reste 0)	30	25	180

Itération 5 (Marron) et 6 (Bleu) :

- On continue en (S_3, C_3), affecte $\min(55, 30) = 30$. Reste 25 en S_3.
- Enfin en (S_3, C_4), affecte $\min(25, 25) = 25$.

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60 (reste 0)
	50	**10**	0	0	
S_2	17	16	15	18	30 (reste 0)
	0	**30**	0	0	
S_3	19	21	20	22	90 (reste 0)
	0	**35**	**30**	**25**	
Commande	50 (reste 0)	75 (reste 0)	30 (reste 0)	25 (reste 0)	180

La solution de base est : $x_{11} = 50, x_{12} = 10, x_{22} = 30, x_{32} = 35, x_{33} = 30, x_{34} = 25.$

Remarque. Cette méthode produit une solution valide mais **inefficace économiquement**, car elle n'optimise pas les coûts ; elle se contente d'écouler les stocks géographiquement.

1.1.2 Méthode de Balas-Hammer

Méthode. Principe et Algorithme
Contrairement au "Coin Nord-Ouest" qui est aveugle aux prix, cette méthode est **orientée vers les coûts**. Son objectif est de minimiser le risque de devoir utiliser des routes chères plus tard.

1. **Calcul des pénalités (Δ)**
   Toute la méthode repose sur le calcul d'une **pénalité** (ou différence) pour chaque ligne et chaque colonne encore actives.
   Cette pénalité représente le **surcoût** subi si l'on ne choisit pas la case la moins chère de la ligne ou de la colonne.
   - **Définition de Δ :** C'est la différence entre le **coût unitaire le plus faible** et le **deuxième coût unitaire le plus faible** d'une ligne ou d'une colonne.
2. **Sélection de la priorité**
   Repérez la ligne ou la colonne présentant la **pénalité Δ la plus élevée**.
   - *Interprétation :* c'est la ligne ou colonne pour laquelle une mauvaise décision serait la plus coûteuse.

     Règle de départage (en cas d'égalité) :
     Si plusieurs pénalités sont égales, on choisit la ligne ou la colonne dont la case la moins chère permet de transporter la **plus grande quantité possible** (offre ou demande maximale).
     Si l'égalité persiste, le choix est arbitraire.
3. **Allocation stratégique**
   Dans la ligne ou la colonne sélectionnée :
   - Identifiez la case ayant le **coût unitaire minimal**.
   - Allouez la **quantité maximale possible**, c'est-à-dire :

   $$\min(\text{offre restante}, \text{demande restante})$$
4. **Mise à jour et itération**
   - Soustrayez la quantité allouée des totaux correspondants.
   - Rayez la ligne si l'offre est épuisée, ou la colonne si la demande est satisfaite.
   - **Recommencez à l'étape 1** avec le tableau restant.

   Attention : les pénalités Δ doivent être **recalculées à chaque itération**, car les coûts encore disponibles évoluent.

Exemple. Trouver une proposition initiale au problème de transport précédent en appliquant la méthode de Balas-Hammer.

Solution.

Itération 1 (Magenta) :

- Calcul des pénalités (différence entre les 2 coûts les plus faibles).
- Max pénalité = 8 (Colonne C_4).
- Min coût dans C_4 = 10 (S_1).
- On affecte $\min(60, 25) = 25$ à (S_1, C_4). C_4 est saturée.

	C_1	C_2	C_3	C_4	Provisions	Pénalités
S_1	11	12	10	10	60 (reste 35)	0
				25		
S_2	17	16	15	18	30	1
				0		
S_3	19	21	20	22	90	1
				0		
Commande	50	75	30	25 (reste 0)	180	
Pénalités	6	4	5	$\boxed{8}$		

Itération 2 (Vert) :

- On recalcule les pénalités sans C_4.
- Max pénalité = 6 (Colonne C_1).
- Min coût dans C_1 = 11 (S_1).
- On affecte $\min(35, 50) = 35$ à (S_1, C_1). S_1 est saturée.

	C_1	C_2	C_3	C_4	Provisions	Pénalités
S_1	11	12	10	10	60 (reste 0)	0 1
	35	0	0	**25**		
S_2	17	16	15	18	30	1 1
				0		
S_3	19	21	20	22	90	1 1
				0		
Commande	50 (reste 15)	75	30	25 (reste 0)	180	
Pénalités	6 $\boxed{6}$	4 4	5 5	$\boxed{8}$ -		

Itération 3 (Orange) et Fin :

- On recalcule les pénalités sans S_1 et C_4.
- Max pénalité = 5 (Colonne C_3, arbitrage sur C_2). Choix C_3.
- Min coût dans C_3 = 15 (S_2).
- On affecte $\min(30, 30) = 30$ à (S_2, C_3). S_2 et C_3 saturées.
- Reste S_3 pour remplir le reste (C_1 et C_2).

	C_1	C_2	C_3	C_4	**Provisions**	**Pénalités**
S_1	11	12	10	10	60 (reste 0)	0 1 -
	35	0	0	**25**		
S_2	17	16	15	18	30 (reste 0)	1 1 1
	0	0	**30**	0		
S_3	19	21	20	22	90 (reste 0)	1 1 1
	15	**75**	0	0		
Commande	50 (reste 0)	75 (reste 0)	30 (reste 0)	25 (reste 0)	180	
Pénalités	6 $\boxed{6}$ 2	4 4 5	5 5 $\boxed{5}$	$\boxed{8}$ - -		

Remarque. Cette méthode demande plus de calculs que le Coin Nord-Ouest, mais elle produit une solution de départ beaucoup plus économique (souvent proche de la solution optimale), car elle évite "intelligemment" les routes les plus coûteuses.

1.1.3 Un cas particulier...

Si vous êtes observateur, alors vous aurez sûrement remarqué que dans le problème que l'on traite en exemple depuis le début les provisions et les commandes sont strictement les mêmes. Or, il est tout à fait possible que l'on ait plus de stock que de demandes (le cas inverse, demande $>$ stock, n'a bien entendu aucune solution possible) : dans ce cas nous rajoutons une nouvelle colonne (un nouveau client fictif) avec des coûts de transports à 0 qui va servir de "poubelle" pour les produits restants. On traitera cette colonne de la même manière que les autres quel que soit l'algorithme d'initialisation choisi.

1.2 Optimiser la proposition initiale

Nous avons désormais notre première solution, mais comment savoir si c'est la meilleure ? Et sinon, comment l'optimiser ? Pour cela, nous allons à chaque fois réaliser les étapes suivantes de façon un peu mécanique. Il s'agit de la méthode du marche-pied.

1.2.1 Graphe associé et proposition non-dégénérée

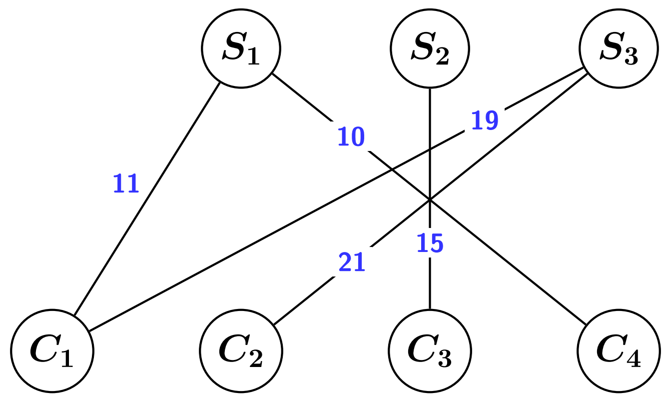
Pour appliquer la méthode du Marche-Pied, il faut changer de perspective : on passe d'une vision "Tableau" (lignes/colonnes) à une vision "Réseau" (Graphe). Voici comment transformer la proposition de transport actuelle en graphe :

Méthode. Construction du graphe associé à une proposition

- Les **Nœuds** (Sommets) : Le graphe est "bipartite". En haut, on aligne tous les points de départ (Sources, S_i) et en bas, tous les points d'arrivée (Destinations, L_j).
- Les **Arcs** (Liaisons) : On trace un trait (un arc) entre une Source et une Destination uniquement si la case correspondante dans le tableau est remplie (c'est-à-dire si la quantité transportée $x_{ij} > 0$).
- Les **Cases Vides** : Les cases où la quantité est 0 ne génèrent pas d'arc. Ce sont des routes inactives.
- Les **Poids** : Les poids des arcs sont les coûts de transport associés à chaque route.

Exemple. Dessiner le graphe associé à la proposition initiale précédente réalisée avec la méthode de Balas-Hammer.

Solution.



Une fois le graphe tracé, il faut vérifier s'il est techniquement apte à subir l'optimisation. C'est ici qu'intervient la notion de dégénérescence.

Méthode. Condition de non-dégénérescence
Pour que la méthode du Marche-Pied fonctionne, le réseau doit être **connexe** (tous les nœuds doivent être reliés entre eux, directement ou indirectement, sans former d'îlots isolés) et **ne pas contenir de cycle inutile**.

Pour vérifier cela sans dessiner, on utilise une règle simple. On compte le nombre de cases remplies (nombre d'arcs) dans notre solution actuelle.

- Soit m le nombre de lignes (Sources).
- Soit n le nombre de colonnes (Destinations).

La condition de **non-dégénérescence** est :

$$\text{Nombre d'arcs (cases remplies)} \geq m + n - 1$$

- Si la condition est respectée : Le graphe est non-dégénéré. Il est sain, connexe, et prêt pour l'optimisation.
- Sinon (Nombre d'arcs $< m + n - 1$) : Le graphe est dégénéré. Il manque des liaisons. Le réseau est morcelé en plusieurs parties isolées. On ne peut pas passer d'un point à un autre pour réajuster les flux.

Si le graphe est dégénéré (il manque des arcs pour atteindre le chiffre magique $m + n - 1$), on doit le "réparer" artificiellement avant de commencer l'optimisation. La solution : l'ajout d'un epsilon (ϵ), il faut créer une liaison artificielle pour reconnecter le réseau.

Méthode. Traitement des propositions dégénérées
Deux causes possibles de dégénérescence peuvent être identifiées après avoir tracé le graphe.

1. **Le graphe contient un ou plusieurs cycles**

   Dans ce cas, le graphe possède au moins $|V|$ arêtes. On applique alors la **méthode standard de suppression de cycle** (marche-pied sur le cycle), ce qui permet :

   - de rendre le graphe **acyclique**,
   - tout en **diminuant le coût total de transport**.

2. **Le graphe est acyclique mais non connexe**

   Le graphe contient strictement moins de $|V| - 1$ arêtes. Plus précisément, s'il contient $|V| - p$ arêtes avec $p > 1$, il possède p composantes connexes.

   Pour pouvoir appliquer la méthode du marche-pied, il faut transformer ce graphe en un **graphe maximalement acyclique**, c'est-à-dire :

   un graphe connexe, sans cycle, comportant exactement $|V| - 1$ arêtes.

   Choix des arêtes à ajouter

   - Il faut ajouter exactement $p - 1$ arêtes.
   - Ces arêtes sont choisies parmi les **cases vides du tableau de transport**.
   - On les examine par **ordre croissant de coût de transport**.
   - Une arête est retenue uniquement si son ajout **ne crée aucun cycle**.

Valeur attribuée à une arête ajoutée

À l'arête choisie, on attribue une quantité transportée égale à **0** (ou ϵ). Cette arête devient alors une **variable de base**, ce qui permet :

- de compléter la structure du graphe,
- sans modifier ni les coûts, ni les contraintes d'offre et de demande.

Une fois toutes les composantes reliées de cette manière, le graphe obtenu est un **arbre**. La méthode du marche-pied peut alors être appliquée normalement.

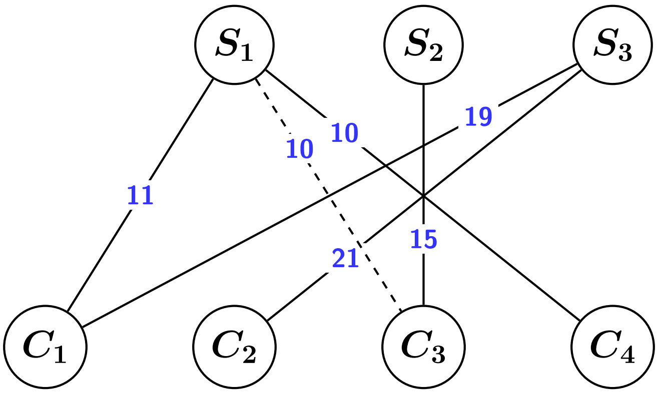
Exemple. Le graphe précédemment construit est-il dégénéré ? Si oui, le rendre non-dégénéré.

Solution.

- On compte les sommets : $|V| = 3$ sources $+$ 4 destinations $= 7$.
- On compte les arêtes (cases de base) : il y en a 5.
- Un arbre sur 7 sommets doit avoir $|V| - 1 = 6$ arêtes : il **manque donc 1 arête**.
- On observe en plus que le graphe n'est pas connexe : le couple (S_2, C_3) forme un îlot séparé.

Le graphe est donc **dégénéré**.

Pour connecter les composantes **sans créer de cycle**, on ajoute une arête entre cet îlot et l'autre composante. La quantité affectée à cette nouvelle arête est 0 **(ou ϵ)** : elle devient une variable de base, sans modifier le coût ni les contraintes.



1.2.2 Calcul des potentiels

Nous avons désormais une proposition non-dégénérée, on peut calculer les potentiels associés à chaque sommet. L'idée centrale consiste à interpréter chaque coût unitaire c_{ij} comme une **différence de potentiel** entre une source S_i et une cible C_j. On associe alors :

- un potentiel $E(S_i)$ à chaque sommet source S_i,
- un potentiel $E(C_j)$ à chaque sommet cible C_j.

Méthode. Calcul des potentiels via un système linéaire

Pour chaque arête (S_i, C_j) appartenant à la base (c'est-à-dire correspondant à une variable de transport strictement positive) avec un coût de transport c_{ij}, on impose la relation linéaire :

$$E(S_i) + E(C_j) = c_{ij}.$$

Cette équation traduit le fait que le coût de transport sur une arête de base est exactement compensé par la différence de potentiel entre ses deux sommets.

Structure du système linéaire

Comme la solution est non dégénérée, le nombre d'arêtes de base est égal à $n + p - 1$. On obtient donc un système linéaire de $n + p - 1$ équations portant sur $n + p$ inconnues :

$$\{E(S_1), \ldots, E(S_n), E(C_1), \ldots, E(C_p)\}.$$

En l'état, ce système linéaire est indéterminé : il y a autant d'équations que d'inconnues.

Fixation d'un potentiel de référence
Pour lever cette indétermination, on fixe arbitrairement le potentiel d'un sommet à 0, par exemple :

$$E(S_1) = 0.$$

Cette convention ne modifie pas les différences de potentiel et permet d'obtenir une solution unique pour l'ensemble des autres potentiels.
Astuce : on choisit généralement le sommet qui possède le plus d'arêtes entrantes ou sortantes

Il suffit désormais de résoudre ce système linéaire pour obtenir les potentiels.

Exemple. Calculer les potentiels associés au graphe précédemment construit.

Solution. On remarque que S_1 possède le plus d'arêtes sortantes, donc on pose $E(S_1) = 0$ et on résout :

$$\begin{cases} E(S_1) - E(C_1) = 11 \\ E(S_1) - E(C_3) = 10 \\ E(S_1) - E(C_4) = 10 \\ E(S_2) - E(C_3) = 15 \\ E(S_3) - E(C_1) = 19 \\ E(S_3) - E(C_2) = 21 \\ E(S_1) = 0 \end{cases} \iff \begin{cases} E(S_1) = 0 \\ E(S_2) = 5 \\ E(S_3) = 8 \\ E(C_1) = -11 \\ E(C_2) = -13 \\ E(C_3) = -10 \\ E(C_4) = -10 \end{cases}$$

1.2.3 Matrices des coûts potentiels et marginaux

Maintenant que nous avons les potentiels, nous pouvons calculer les coûts potentiels et marginaux.

Méthode. Calcul des matrices des coûts potentiels et marginaux
1. Matrice des coûts potentiels
La matrice des coûts potentiels représente le coût théorique induit uniquement par les potentiels. Pour toute paire (i, j), on définit le coût potentiel par :

$$p(i, j) = E(S_i) - E(C_j).$$

En calculant cette quantité pour toutes les sources et toutes les cibles, on obtient une matrice de même dimension que la matrice des coûts initiaux.

2. Matrice des coûts marginaux
La matrice des coûts marginaux mesure l'écart entre le coût réel et le coût potentiel. Pour toute case (i, j), on définit le coût marginal par :

$$m(i, j) = c(i, j) - p(i, j)$$

Où $c(i, j)$ est le coût réel de la case (i, j) (celui que l'on peut lire dans le tableau initial).

Exemple. Calculer les coûts potentiels et marginaux associés au graphe précédemment construit.

Solution. Calculons le coût potentiel de la case (S_1, C_1) :

$$p(S_1, C_1) = E(S_1) - E(C_1) = 0 - (-11) = 11$$

Le coût marginal de cette même case est donné par :

$$m(S_1, C_1) = c(S_1, C_1) - p(S_1, C_1) = 11 - 11 = 0$$

On fait de même pour toutes les cases et on obtient les matrices suivantes :

Coûts potentiels

	C_1	C_2	C_3	C_4
S_1	11	13	10	10
S_2	16	18	15	15
S_3	19	21	18	18

Coûts marginaux

	C_1	C_2	C_3	C_4
S_1	0	-1	0	0
S_2	1	-2	0	3
S_3	0	0	2	4

1.2.4 Fin ou bouclage

Fini les calculs ! Ou pas ? C'est le moment de vérifier si la proposition est optimale. Pour cela, nous regardons la matrice des coûts marginaux.

Méthode. Critère d'optimalité
Pour savoir si une proposition est optimale, on applique le critère suivant :

- Si **au moins un** coût marginal est **négatif**, alors x **n'est pas optimale** : il existe une amélioration possible en ajoutant une arête de coût marginal négatif.

Exemple. Vérifier si la proposition est optimale.

Solution. On remarque que plusieurs coûts marginaux sont négatifs :

$$m(S_1, C_2) = -1, \quad m(S_2, C_2) = -2$$

Donc la proposition n'est pas optimale.

Méthode. Optimisation d'une proposition

1. **Choisir l'arête à ajouter**
   - S'il existe **un seul** coût marginal négatif, on choisit naturellement l'arête correspondante.
   - S'il y en a **plusieurs**, on choisit **celui qui est le plus négatif** (le plus petit coût marginal).

   Cette arête est celle qui permet la **plus forte diminution du coût total**.
2. **Ajouter l'arête au graphe**
   - On ajoute au graphe l'arête correspondant au coût marginal choisi.
   - Comme le graphe initial est un arbre, cet ajout crée **un unique cycle**.
   - Ce cycle est appelé **cycle du marche-pied**.
3. **Modifier les quantités sur le cycle**
   La méthode détaillée pour cette étape est disponible dans la remarque ci-dessous.
4. **Obtenir une nouvelle proposition de transport**
   - L'arête ajoutée devient alors **utilisée**.
   - Une arête du cycle (celle dont la quantité devient nulle) est **retirée**.
   - Le graphe reste un arbre : la proposition est à nouveau **non dégénérée**.
   - Le coût total de transport est **strictement plus faible**.
5. **Relancer le processus**
   À partir de cette nouvelle proposition de transport, on recommence les étapes :
   - calcul des potentiels,
   - calcul des coûts marginaux,
   - test d'optimalité.

   Le processus est répété jusqu'à ce que **tous les coûts marginaux soient positifs**, ce qui garantit que la proposition finale est optimale.

Remarque. Comment modifier les quantités sur le cycle du marche-pied

1. **Identifier les arêtes du cycle**
   - Le cycle est constitué de :
     - l'arête nouvellement ajoutée,
     - et des arêtes déjà utilisées reliant ses extrémités.
   - Toutes les autres arêtes du graphe ne sont **pas modifiées**.
2. **Attribuer les signes sur le cycle**
   - On commence par placer un signe $+$ sur l'arête ajoutée.
   - En parcourant le cycle, on alterne ensuite les signes : $+$, $-$, $+$, $-$, . . .
   - Les arêtes marquées $+$ verront leur quantité **augmenter**.

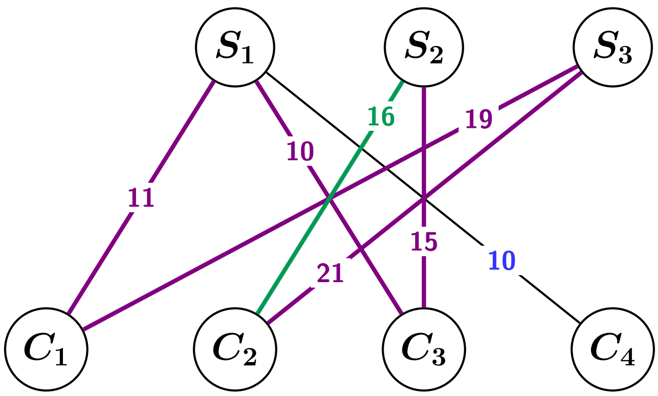
- Les arêtes marquées − verront leur quantité **diminuer**.

3. **Déterminer la quantité maximale transférable**
   - On cherche la plus grande quantité que l'on peut déplacer le long du cycle sans rendre une quantité transportée négative.
   - Cette limite est imposée par les arêtes marquées −.
   - On regarde donc les quantités actuellement transportées sur ces arêtes.
   - La plus petite de ces quantités fixe la **quantité maximale transférable**.

   Intuitivement : on ne peut pas retirer plus que ce qui est déjà transporté.
4. **Mettre à jour les quantités**
   - On ajoute cette quantité sur toutes les arêtes marquées +.
   - On retranche la même quantité sur toutes les arêtes marquées −.
   - Toutes les contraintes d'offre et de demande restent satisfaites : ce que l'on enlève à un sommet est exactement compensé ailleurs sur le cycle.
5. **Mettre à jour la structure du graphe**
   - Au moins une arête marquée − voit sa quantité devenir nulle.
   - Cette arête est retirée de la base (elle disparaît du graphe).
   - L'arête ajoutée devient effectivement utilisée.
   - Le graphe redevient un **arbre**.

Exemple. Donner une nouvelle proposition de transport.

Solution. Plusieurs coûts marginaux sont négatifs, celui de l'arête (S_2, C_2) est le plus petit (-2). On rajoute cette arête au graphe.



Cela crée le cycle $S_2C_2S_3C_1S_1C_3S_2$.
Détail des calculs sur le cycle :

- On attribue le signe **(+)** à la nouvelle arête (S_2, C_2).
- On parcourt le cycle en alternant les signes $+/-$:

$$(S_2, C_2)^+ \to (C_2, S_3)^- \to (S_3, C_1)^+ \to (C_1, S_1)^- \to (S_1, C_3)^+ \to (C_3, S_2)^-$$

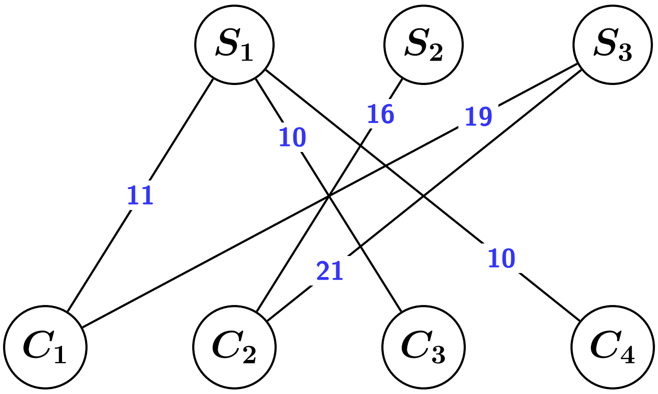
- Les quantités sur les arêtes marquées **(-)** sont : 30 pour (S_2, C_3), 35 pour (S_1, C_1) et 75 pour (S_3, C_2).
- La quantité maximale transférable est déterminée par la plus petite de ces valeurs (pour ne pas avoir de quantité négative) :

$$\theta = \min(30, 35, 75) = 30$$

On applique cette modification $\theta = 30$ (voir tableau ci-dessus avec les ajouts en vert et retraits en rouge). L'arête (S_2, C_3) tombe à 0 et sort de la base.
On a donc la nouvelle proposition de transport :

	C_1	C_2	C_3	C_4	**Provisions**
S_1	11	12	10	10	60
	35 -30	0	0 +30	25 (Pas dans cycle)	
S_2	17	16	15	18	30
	0	0 +30	30 -30	0	
S_3	19	21	20	22	90
	15 +30	75 -30	0	0	
Commande	50	75	30	25	180

Graphe de la nouvelle proposition :



Désormais, nous pouvons appliquer les étapes de cette section 1.2 pour savoir si la nouvelle proposition est optimale.

2 Programmation linéaire et la méthode du simplexe

La **programmation linéaire** regroupe des problèmes d'optimisation dans lesquels on cherche à **maximiser ou minimiser une quantité** tout en respectant un ensemble de **contraintes linéaires**.
Un programme linéaire est construit à partir de :

- variables qui représentent les décisions à prendre ;
- une **fonction objectif**, qui indique ce que l'on cherche à optimiser ;
- des **contraintes**, qui limitent les valeurs possibles des variables.

De manière générale, un programme linéaire s'écrit sous la forme suivante :

$$\begin{array}{rl} \texttt{maximiser} & \text{fonction objectif} \\[1em] \texttt{sous les contraintes} & \begin{cases} \text{contraintes linéaires (équations),} \\ \text{contraintes de positivité des variables.} \end{cases} \end{array}$$

Le but est de trouver une solution qui respecte toutes les contraintes et qui donne la **meilleure valeur possible de la fonction objectif**. Pour résoudre ce type de problème, on utilisera la **méthode du simplexe**.

2.1 Traduire un problème en un programme linéaire

Méthode.
La première étape pour résoudre un problème par programmation linéaire consiste à **le modéliser correctement**. Cette modélisation se fait toujours selon les étapes suivantes.

1. Identifier les variables de décision

- Les variables représentent les **quantités inconnues** que l'on cherche à déterminer.
- Elles correspondent aux **choix possibles** dans le problème (produire, transporter, affecter, utiliser, etc.).
- Chaque variable doit être clairement définie et interprétable.

2. Définir la fonction objectif

- La fonction objectif traduit **le but du problème**.
- Elle est toujours une **combinaison linéaire des variables**.
- On précise s'il s'agit de **maximiser** (profit, rendement, production) ou de **minimiser** (coût, temps, perte).
- Les coefficients de la fonction objectif représentent l'impact de chaque variable sur le résultat final.

3. Écrire les contraintes
Les contraintes traduisent les **limitations du problème**. Elles sont toutes linéaires.

- **Contraintes de ressources** : capacités, budgets, temps disponibles.
- **Contraintes de demande ou d'objectif** : quantités minimales ou maximales à atteindre.
- **Contraintes d'équilibre** : conservation, répartition ou affectation des quantités.

Chaque contrainte relie les variables par une équation ou une inéquation linéaire.

4. Ajouter les contraintes de positivité

- Les variables représentent généralement des quantités physiques.
- Elles doivent donc satisfaire des contraintes de **non-négativité**.
- On impose systématiquement :

$$x_i \geq 0 \quad \text{pour toute variable } x_i.$$

5. Vérifier la cohérence du modèle

- Toutes les données du problème doivent apparaître dans le modèle.

- Chaque variable doit intervenir dans la fonction objectif et/ou dans les contraintes.
- Le sens des inégalités doit être cohérent avec l'interprétation du problème.

Une fois ces étapes terminées, le problème est correctement formulé sous la forme d'un **programme linéaire** et peut être résolu par la méthode du simplexe.

Exemple. Le directeur de production d'une usine de produits chimiques veut estimer le nombre de travailleurs dont il a besoin pour faire fonctionner son usine à tout moment. Chaque jour de la semaine est divisé en trois périodes de huit heures (de minuit à 8h, de 8h à 16h et de 16h à minuit), désignées respectivement par nuit, jour et soirée. L'usine doit être occupée à tout moment et le nombre minimal de travailleurs requis pour chacune de ces tranches horaires est indiqué ci-dessous :

	Lun	Mar	Merc	Jeu	Vend	Sam	Dim
Nuit	5	3	2	4	3	2	2
Jour	7	8	9	5	7	2	5
Soirée	9	10	10	7	11	2	2

De plus, la convention collective stipule que :

- Aucun employé ne peut travailler plus qu'un créneau par jour.
- Chaque employé doit travailler au même créneau pendant la semaine.
- Chaque employé doit travailler quatre jours consécutifs dans une période de sept jours.

Formuler le problème comme un programme linéaire.

Solution. Il y a 7 jours divisés en 3 tranches horaires $\rightarrow$ 21 tranches. Sur chaque tranche, on décide de combien de personnes travaillent $\Longrightarrow$ **21 variables.**
On définit les variables comme le nombre de travailleurs **commençant** leur cycle de 4 jours le jour J pour la tranche T :

$$\begin{cases} x_{LN} \text{ : travailleurs commençant Lundi Nuit} \\ x_{LJ} \text{ : travailleurs commençant Lundi Jour} \\ x_{LS} \text{ : travailleurs commençant Lundi Soir} \\ \vdots \\ x_{DJ} \text{ : travailleurs commençant Dimanche Jour} \\ x_{DS} \text{ : travailleurs commençant Dimanche Soir} \end{cases}$$

Contraintes : Le tableau indique les quantités minimales à respecter. La convention nous simplifie les calculs :
$\rightarrow$ Pour chaque créneau il n'y aura que les travailleurs qui sont dans leurs 4 jours consécutifs sur la même plage horaire.
Ainsi, la contrainte du **Lundi Soir** (besoin ≥ 9) implique ceux qui ont commencé Vendredi (travaillent V, S, D, **L**), Samedi (S, D, **L**, M), Dimanche (D, **L**, M, M) ou Lundi (**L**, M, M, J). On écrit : $x_{VS}+x_{SS}+x_{DS}+x_{LS} \geq 9$.
On a donc **21 contraintes** au total :

$$\begin{cases} x_{LN}+x_{DN}+x_{SN}+x_{VN} \geq 5 \quad \text{(Lundi Nuit)} \\ x_{LS}+x_{DS}+x_{SS}+x_{VS} \geq 9 \quad \text{(Lundi Soir)} \\ \vdots \\ x_{DS}+x_{SS}+x_{VS}+x_{JS} \geq 2 \quad \text{(Dimanche Soir)} \\ x_{DN}+x_{SN}+x_{VN}+x_{JN} \geq 2 \quad \text{(Dimanche Nuit)} \end{cases}$$

Objectif : L'objectif est de minimiser le nombre total de travailleurs embauchés, donc la somme de toutes les variables (ceux qui commencent lundi + ceux qui commencent mardi + ...).

$$\boxed{\min f = \sum x_{ij} = x_{LS}+x_{LJ}+x_{LN}+\cdots+x_{DS}+x_{DN}}$$

2.2 Résoudre graphiquement un programme linéaire

La résolution graphique est une méthode intuitive permettant de résoudre un **programme linéaire à deux variables**. Elle repose sur l'interprétation géométrique des contraintes et de la fonction objectif dans le plan.

Méthode.

1. **Vérifier que la méthode est applicable**
   - La méthode graphique n'est possible que si le programme comporte **deux variables**.
   - Chaque solution correspond alors à un point du plan.
2. **Tracer les contraintes**
   Pour chaque contrainte linéaire :
   - On commence par remplacer l'inégalité par une **égalité**.
   - On trace la droite correspondante dans le plan.
   - On détermine ensuite le **demi-plan autorisé** à l'aide d'un point test (par exemple l'origine).

   Conseil important : Il est fortement recommandé d'utiliser **une couleur différente par contrainte** afin de distinguer clairement les différents demi-plans.
3. **Déterminer la région admissible**
   - La région admissible est l'**intersection de tous les demi-plans** définis par les contraintes.
   - Elle correspond à l'ensemble des solutions possibles du programme linéaire.
   - Les contraintes de positivité limitent généralement la région au premier quadrant.
4. **Déterminer les sommets de la région admissible**
   - Les sommets de la région admissible sont les **points d'intersection** entre deux droites issues des contraintes.
   - On calcule leurs coordonnées en **résolvant les systèmes de deux équations** correspondant aux contraintes concernées si l'on ne peut pas les repérer graphiquement.
   - On ne conserve que les points qui **satisfont toutes les contraintes** du problème.
5. **Évaluer la fonction objectif aux sommets**
   - On calcule la valeur de la fonction objectif en chacun des sommets admissibles.
   - On compare ces valeurs entre elles.
   - La valeur optimale (minimale ou maximale selon le problème) est atteinte en l'un de ces sommets.
6. **Interpréter la solution**
   - Les coordonnées du sommet optimal donnent les valeurs des variables du problème.
   - La valeur de la fonction objectif en ce point est la solution recherchée.
   - D'un point de vue géométrique, ceci illustre que l'optimum d'un programme linéaire est toujours atteint en un **sommet** de la région admissible.
   - Ces valeurs doivent enfin être interprétées dans le contexte du problème initial.

Exemple. On considère le programme linéaire suivant :

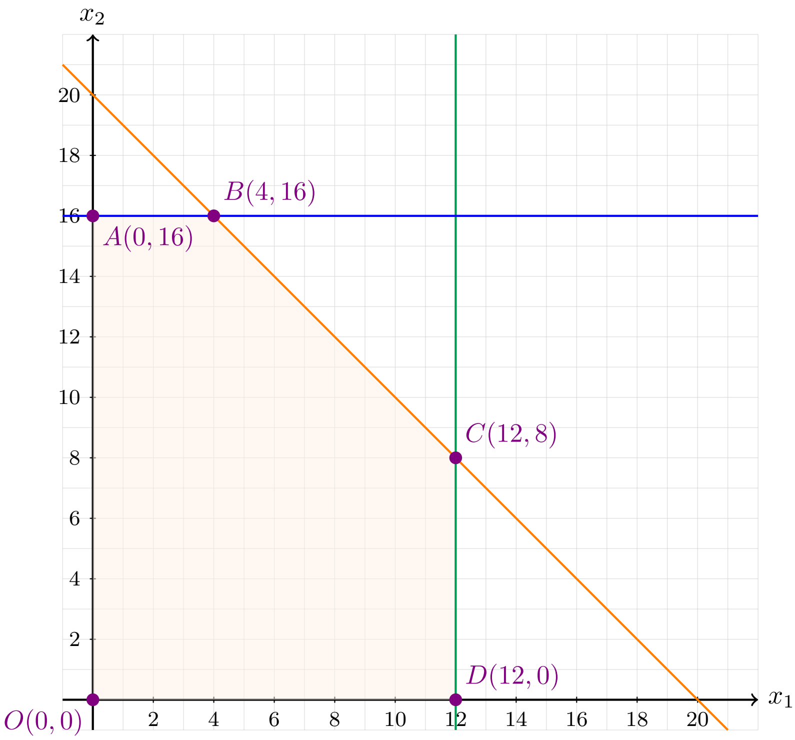
$$\begin{array}{rl} \texttt{maximiser} & 18x_1 + \dfrac{25}{2}x_2 \\ \texttt{sous les contraintes} & \begin{cases} x_1 + x_2 \leq 20 \\ 0 \leq x_1 \leq 12 \\ 0 \leq x_2 \leq 16 \end{cases} \end{array}$$

Résoudre graphiquement ce problème.

Solution.
Nous allons d'abord tracer les 3 contraintes

- $x_1 + x_2 \leq 20$
- $0 \leq x_1 \leq 12$

- $0 \le x_2 \le 16$



Détermination des sommets admissibles
La région réalisable est le polygone orange. Ses sommets (déjà repérés sur le graphique) sont :

$$O(0,0),\quad A(0,16),\quad B(4,16),\quad C(12,8),\quad D(12,0).$$

Calcul de la fonction objectif sur chaque sommet
On note

$$Z(x_1,x_2) = 18x_1 + \frac{25}{2}x_2.$$

Sommet	$Z = 18x_1 + \frac{25}{2}x_2$
$O(0,0)$	$18 \cdot 0 + \frac{25}{2} \cdot 0 = 0$
$A(0,16)$	$18 \cdot 0 + \frac{25}{2} \cdot 16 = 200$
$B(4,16)$	$18 \cdot 4 + \frac{25}{2} \cdot 16 = 72 + 200 = 272$
$C(12,8)$	$18 \cdot 12 + \frac{25}{2} \cdot 8 = 216 + 100 = 316$
$D(12,0)$	$18 \cdot 12 + \frac{25}{2} \cdot 0 = 216$

La valeur maximale est donc :

$$Z_{\max} = 316,$$

atteinte au sommet

$$(x_1^\star, x_2^\star) = C(12,8).$$

Solution optimale : $\boxed{x_1 = 12,\ x_2 = 8}$ et $\boxed{Z_{\max} = 316}$.

2.3 Utiliser la méthode du simplexe

Tous ces dessins sont très jolis mais bon... lorsque l'on dépasse 2 variables cela devient difficile à représenter graphiquement ! Pour cela, nous allons utiliser la méthode du simplexe.

C'est une méthode itérative permettant de résoudre un **programme linéaire** en se déplaçant de **sommet en sommet** du polyèdre des contraintes, tout en améliorant la valeur de la fonction objectif à chaque étape.

2.3.1 Passage à la forme standard

La première étape de l'algorithme du simplexe est de passer le programme linéaire sous une forme standard. L'objectif est de réécrire n'importe quel programme linéaire sous une forme utilisable par le simplexe, c'est-à-dire :

$$\text{maximiser une fonction linéaire} \quad \text{sous } AX = B \text{ et } X \geq 0.$$

Méthode. Passer un programme linéaire en forme standard

1. **Mettre le problème sous forme "maximiser"**

   - Si le problème est un **minimum**, on remplace la fonction objectif f par $-f$:

   $$\min f \quad \Longleftrightarrow \quad \max(-f).$$

   - Les contraintes ne changent pas.

2. **Rendre toutes les variables *positives***

   - La forme standard impose $X \geq 0$.
   - Si une variable est déjà contrainte par $x \geq 0$, on ne fait rien.
   - Si une variable est **libre** (pas de contrainte de signe), on la remplace par la différence de deux variables positives :

   $$x = x^+ - x^- \qquad \text{avec } x^+ \geq 0,\ x^- \geq 0.$$

   - Si une variable est de signe négatif (par exemple $x \leq 0$), on pose $x = -z$ avec $z \geq 0$.

3. **Transformer toutes les inéquations en équations (variables d'écart)**

   - Pour une contrainte du type $\sum a_k x_k \leq b$, on ajoute une **variable d'écart** $s \geq 0$:

   $$\sum a_k x_k + s = b, \qquad s \geq 0.$$

   - Pour une contrainte du type $\sum a_k x_k \geq b$, on soustrait une variable d'écart $s \geq 0$:

   $$\sum a_k x_k - s = b, \qquad s \geq 0.$$

   - Après cette étape, **toutes les contraintes** sont des **équations**.

4. **Faire attention au second membre**

   - Dans la forme standard utilisée par le simplexe, on veut typiquement un second membre B (les constantes à droite) clair et stable.
   - Si une équation a un second membre négatif (par exemple $b < 0$), on peut multiplier toute l'équation par -1 (cela inverse les signes de tous les coefficients).
   - **Remarque importante :** le fait d'avoir un $b < 0$ peut empêcher le point de base "évident" d'être admissible. L'initialisation du simplexe traitera ce problème plus tard (variables artificielles).

5. **Écrire le résultat final sous la forme standard**

   - On regroupe toutes les variables (variables d'origine, variables issues des décompositions, variables d'écart) dans un grand vecteur X.
   - On obtient finalement :

   $$\text{maximiser } C^\top X \quad \text{sous } AX = B \quad \text{et } X \geq 0.$$

6. **Contrôle final (à toujours faire)**

   - Vérifier que la fonction objectif est bien une **maximisation**.
   - Vérifier que toutes les contraintes sont des **équations**.
   - Vérifier que **toutes les variables** apparaissent avec une contrainte de type ≥ 0.
   - Vérifier que chaque variable ajoutée (écart, décomposition) a un sens : elle sert uniquement à mettre le problème dans la forme requise.

Exemple. On considère le programme linéaire suivant :

$$\begin{array}{rl} \texttt{maximiser} & 18x_1 + \dfrac{25}{2}x_2 \\ \texttt{sous les contraintes} & \begin{cases} x_1 + x_2 \leq 20 \\ 0 \leq x_1 \leq 12 \\ 0 \leq x_2 \leq 16 \end{cases} \end{array}$$

Mettre le programme linéaire sous forme standard.

Solution. Les contraintes données sont :

$$\begin{cases} x_1 + x_2 \leq 20 \\ 0 \leq x_1 \leq 12 \\ 0 \leq x_2 \leq 16 \end{cases}$$

Les contraintes $0 \leq x_1$ et $0 \leq x_2$ sont déjà des **contraintes de positivité**. Il reste donc à traiter les **bornes supérieures** :

$$x_1 \leq 12 \qquad \text{et} \qquad x_2 \leq 16.$$

Ajouter des variables d'écart pour transformer en équations
Pour toute contrainte de type $\leq$, on ajoute une variable d'écart **positive** :

- Pour $x_1 + x_2 \leq 20$, on introduit $y_1 \geq 0$:

$$x_1 + x_2 + y_1 = 20.$$

- Pour $x_1 \leq 12$, on introduit $y_2 \geq 0$:

$$x_1 + y_2 = 12.$$

- Pour $x_2 \leq 16$, on introduit $y_3 \geq 0$:

$$x_2 + y_3 = 16.$$

Forme standard finale
La fonction objectif ne change pas (c'est déjà une maximisation) et toutes les variables sont positives :

$$\begin{array}{rl} \texttt{maximiser} & f = 18x_1 + \dfrac{25}{2}x_2 \\ \texttt{sous les contraintes} & \begin{cases} x_1 + x_2 + y_1 = 20 \\ x_1 + y_2 = 12 \\ x_2 + y_3 = 16 \\ x_1, x_2, y_1, y_2, y_3 \geq 0. \end{cases} \end{array}$$

2.3.2 Construction du tableau et choix du pivot

On suppose dans ce qui suit que l'on dispose d'un **point de base admissible initial**. L'étape d'initialisation sera détaillée ultérieurement.
On peut appliquer l'algorithme du simplexe à l'aide d'un **tableau**. Cette méthode permet d'identifier, à chaque itération, quelle variable doit entrer dans la base et laquelle doit en sortir.

Méthode. Construire le tableau du simplexe et choisir le pivot

1. **Construire le tableau du simplexe**

   - Chaque **ligne** (sauf la dernière) correspond à une contrainte.
   - Chaque **colonne** correspond à une variable (variables de décision, variables d'écart, etc.).
   - La dernière colonne contient le **second membre** des contraintes.
   - La **dernière ligne** représente la fonction objectif, écrite en fonction des variables hors-base.

   Le tableau regroupe ainsi toutes les informations nécessaires pour effectuer une itération du simplexe.

2. **Choisir la variable entrante (colonne pivot)**

- On observe les coefficients de la **fonction objectif** dans la dernière ligne du tableau.
- Si tous ces coefficients sont **positifs ou nuls**, alors la solution actuelle est **optimale** et l'algorithme s'arrête.
- Sinon, on choisit la variable associée au **coefficient le plus négatif**.
- Cette variable est appelée **variable entrante** : l'augmenter permet d'améliorer la valeur de la fonction objectif.

3. **Choisir la variable sortante (ligne pivot)**

   - Pour chaque ligne où le coefficient de la variable entrante est **strictement positif**, on calcule le rapport :
$$\frac{\text{second membre}}{\text{coefficient de la variable entrante}}.$$
   - On conserve uniquement les rapports **positifs**.
   - Le **plus petit rapport strictement positif** détermine la ligne pivot.
   - La variable associée à cette ligne est la **variable sortante**.

4. **Identifier le pivot**

   - Le **pivot** est l'élément situé à l'intersection de la colonne de la variable entrante et de la ligne de la variable sortante.
   - Il sert de point de départ pour effectuer les opérations de Gauss.

Propriété. Règle de Bland
La **règle de Bland** est une règle de sélection utilisée dans l'algorithme du simplexe pour **éviter les cycles** (c'est-à-dire le retour indéfini aux mêmes tableaux sans amélioration).

Elle s'applique **uniquement en cas d'égalité** lors du choix :

- de la **variable entrante** (plusieurs coefficients identiques dans la fonction objectif),
- ou de la **variable sortante** (plusieurs rapports minimaux identiques).

La règle est la suivante :

- parmi les variables **candidates à entrer**, on choisit celle dont l'**indice est le plus petit** ;
- parmi les variables **candidates à sortir**, on choisit également celle dont l'**indice est le plus petit**.

Cette règle garantit que l'algorithme du simplexe **termine toujours** et empêche les boucles infinies, sans modifier le résultat final.

Exemple. On reprend le programme linéaire standardisé précédent :

$$\texttt{maximiser} \quad f = 18x_1 + \frac{25}{2}x_2$$
$$\texttt{sous les contraintes} \quad \begin{cases} x_1 + x_2 + y_1 = 20 \\ x_1 + y_2 = 12 \\ x_2 + y_3 = 16 \\ x_1, x_2, y_1, y_2, y_3 \geq 0. \end{cases}$$

Écrire le tableau associé et identifier le premier pivot.

Solution. Les variables de base initiales sont y_1, y_2, y_3. On écrit les contraintes sous forme matricielle et la fonction objectif sur la dernière ligne.

	x_1	x_2	y_1	y_2	y_3	f	B
y_1	1	1	1	0	0	0	20
y_2	1	0	0	1	0	0	12
y_3	0	1	0	0	1	0	16
f	$\boxed{-18}$	$-\dfrac{25}{2}$	0	0	0	1	0

Choix de la variable entrante
On observe la dernière ligne (fonction objectif) :

$$-18 \quad \text{et} \quad -\frac{25}{2}.$$

Le **coefficient le plus négatif** est $\boxed{-18}$, associé à la variable x_1.

$$\Rightarrow \quad \boxed{x_1 \text{ est la variable entrante}}$$

Choix de la variable sortante
On calcule les rapports $\dfrac{B_i}{a_{i,x_1}}$ pour les lignes où le coefficient de x_1 est strictement positif :

$$\frac{20}{1} = 20 \qquad \frac{12}{1} = 12$$

Le plus petit rapport strictement positif est 12, obtenu sur la ligne y_2.

$$\Rightarrow \quad \boxed{y_2 \text{ est la variable sortante}}$$

Pivot
Le pivot est l'élément situé à l'intersection entre la colonne x_1 et la ligne y_2 (il est **encerclé dans le tableau**). On peut maintenant effectuer le pivot de Gauss pour passer au tableau suivant.

2.3.3 Exécution du pivot de Gauss

Une fois le tableau du simplexe construit et le **pivot identifié**, l'objectif est de **mettre à jour le tableau** afin d'obtenir une **nouvelle base admissible** améliorant la valeur de la fonction objectif.

Méthode. Effectuer le pivot de Gauss dans le tableau du simplexe

1. **Identifier la ligne et la colonne pivot**
   - La **colonne pivot** correspond à la variable entrante.
   - La **ligne pivot** correspond à la variable sortante.
   - L'élément à l'intersection est appelé le **pivot**.

2. **Normaliser la ligne pivot**
   - On divise toute la ligne pivot par la valeur du pivot.
   - Après cette opération, le pivot vaut exactement 1.
   - Cette ligne exprime désormais la **nouvelle variable de base** en fonction des autres variables.

3. **Éliminer les autres coefficients de la colonne pivot**
   - Pour chaque autre ligne (y compris la ligne de la fonction objectif), on effectue une combinaison linéaire de lignes afin d'annuler le coefficient de la colonne pivot.
   - Concrètement, pour une ligne L_i, on applique :

   $$L_i \leftarrow L_i - a_i \times L_{\text{pivot}},$$

   où a_i est le coefficient de la colonne pivot dans la ligne L_i.
   - À la fin, la colonne pivot est une **colonne unité**.

4. **Obtenir le nouveau tableau**

- La variable entrante devient une **variable de base**.
- La variable sortante devient une **variable hors-base**.
- Le tableau obtenu correspond à un nouveau point de base admissible.

5. **Interprétation du résultat**

   - La valeur de la fonction objectif a **augmenté** (cas d'une maximisation).
   - Les valeurs des variables de base se lisent dans la colonne B.
   - Les variables hors-base sont fixées à 0.

6. **Décider de la suite de l'algorithme**

   - On examine à nouveau la dernière ligne du tableau.
   - Si tous les coefficients sont **positifs ou nuls**, la solution est **optimale** et l'algorithme s'arrête.
   - Sinon, on identifie un nouveau pivot et on **répète l'opération**.

Remarque. Chaque pivot de Gauss correspond à un déplacement le long d'une arête du polyèdre des contraintes vers un sommet donnant une meilleure valeur de la fonction objectif.

Exemple. Résoudre le problème précédent en poursuivant la méthode du simplexe.

Solution. **Pivot 1 : x_1 entre, y_2 sort**
Le pivot est déjà 1 (ligne y_2, colonne x_1), donc on élimine les autres coefficients de la colonne x_1 :

	x_1	x_2	y_1	y_2	y_3	f	B
y_1	1	1	1	0	0	0	20
y_2	1	0	0	1	0	0	12
y_3	0	1	0	0	1	0	16
f	$\boxed{-18}$	$-\frac{25}{2}$	0	0	0	1	0

$$L_1 \leftarrow L_1 - L_2 \qquad L_f \leftarrow L_f + 18L_2 \quad \longleftrightarrow$$

	x_1	x_2	y_1	y_2	y_3	f	B
y_1	0	1	1	-1	0	0	8
x_1	1	0	0	1	0	0	12
y_3	0	1	0	0	1	0	16
f	0	$-\frac{25}{2}$	0	18	0	1	216

On continue car la dernière ligne contient encore un coefficient négatif (ici $-\frac{25}{2}$).

Choix du pivot 2 : x_2 entre, test des rapports
La variable entrante est x_2 (coefficient le plus négatif sur la dernière ligne).
On calcule les rapports $\dfrac{B}{\text{coef de } x_2}$ uniquement sur les lignes où le coef de x_2 est > 0 :

$$\frac{8}{1} = 8 \qquad \text{et} \qquad \frac{16}{1} = 16.$$

Le plus petit est 8, donc y_1 **sort** et le **pivot** est le 1 de la ligne y_1, colonne x_2.

Pivot 2 : x_2 entre, y_1 sort
Le pivot vaut 1, donc on élimine les autres coefficients de la colonne x_2 :

	x_1	x_2	y_1	y_2	y_3	f	B
y_1	0	1	1	-1	0	0	8
x_1	1	0	0	1	0	0	12
y_3	0	1	0	0	1	0	16
f	0	$\boxed{-\frac{25}{2}}$	0	18	0	1	216

$$L_3 \leftarrow L_3 - L_1 \qquad L_f \leftarrow L_f + \frac{25}{2}L_1 \quad \longleftrightarrow$$

	x_1	x_2	y_1	y_2	y_3	f	B
x_2	0	1	1	-1	0	0	8
x_1	1	0	0	1	0	0	12
y_3	0	0	-1	1	1	0	8
f	0	0	$\frac{25}{2}$	$\frac{11}{2}$	0	1	316

Conclusion (optimalité et solution)
Dans la dernière ligne, les coefficients des variables de décision x_1 et x_2 sont nuls (et il n'y a plus de coefficient négatif à corriger) : on ne peut plus améliorer f. La solution est donc **optimale**.

On lit la solution en posant les variables hors-base à 0 :

$$x_1 = 12, \qquad x_2 = 8, \qquad y_3 = 8, \qquad y_1 = 0, \qquad y_2 = 0,$$

et la valeur optimale :

$$f_{\max} = 316.$$

2.3.4 Objectif non borné

Lors de l'application de la méthode du simplexe, il peut arriver que le **problème de programmation linéaire n'admette pas de solution optimale finie**. On dit alors que la **fonction objectif est non bornée**.

Méthode. Objectif non borné dans la méthode du simplexe

1. **Détection dans le tableau du simplexe**
   - On se place dans un problème de **maximisation**.
   - On identifie une **variable entrante** dont le coefficient dans la dernière ligne du tableau est **strictement négatif**.
   - On examine ensuite la colonne correspondante dans les contraintes.
2. **Absence de variable sortante**
   - Si **tous les coefficients de la colonne de la variable entrante sont inférieurs ou égaux à** 0, alors aucun rapport $\frac{B_i}{a_i}$ $(a_i > 0)$ ne peut être calculé.
   - Il est donc **impossible de choisir une variable sortante**.
3. **Interprétation géométrique**
   - La direction associée à la variable entrante permet d'augmenter indéfiniment la valeur de la fonction objectif.
   - La région admissible est ouverte dans cette direction.
   - Il n'existe donc **aucun sommet optimal**.
4. **Conclusion algorithmique**
   - La méthode du simplexe s'arrête immédiatement.
   - Le programme linéaire admet une **fonction objectif non bornée**.
   - On conclut que $f \longrightarrow +\infty$ dans le cas d'une maximisation.

Remarque. Dans un problème de **minimisation**, les signes sont inversés : une variable entrante est associée à un coefficient strictement positif dans la dernière ligne du tableau, et l'absence de coefficient strictement négatif dans la colonne correspondante conduit de la même manière à un objectif non borné.

Exemple. Nous résolvons un problème de maximisation. On considère le tableau du simplexe suivant :

	x_1	x_2	x_3	y_1	y_2	f	B
L_1	1	0	-1	$\frac{1}{4}$	$-\frac{1}{4}$	0	1
L_2	0	1	-1	$\frac{3}{4}$	$\frac{1}{4}$	0	6
f	0	0	-3	$\frac{3}{2}$	$\frac{1}{2}$	1	12

Que peut-on conclure ?

Solution. On se place dans un problème de **maximisation**. On observe la dernière ligne :

- Le coefficient associé à la variable x_3 est **strictement négatif** (il vaut -3).
- La variable x_3 est donc une **candidate naturelle pour entrer dans la base**.

Test de la colonne de la variable entrante

On examine maintenant la colonne correspondant à x_3 dans les contraintes :

$$\begin{cases} -1 \\ -1 \end{cases}$$

- Tous les coefficients de cette colonne sont **négatifs**.
- Il n'existe donc **aucune ligne** pour laquelle le coefficient de x_3 est strictement positif.
- Par conséquent, aucun rapport $\frac{B_i}{a_i}$ ne peut être calculé.

Conclusion
Il est impossible de choisir une variable sortante :

- on peut augmenter indéfiniment la variable x_3,
- tout en respectant les contraintes,
- et en augmentant strictement la valeur de la fonction objectif.

$$\boxed{\text{La fonction objectif est non bornée.}}$$

La méthode du simplexe s'arrête : le programme linéaire n'admet **pas de solution optimale finie**.

2.4 Initialiser l'algorithme du simplexe

Si vous vous souvenez bien, j'ai dit un peu plus haut que l'on considérait que le point de base $(0, 0)$ était admissible et que par conséquent on pouvait appliquer l'algorithme du simplexe. En réalité, ce n'est pas toujours le cas... Mais déjà ça veut dire quoi qu'un "point de base est admissible" ?

Définition. Point de base admissible
Dans le cadre de la méthode du simplexe, un **point de base** est une solution obtenue en :

- choisissant un ensemble de variables dites **de base**,
- fixant toutes les autres variables (dites **hors-base**) à 0,
- puis en calculant les valeurs des variables de base à l'aide des contraintes.

Un point de base est dit **admissible** si et seulement si :

- toutes les variables du problème sont **positives ou nulles**,
- en particulier, les variables de base obtenues sont toutes ≥ 0,
- les contraintes du programme linéaire sont satisfaites.

Autrement dit, un point de base admissible est :

- une solution du système de contraintes,
- qui respecte les contraintes de positivité,
- et qui correspond à un **sommet de la région admissible**.

Remarque. Géométriquement, un point de base admissible correspond à un **coin (sommet)** du polyèdre des contraintes. L'algorithme du simplexe se déplace ensuite de sommet en sommet, en cherchant à améliorer la valeur de la fonction objectif.

Autrement dit, si le point de base n'est pas un sommet de la région admissible, on utilise une procédure d'initialisation en **deux phases** pour trouver un sommet.

Méthode. Initialisation du simplexe (Phase I $\rightarrow$ Phase II)

1. **Constat : le point de base n'est pas admissible**
   - Après passage en forme standard (avec variables d'écart), on construit le tableau.
   - Si, en mettant les variables hors-base à 0, on obtient une valeur négative pour une variable de base (équivalent à un B négatif dans une ligne), alors **on ne peut pas démarrer le simplexe directement**.

- Ces lignes sont les **contraintes problématiques**.

2. **Construire un problème modifié P' (ajout de variables artificielles)**
   - Pour **chaque contrainte problématique**, on ajoute une nouvelle variable **artificielle** (souvent notée z_i ou x_{n+i}), avec la contrainte :
   $$z_i \geq 0.$$
   - L'objectif est que cette variable artificielle permette de former une **base initiale** et donc un **point de base admissible immédiat**.
   - Concrètement, on modifie uniquement les contraintes concernées pour que la variable artificielle apparaisse **comme variable de base** (colonne pivot).

3. **Définir la fonction auxiliaire (Phase I)**
   - On veut forcer les variables artificielles à devenir 0.
   - On définit donc une **fonction auxiliaire** :
   $$f_{\text{aux}} = -\sum z_i$$
   (maximiser f_{aux} revient à **minimiser** la somme des z_i).
   - Avant de lancer le simplexe, on doit écrire la dernière ligne du tableau (celle de f_{aux}) **en termes des variables hors-base**.
   Idée : on élimine dans la ligne de f_{aux} les variables de base (les z_i), en combinant les lignes de contraintes, afin d'obtenir une vraie ligne de tableau exploitable.

4. **Résoudre le problème auxiliaire avec le simplexe (Phase I)**
   - On applique la méthode du simplexe **normalement** sur le tableau de Phase I.
   - On s'arrête quand on ne peut plus améliorer f_{aux}.

5. **Conclusion de Phase I : deux cas**
   - **Cas 1 :** $f_{\text{aux}}^{\max} \neq 0$ (donc strictement négatif en pratique)
     - Cela signifie que **au moins une variable artificielle reste** > 0.
     - Donc il est **impossible** de satisfaire toutes les contraintes originales sans artifices.
     - Conclusion : **le problème initial n'a aucune solution admissible** (contraintes incompatibles). **Fin.**
   - **Cas 2 :** $f_{\text{aux}}^{\max} = 0$
     - Cela signifie que l'on a réussi à obtenir **toutes les variables artificielles à** 0.
     - On a donc trouvé un **point admissible** pour le problème initial.
     - On peut passer à la **Phase II**.

6. **Passage à la Phase II : réutiliser le tableau pour le problème initial**
   (a) **Remplacer la fonction auxiliaire par la fonction objectif initiale**
     - On enlève la ligne f_{aux}.
     - On écrit la ligne de la vraie fonction objectif f dans le tableau.

   (b) **Supprimer les colonnes des variables artificielles**
     - Les variables artificielles n'ont servi qu'à démarrer.
     - On supprime donc leurs colonnes (elles ne doivent plus intervenir).

   (c) **Réécrire la ligne de f en termes des variables hors-base**
     - Comme au simplexe classique : la dernière ligne doit être cohérente avec la base actuelle.
     - Donc, si des variables de base apparaissent dans la ligne de f, on les élimine par combinaisons linéaires des lignes (pivot de Gauss sur la dernière ligne).

7. **Résoudre le problème initial (Phase II)**
   - Le tableau obtenu est maintenant un tableau de simplexe standard, avec un **point de base admissible**.
   - On applique alors la méthode du simplexe normalement jusqu'à optimalité (ou non-borné).

Exemple. On considère le programme linéaire suivant :

$$\texttt{maximiser} \quad f(x_1, x_2) = x_1 + 3x_2$$

$$\texttt{sous les contraintes} \quad (P) : \begin{cases} 2x_1 - 2x_2 \leq 4 \\ 8x_1 + 4x_2 \leq 16 \\ 3x_1 + 6x_2 \geq 9 \\ x_1, x_2 \geq 0 \end{cases}$$

Trouvez un premier point de base admissible de P.

Solution. On simplifie chaque inégalité en divisant par un facteur commun :

$$\begin{cases} 2x_1 - 2x_2 \leq 4 \iff x_1 - x_2 \leq 2, \\ 8x_1 + 4x_2 \leq 16 \iff 2x_1 + x_2 \leq 4, \\ 3x_1 + 6x_2 \geq 9 \iff x_1 + 2x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

Passage en forme standard (avec variables d'écart)
On transforme les $\leq$ en équations en ajoutant des variables d'écart $y_1, y_2 \geq 0$, et on transforme la contrainte $\geq$ en équation en *retirant* une variable d'écart $y_3 \geq 0$:

$$\begin{aligned} x_1 - x_2 + y_1 &= 2, \\ 2x_1 + x_2 + y_2 &= 4, \\ x_1 + 2x_2 - y_3 &= 3, \\ x_1, x_2, y_1, y_2, y_3 &\geq 0. \end{aligned}$$

Le point de base naturel est-il admissible ?
Si on met les variables « libres » à 0 (ici $x_1 = x_2 = 0$), alors on obtient :

$$y_1 = 2, \qquad y_2 = 4, \qquad y_3 = -3,$$

donc **ce point de base n'est pas admissible** (car $y_3 < 0$).

La difficulté vient de la 3ᵉ équation. On ajoute une variable artificielle $y_4 \geq 0$:

$$x_1 + 2x_2 - y_3 + y_4 = 3$$

ce qui permet d'avoir un point de base admissible pour le problème modifié P' en prenant

$$x_1 = x_2 = y_3 = 0 \quad \Longrightarrow \quad y_4 = 3 \ (\geq 0).$$

On résout alors le **problème auxiliaire** :

$$\texttt{maximiser} \quad f_{\text{aux}} = -y_4 \qquad \text{(ainsi on force } y_4 \to 0 \text{ si c'est possible).}$$

Écrire f_{aux} en fonction des variables hors-base
Depuis $x_1 + 2x_2 - y_3 + y_4 = 3$, on a

$$y_4 = 3 - x_1 - 2x_2 + y_3 \quad \Longrightarrow \quad f_{\text{aux}} = -y_4 = -3 + x_1 + 2x_2 - y_3.$$

On place donc la dernière ligne sous la forme :

$$-x_1 - 2x_2 + y_3 + f_{\text{aux}} = -3.$$

Tableau initial du problème auxiliaire

	x_1	x_2	y_1	y_2	y_3	y_4	f_{aux}	B
y_1	1	-1	1	0	0	0	0	2
y_2	2	1	0	1	0	0	0	4
y_4	1	2	0	0	-1	1	0	3
f_{aux}	-1	$\boxed{-2}$	0	0	1	0	1	-3

Le coefficient le plus négatif sur la dernière ligne est $\boxed{-2}$: x_2 **est la variable entrante.** Le pivot est le 2 (ligne y_4, colonne x_2).

Réalisation du pivot de Gauss sur 2

	x_1	x_2	y_1	y_2	y_3	y_4	f_{aux}	B
y_1	1	-1	1	0	0	0	0	2
y_2	2	1	0	1	0	0	0	4
y_4	1	2	0	0	-1	1	0	3
f_{aux}	-1	-2	0	0	1	0	1	-3

$$\begin{array}{c} L_3 \leftarrow \frac{1}{2}L_3 \\ L_1 \leftarrow L_1 + L_3 \\ L_2 \leftarrow L_2 - L_3 \\ L_4 \leftarrow L_4 + 2L_3 \end{array} \quad \longleftrightarrow$$

	x_1	x_2	y_1	y_2	y_3	y_4	f_{aux}	B
y_1	$\frac{3}{2}$	0	1	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{7}{2}$
y_2	$\frac{3}{2}$	0	0	1	$\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{5}{2}$
x_2	$\frac{1}{2}$	1	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{3}{2}$
f_{aux}	0	0	0	0	0	1	1	0

Conclusion
La dernière ligne donne :

$$f_{\text{aux}} = 0 \quad \text{(et on ne peut pas faire mieux car } f_{\text{aux}} \leq 0 \text{ puisque } f_{\text{aux}} = -y_4 \text{ et } y_4 \geq 0\text{).}$$

Donc on a réussi à atteindre $y_4 = 0$: **le programme initial P admet bien des points admissibles.**
Un **premier point de base admissible** (pour P) se lit en posant les variables hors-base à 0. Ici, on peut prendre $x_1 = 0$, $y_3 = 0$ (et $y_4 = 0$), ce qui donne :

$$x_2 = \frac{3}{2}, \qquad y_1 = \frac{7}{2}, \qquad y_2 = \frac{5}{2}, \qquad y_3 = 0.$$

Ainsi, un premier point admissible pour le problème original est :

$$(x_1, x_2) = \left(0, \frac{3}{2}\right).$$

3

La dualité

Jusqu'ici, nous avons appris à résoudre un programme linéaire avec la méthode du simplexe. La **dualité** apporte une deuxième lecture du même problème : à tout programme linéaire (**primal**), on peut associer un autre programme (**dual**) qui porte sur les **contraintes** plutôt que sur les variables.
Concrètement, la dualité sert à :

- **interpréter** les contraintes comme des **prix** ou des **valeurs de ressources** ("combien vaut une unité de ressource ?") ;
- **certifier l'optimalité** : le simplexe fournit une solution, et le dual permet de **prouver** qu'on ne peut pas faire mieux ;
- **détecter des cas limites** : non-borne du primal, impossibilité des contraintes, etc., apparaissent naturellement via le dual ;
- **résoudre autrement** : parfois le dual est plus simple (moins de variables ou contraintes), et on préfère donc le résoudre à la place.

L'idée essentielle à retenir est la suivante : **le maximum du primal et le minimum du dual coïncident** lorsqu'ils admettent des solutions optimales. La dualité relie donc directement le calcul (simplexe) et le sens du problème (valeur des ressources, interprétation économique, preuve d'optimalité).

3.1 Écrire le dual d'un programme linéaire

Avant toute chose, il est essentiel de savoir sans hésitation passer du primal au dual. La règle fondamentale est la suivante :

Les variables du primal deviennent les contraintes du dual,
et les contraintes du primal deviennent les variables du dual.

Pour éviter toute erreur, on procède toujours dans l'ordre ci-dessous.

Méthode. Écrire le dual d'un programme linéaire

1. **Identifier clairement le primal**
   On commence par lire le programme linéaire et identifier :
   - s'il s'agit d'un problème de **maximisation** ou de **minimisation**,
   - le **nombre de variables** et le **nombre de contraintes**,
   - le **type des contraintes** ($\leq$, $\geq$, $=$),
   - le **type des variables** (≥ 0, ≤ 0, variables libres).

   Aucune transformation n'est faite à cette étape : on ne fait que **classer l'information**.
2. **Utiliser le tableau de correspondance primal / dual**
   Le passage du primal au dual se fait à l'aide du tableau suivant, qui doit être lu **ligne par ligne** :

Maximisation	**Minimisation**
Nombre de variables	Nombre de contraintes
Nombre de contraintes	Nombre de variables
Type de contraintes	***Type de variables***
$\leq$	≥ 0
$\geq$	≤ 0
$=$	variable libre
Type de variables	***Type de contraintes***
≥ 0	$\geq$
≤ 0	$\leq$
variable libre	$=$

   Ce tableau se lit ainsi :

- chaque **contrainte du primal** donne **une variable du dual**,
- le **signe de cette variable** dépend du **type de la contrainte**,
- chaque **variable du primal** donne **une contrainte du dual**,
- le **signe de cette contrainte** dépend du **type de la variable**.

3. **Écrire la fonction objectif du dual**

   - Le type d'optimisation s'inverse :

   $$\text{maximiser} \iff \text{minimiser}.$$

   - Les coefficients de la fonction objectif du dual sont les **seconds membres** (b_i) des contraintes du primal.

   Ainsi :
   $$\text{Primal : } \max\ c^T x \quad \Longrightarrow \quad \text{Dual : } \min\ b^T y.$$

4. **Écrire les contraintes du dual**
   Pour chaque variable x_j du primal :
   - on regarde la colonne j de la matrice A,
   - on forme la combinaison linéaire $\sum_i a_{ij} y_i$,
   - on compare cette expression au coefficient c_j de la fonction objectif du primal,
   - le sens de l'inégalité est donné par le **type de la variable** x_j.

   Formellement :
   $$\sum_i a_{ij} y_i \begin{cases} \geq c_j & \text{si } x_j \geq 0, \\ \leq c_j & \text{si } x_j \leq 0, \\ = c_j & \text{si } x_j \text{ est libre.} \end{cases}$$

5. **Vérification finale**
   Avant de conclure, on vérifie systématiquement :
   - que le nombre de variables du dual est égal au nombre de contraintes du primal,
   - que le nombre de contraintes du dual est égal au nombre de variables du primal,
   - que tous les signes ($\leq$, $\geq$, $=$) sont cohérents avec le tableau,
   - que le dual du dual redonne bien le primal.

   Cette étape permet d'éviter les erreurs de signe et de formulation.

Exemple. Soit le programme linéaire suivant :

$$\begin{aligned} &\texttt{minimiser} && -2x_1 - 3x_2 - 2x_3 - 3x_4 \\ &\texttt{sous les contraintes} \quad (P) : && \begin{cases} -2x_1 - x_2 - 3x_3 - 2x_4 \geq -8 \\ 3x_1 + 2x_2 + 2x_3 + x_4 \leq 7 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases} \end{aligned}$$

Écrire le dual (D) du programme (P).

Solution.

1. **Associer une variable duale à chaque contrainte du primal.**
   Le primal a 2 contraintes, donc le dual aura 2 variables :

   $$\text{(C1)} \ : \ -2x_1 - x_2 - 3x_3 - 2x_4 \geq -8 \longrightarrow y_1, \qquad \text{(C2)} \ : \ 3x_1 + 2x_2 + 2x_3 + x_4 \leq 7 \longrightarrow y_2.$$

2. **Déterminer le signe de chaque variable duale en lisant le tableau.**
   Ici, le primal est une **minimisation**. Le tableau dit :

   $$\boxed{\text{contrainte } \geq \Rightarrow y_i \geq 0} \quad \text{et} \quad \boxed{\text{contrainte } \leq \Rightarrow y_i \leq 0}.$$

   Donc :
   $$\boxed{y_1 \geq 0} \quad \text{et} \quad \boxed{y_2 \leq 0}.$$

3. **Écrire l'objectif du dual.**
Quand le primal est une **minimisation**, le dual est une **maximisation**. L'objectif du dual se construit avec les seconds membres b_i des contraintes du primal :

$$b_1 = -8, \quad b_2 = 7 \quad \Longrightarrow \quad \boxed{\texttt{maximiser} \quad -8y_1 + 7y_2.}$$

4. **Déterminer le type des contraintes du dual via le signe des variables du primal.**
Le tableau dit (toujours côté **minimisation**) :

$$\boxed{x_j \geq 0 \Rightarrow \text{ la contrainte duale correspondante est de type } \leq .}$$

Or ici $x_1, x_2, x_3, x_4 \geq 0$, donc **les 4 contraintes du dual seront toutes des** $\leq$.

5. **Construire chaque contrainte du dual en "prenant la colonne" de x_j.**
Règle pratique : pour chaque variable x_j, on récupère ses coefficients dans les contraintes (C1) et (C2), on les combine avec (y_1, y_2), et on compare au coefficient de x_j dans l'objectif du primal.

<u>Pour x_1 :</u> coefficients (-2) dans (C1) et (3) dans (C2), et coefficient objectif $c_1 = -2$:

$$-2y_1 + 3y_2 \leq -2.$$

<u>Pour x_2 :</u> coefficients (-1) et (2), et $c_2 = -3$:

$$-y_1 + 2y_2 \leq -3.$$

<u>Pour x_3 :</u> coefficients (-3) et (2), et $c_3 = -2$:

$$-3y_1 + 2y_2 \leq -2.$$

<u>Pour x_4 :</u> coefficients (-2) et (1), et $c_4 = -3$:

$$-2y_1 + y_2 \leq -3.$$

Conclusion : le dual (D) de (P) est :

$$\boxed{\begin{array}{rl} \texttt{maximiser} & -8y_1 + 7y_2 \\ \texttt{sous les contraintes} \quad (D) : & \begin{cases} -2y_1 + 3y_2 \leq -2 \\ -y_1 + 2y_2 \leq -3 \\ -3y_1 + 2y_2 \leq -2 \\ -2y_1 + y_2 \leq -3 \\ y_1 \geq 0, \ y_2 \leq 0 \end{cases} \end{array}}$$

3.2 Résoudre le primal à l'aide du dual géométriquement

Parfois, il peut se révéler plus simple de résoudre le primal à l'aide du dual géométriquement si ce dernier comporte **deux variables**. L'idée centrale est la suivante : **on résout le dual graphiquement, puis on en déduit la solution du primal** grâce au théorème de dualité.

Cette méthode s'inscrit dans la continuité de la résolution graphique déjà présentée. On suppose donc que :

- le **dual du programme linéaire a déjà été correctement écrit** ;
- le **dual comporte deux variables**, ce qui permet une résolution graphique.

On ne va donc pas redétailler la construction graphique, si vous avez besoin d'un rappel, allez consulter la section 2.2.

Méthode. Résoudre le primal à partir du dual par une approche géométrique

1. **Résoudre graphiquement le dual**

   - On applique directement la méthode graphique déjà connue : tracé des contraintes, détermination de la région admissible, calcul des sommets et évaluation de la fonction objectif.

- On obtient ainsi :
  - un **point optimal du dual** $y^\star$;
  - une **valeur optimale** $f_D^\star$.

2. **Conclure sur la valeur optimale du primal**
   - Si le primal et le dual sont tous deux admissibles, le **théorème de dualité forte** s'applique :

$$f_P^\star = f_D^\star.$$

   - La valeur trouvée graphiquement pour le dual est donc **exactement la valeur optimale du primal**.

3. **Identifier les contraintes actives du dual**
   - On repère, au point optimal $y^\star$, les contraintes du dual qui sont **saturées** (vérifiées avec égalité).
   - Ces contraintes jouent un rôle clé pour retrouver la solution du primal.

4. **Retrouver les variables du primal**
   - Chaque contrainte saturée du dual correspond à une **variable du primal strictement positive**.
   - Chaque contrainte non saturée du dual correspond à une **variable du primal nulle**.
   - On sélectionne alors les contraintes du primal associées aux variables positives et on résout le système correspondant pour obtenir les valeurs de $x^\star$.

5. **Vérifier la cohérence de la solution**
   - Vérifier que la solution $x^\star$ satisfait bien toutes les contraintes du primal.
   - Calculer la valeur de la fonction objectif du primal en $x^\star$.
   - Cette valeur doit coïncider avec celle obtenue via le dual, ce qui confirme la cohérence de la résolution.

Exemple. On considère le problème dual écrit précédemment :

$$\begin{array}{ll} \texttt{maximiser} & -8y_1 + 7y_2 \\ \texttt{sous les contraintes} \quad (D): & \begin{cases} -2y_1 + 3y_2 \leq -2 \\ -y_1 + 2y_2 \leq -3 \\ -3y_1 + 2y_2 \leq -2 \\ -2y_1 + y_2 \leq -3 \\ y_1 \geq 0,\ y_2 \leq 0 \end{cases} \end{array}$$

Résoudre le problème graphiquement pour retrouver la solution du primal.

Solution. On associe une **couleur à chaque contrainte** :

$$\begin{cases} -2y_1 + 3y_2 \leq -2 \\ -y_1 + 2y_2 \leq -3 \\ -3y_1 + 2y_2 \leq -2 \\ -2y_1 + y_2 \leq -3 \end{cases} \quad \text{et} \quad y_1 \geq 0,\ y_2 \leq 0.$$

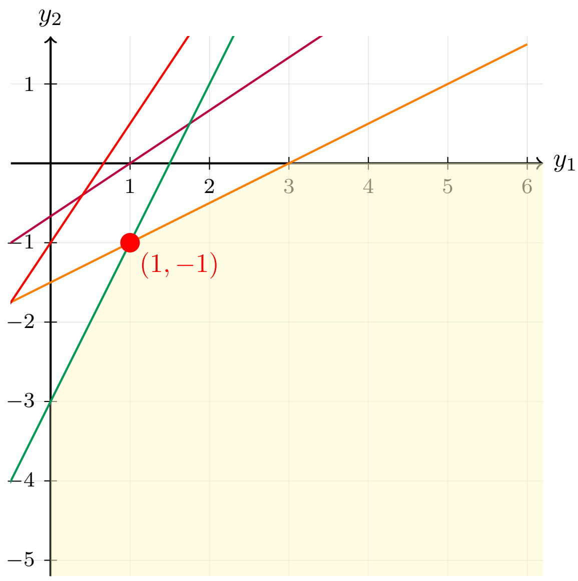
On trace les droites frontières (on remplace $\leq$ par $=$) :

$$\begin{aligned} -2y_1 + 3y_2 = -2 &\iff y_2 = \frac{2y_1 - 2}{3} \\ -y_1 + 2y_2 = -3 &\iff y_2 = \frac{y_1 - 3}{2} \\ -3y_1 + 2y_2 = -2 &\iff y_2 = \frac{3y_1 - 2}{2} \\ -2y_1 + y_2 = -3 &\iff y_2 = 2y_1 - 3 \end{aligned}$$

et on ajoute les contraintes de signe $y_1 \geq 0$ (à droite de l'axe vertical) et $y_2 \leq 0$ (en dessous de l'axe horizontal).

Comme les contraintes sont toutes de la forme $y_2 \le (\cdots)$, le domaine admissible est l'intersection des demi-plans **situés en dessous** de ces droites, en plus de $y_1 \ge 0$ et $y_2 \le 0$.

Représentation graphique



Valeur optimale du dual
D'après le graphique, le maximum est atteint au point $(y_1^\star, y_2^\star) = (1, -1)$. On calcule alors :

$$-8y_1^\star + 7y_2^\star = -8 \times 1 + 7 \times (-1) = -15.$$

Donc $f_D^\star = -15$.

Retrouver la solution du primal
Comme le primal et le dual sont réalisables, la **dualité forte** donne :

$$f_P^\star = f_D^\star = -15.$$

Ainsi, **la valeur optimale du primal est** $\boxed{-15}$. (La résolution du dual suffit donc à retrouver la valeur optimale du primal.)

3.3 Réaliser une analyse en stratégies pures (jeu à somme nulle 2×2)

Nous allons maintenant parler de théorie des jeux, plus précisément des jeux à somme nulle. On considère un jeu matriciel à somme nulle à deux joueurs A et B, défini par une matrice des gains

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix},$$

où A choisit une ligne et B choisit simultanément une colonne. Le gain de A est m_{ij} et la perte de B est $-m_{ij}$.

Objectif :

- le joueur A cherche à **maximiser son gain**,
- le joueur B cherche à **minimiser la perte de** A.

Méthode.

1. **Calcul du *pire gain* pour chaque stratégie de** A
   Pour chaque ligne i, on calcule le **gain minimal** possible pour A si B choisit la colonne la plus défavorable :

$$\min_j(m_{ij}).$$

   - Ligne 1 : $\min(m_{11}, m_{12})$
   - Ligne 2 : $\min(m_{21}, m_{22})$

Ces valeurs représentent le *pire cas* pour A s'il choisit la stratégie correspondante.

2. **Choix optimal de A : critère *maximin***
   Le joueur A adopte la stratégie qui maximise son pire gain :

$$\max_i \big(\min_j (m_{ij})\big).$$

   - La ligne correspondante est la stratégie optimale de A en stratégie pure.

3. **Calcul du *pire gain* pour chaque stratégie de B**
   Pour chaque colonne j, on calcule le **gain maximal** que A pourrait obtenir si B choisit cette colonne :

$$\max_i (m_{ij}).$$

   - Colonne 1 : $\max(m_{11}, m_{21})$
   - Colonne 2 : $\max(m_{12}, m_{22})$

   Ces valeurs représentent le *pire cas* pour B.

4. **Choix optimal de B : critère *minimax***
   Le joueur B choisit la colonne qui minimise ce pire gain :

$$\min_j \big(\max_i (m_{ij})\big).$$

   - La colonne correspondante est la stratégie optimale de B en stratégie pure.

5. **Comparaison des deux valeurs**
   On compare :

$$\max_i \big(\min_j (m_{ij})\big) \quad \text{et} \quad \min_j \big(\max_i (m_{ij})\big).$$

   - Si les deux valeurs sont **égales**, le jeu admet une **valeur en stratégie pure**.
   - Cette valeur est la **valeur du jeu**.
   - Le couple (ligne optimale, colonne optimale) forme un **équilibre** : aucun joueur n'a intérêt à changer seul de stratégie.

6. **Conclusion**
   - Si

$$\max_i \big(\min_j (m_{ij})\big) = \min_j \big(\max_i (m_{ij})\big),$$

     alors le jeu est résolu en **stratégies pures**.
   - Sinon, il existe un **saut de dualité** et il faut passer à une **analyse en stratégies mixtes**.

Exemple. Soit le jeu matriciel à somme nulle

$$M = \begin{pmatrix} 3 & 1 \\ 0 & 4 \end{pmatrix}.$$

Le joueur A choisit la ligne et le joueur B choisit la colonne.
Ce jeu a-t-il une valeur en stratégie pure ?

Solution. On commence par calculer le *pire gain* de A pour chaque ligne (critère du maximin).

- Ligne A_1 : $\min(3, 1) = 1$
- Ligne A_2 : $\min(0, 4) = 0$

Ainsi,

$$\max_i \big(\min_j (m_{ij})\big) = \max(1, 0) = 1.$$

On calcule ensuite le *pire cas* pour B colonne par colonne (critère du minimax).

- Colonne B_1 : $\max(3, 0) = 3$
- Colonne B_2 : $\max(1, 4) = 4$

Ainsi,

$$\min_j\big(\max_i(m_{ij})\big) = \min(3,4) = 3.$$

Conclusion
On obtient :

$$\max_i\big(\min_j(m_{ij})\big) = 1 \quad \text{et} \quad \min_j\big(\max_i(m_{ij})\big) = 3.$$

Comme ces deux valeurs sont différentes, le jeu **n'a pas de valeur en stratégie pure**. Il n'existe donc pas de couple de stratégies pures formant un équilibre.

Le **saut de dualité** vaut ici :

$$3 - 1 = 2.$$

Il faudra donc analyser ce jeu en **stratégies mixtes**.

Remarque. Pour aller plus vite sur la rédaction, on peut résumer l'analyse en stratégies pures à l'aide du tableau suivant, qui met en évidence les critères de *maximin* pour le joueur A et de *minimax* pour le joueur B :

	B_1	B_2	Pire gain de A
A_1	3	1	$\boxed{1}$
A_2	0	4	0
Pire choix de B	$\boxed{3}$	4	

On écrira après si minimax $\neq$ maximin et si c'est le cas, le saut de dualité.

3.4 Résoudre des problèmes en stratégies mixtes (jeu à somme nulle 2 $\times$ 2)

Évidemment cette méthode ne s'applique que si le jeu n'admet pas de valeur en stratégie pure. Il faut donc toujours exécuter la méthode précédente avant d'éventuellement s'embarquer dans celle-ci.

Méthode. Résoudre un jeu à somme nulle 2×2 en stratégies mixtes

1. **Définir les stratégies mixtes des joueurs**
   - Le joueur A joue :
     $$A_1 \text{ avec la probabilité } p, \qquad A_2 \text{ avec la probabilité } 1-p.$$
   - Le joueur B joue :
     $$B_1 \text{ avec la probabilité } q, \qquad B_2 \text{ avec la probabilité } 1-q.$$
2. **Déterminer la stratégie optimale du joueur A**
   - On exprime le **gain espéré de** A en fonction de p :
     - si B joue B_1 ;
     - si B joue B_2.
   - À l'équilibre, A choisit p de façon à rendre B **indifférent** entre B_1 et B_2.
   - On impose donc l'égalité des deux gains espérés.
   - La résolution donne la probabilité optimale $p^\star$.
3. **Déterminer la stratégie optimale du joueur B**
   - On exprime le **gain espéré de** A en fonction de q :
     - si A joue A_1 ;
     - si A joue A_2.
   - À l'équilibre, B choisit q de façon à rendre A **indifférent** entre A_1 et A_2.

- On impose donc l'égalité des deux gains espérés.
- La résolution donne la probabilité optimale $q^\star$.

4. **Calculer la valeur du jeu**
   - On remplace $p^\star$ ou $q^\star$ dans une expression du gain espéré.
   - La valeur obtenue est la **valeur du jeu** :

$$\mu = \text{gain moyen de } A \text{ à l'équilibre.}$$

5. **Conclusion**
   - Les stratégies $(p^\star, 1-p^\star)$ et $(q^\star, 1-q^\star)$ forment un **équilibre en stratégies mixtes**.
   - Aucun joueur n'a intérêt à modifier seul sa stratégie.
   - La valeur μ est unique et représente le gain moyen garanti de A.

Exemple. Soit le jeu matriciel à somme nulle

$$M = \begin{pmatrix} 3 & 1 \\ 0 & 4 \end{pmatrix}.$$

Le joueur A choisit la ligne et le joueur B choisit la colonne.
En stratégie mixte, quelle sera la stratégie de A et de B à l'équilibre ? Quelle est alors la valeur du jeu ?

Solution. Comme le jeu n'a pas de valeur en stratégie pure, on cherche un équilibre en **stratégies mixtes**.

Stratégie optimale du joueur A
On note x la probabilité que A joue A_1 (donc A_2 avec probabilité $1-x$).

- Si B joue B_1, le gain espéré de A vaut :

$$3x + 0(1-x) = 3x.$$

- Si B joue B_2, le gain espéré de A vaut :

$$1x + 4(1-x) = -3x + 4.$$

À l'équilibre, A choisit x de sorte que B soit indifférent entre B_1 et B_2 :

$$3x = -3x + 4 \iff 6x = 4 \iff x = \frac{2}{3}.$$

Donc la stratégie mixte optimale de A est :

$$(A_1, A_2) = \left(\frac{2}{3}, \frac{1}{3}\right).$$

Stratégie optimale du joueur B
On note y la probabilité que B joue B_1 (donc B_2 avec probabilité $1-y$).

- Si A joue A_1, le gain espéré de A vaut :

$$3y + 1(1-y) = 2y + 1.$$

- Si A joue A_2, le gain espéré de A vaut :

$$0y + 4(1-y) = -4y + 4.$$

À l'équilibre, B choisit y de sorte que A soit indifférent entre A_1 et A_2 :

$$2y + 1 = -4y + 4 \iff 6y = 3 \iff y = \frac{1}{2}.$$

Donc la stratégie mixte optimale de B est :

$$(B_1, B_2) = \left(\frac{1}{2}, \frac{1}{2}\right).$$

Valeur du jeu
On calcule le gain moyen de A à l'équilibre (par exemple avec B_1) :

$$\mu = 3x = 3 \times \frac{2}{3} = 2.$$

$$\boxed{\text{Valeur du jeu : } \mu = 2.}$$

Ainsi, à l'équilibre :

$$\boxed{A : \left(\frac{2}{3}, \frac{1}{3}\right), \qquad B : \left(\frac{1}{2}, \frac{1}{2}\right), \qquad \mu = 2.}$$

3.5 Jeux matriciels de taille $2 \times p$

Jusqu'à présent, nous avons étudié des jeux à somme nulle de type 2×2, pour lesquels les stratégies mixtes peuvent être déterminées explicitement à l'aide d'égalités d'indifférence. Cette méthode est simple, efficace et calculable à la main, mais elle repose fortement sur le fait que chaque joueur ne dispose que de deux stratégies possibles.
Lorsque le joueur B dispose de plus de deux stratégies (cas $2 \times p$), la démarche doit être adaptée. Il n'est alors plus possible de rendre toutes les stratégies pures de l'adversaire indifférentes simultanément par un simple système de deux équations.

Dans ce contexte, l'analyse repose sur une **interprétation géométrique** du gain espéré du joueur A : chaque stratégie pure de B définit une fonction affine du paramètre de mélange de A, et le comportement optimal des joueurs peut être déterminé en étudiant l'enveloppe inférieure de ces fonctions.

Méthode. Résoudre un jeu à somme nulle de type $2 \times p$
On considère un jeu matriciel à somme nulle de taille $2 \times p$:

	B_1	B_2	$\cdots$	B_p
A_1	m_{11}	m_{12}	$\cdots$	m_{1p}
A_2	m_{21}	m_{22}	$\cdots$	m_{2p}

où le joueur A choisit une ligne et le joueur B une colonne.
On suppose que le jeu **n'admet pas de valeur en stratégie pure** (maximin $\neq$ minimax).

1. **Stratégie mixte du joueur A**

   - Le joueur A mélange entre ses deux stratégies.
   - On note $x_1 \in [0, 1]$ la probabilité de jouer A_1.
   - La probabilité de jouer A_2 est alors $1 - x_1$.

   La stratégie de A est donc entièrement décrite par le paramètre x_1.

2. **Gain espéré de A face aux stratégies pures de B**
   Si le joueur B joue la colonne B_j en stratégie pure, alors le gain espéré de A vaut :

   $$G_j(x_1) = m_{1j}x_1 + m_{2j}(1 - x_1).$$

   - Chaque fonction $G_j(x_1)$ est une **fonction affine** de x_1.
   - Géométriquement, chaque colonne B_j correspond à une **droite** dans le plan (x_1, gain).

3. **Choix défavorable du joueur B**

   - Pour une valeur donnée de x_1, le joueur B choisit la colonne qui **minimise** le gain de A.

- Le gain garanti de A est donc :

$$G(x_1) = \min_{1 \le j \le p} G_j(x_1).$$

Géométriquement, $G(x_1)$ est l'**enveloppe inférieure** des droites $G_j(x_1)$.

4. **Détermination de la stratégie optimale de A**

   - Le joueur A choisit x_1 de façon à **maximiser** son gain garanti $G(x_1)$.
   - Le maximum est atteint en un point d'intersection de deux droites G_{j_1} et G_{j_2}.

   À l'optimum, le joueur A rend le joueur B **indifférent** entre au moins deux stratégies pures.

5. **Élimination des stratégies dominées de B**

   - Toute colonne dont la droite est **toujours au-dessus** d'une autre n'appartient jamais à l'enveloppe inférieure.
   - Ces stratégies sont **strictement dominées** et ne seront jamais jouées par B.

   Le problème se réduit alors à un jeu 2×2 entre les colonnes actives.

6. **Stratégie mixte optimale du joueur B**

   - Le joueur B mélange uniquement entre les colonnes B_{j_1} et B_{j_2} actives à l'optimum.
   - Il choisit les probabilités de manière à rendre A indifférent entre A_1 et A_2.

   On détermine ces probabilités en égalisant les pertes espérées de B associées à A_1 et A_2.

7. **Valeur du jeu**

   - La valeur du jeu est le gain commun obtenu à l'équilibre :

   $$\mu = \max_{x_1 \in [0,1]} \min_j G_j(x_1).$$

   - Cette valeur est unique.
   - Elle coïncide avec la valeur obtenue en résolvant le problème dual du joueur B (théorème du minimax de von Neumann).

Exemple. Soit le jeu matriciel à somme nulle défini par la matrice suivante

$$M = \begin{pmatrix} 2 & 5 & 6 \\ 9 & 1 & 2 \end{pmatrix}.$$

Dans ce jeu, à chaque tour, le joueur A choisit la ligne et le joueur B choisit la colonne.

1. Ce jeu a-t-il une valeur en stratégie pure ? Si oui, laquelle.
2. En stratégie mixte, tracez le gain de A en fonction de sa stratégie. Que pouvez-vous alors en déduire sur la stratégie de B.
3. Quelle est alors la stratégie de B ?
4. En stratégie mixte, ce jeu a-t-il une valeur ?

Solution.

1. **Valeur en stratégie pure**
   On résume l'analyse *maximin/minimax* dans le tableau suivant :

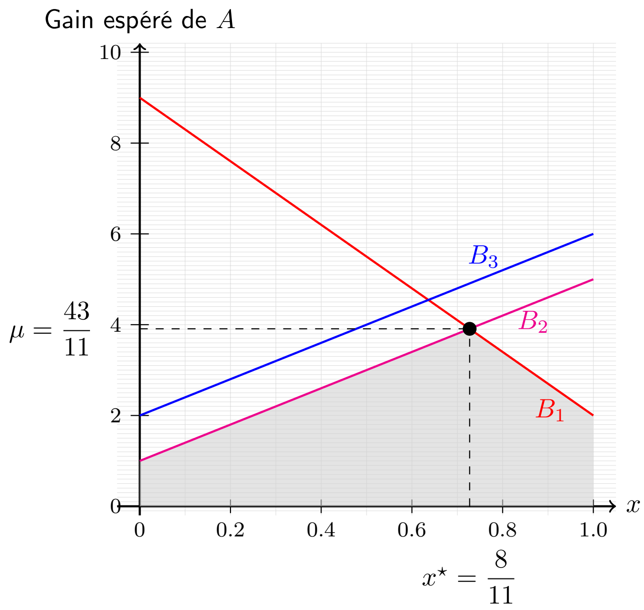
	B_1	B_2	B_3	Pire gain de A
A_1	2	5	6	$\boxed{2}$
A_2	9	1	2	1
Pire choix de B	9	$\boxed{5}$	6	

   Comme $2 \neq 5$, le jeu **n'a pas de valeur en stratégie pure**. Le saut de dualité vaut $5 - 2 = 3$.

2. **Gain de A en fonction de sa stratégie mixte**
Soit x la probabilité que A joue A_1 (et $1-x$ celle de jouer A_2). Pour chaque colonne B_j, on calcule le gain espéré de A :

   - Si B joue B_1 :
$$G_{B_1}(x) = 2x + 9(1-x) = -7x + 9.$$
   - Si B joue B_2 :
$$G_{B_2}(x) = 5x + 1(1-x) = 4x + 1.$$
   - Si B joue B_3 :
$$G_{B_3}(x) = 6x + 2(1-x) = 4x + 2.$$

   On trace alors ces trois droites $G_{B_1}, G_{B_2}, G_{B_3}$ sur $x \in [0,1]$ et on considère leur **enveloppe inférieure** (car B choisit la colonne la plus défavorable à A).



   On observe que pour tout x, on a
$$G_{B_3}(x) = 4x + 2 \; > \; 4x + 1 = G_{B_2}(x).$$
   Donc la stratégie B_3 est **toujours moins défavorable** à A que B_2 : B ne la choisira jamais à l'optimum. Autrement dit, B_3 **est dominée par** B_2 et on peut l'éliminer.

3. **Stratégie optimale de B**
Le jeu se réduit donc aux colonnes B_1 et B_2, c'est-à-dire à la matrice :
$$\widetilde{M} = \begin{pmatrix} 2 & 5 \\ 9 & 1 \end{pmatrix}.$$
   Soit y la probabilité que B joue B_1 (et $1-y$ celle de jouer B_2). On calcule le gain espéré de A selon la ligne jouée :

   - Si A joue A_1 :
$$H_{A_1}(y) = 2y + 5(1-y) = -3y + 5.$$
   - Si A joue A_2 :
$$H_{A_2}(y) = 9y + 1(1-y) = 8y + 1.$$

   À l'équilibre, B choisit y de façon à rendre A indifférent entre A_1 et A_2 :
$$-3y + 5 = 8y + 1 \quad \Longleftrightarrow \quad 11y = 4 \quad \Longleftrightarrow \quad y = \frac{4}{11}.$$
   Donc la stratégie optimale de B est :
$$(B_1, B_2, B_3) = \left(\frac{4}{11}, \frac{7}{11}, 0\right).$$

4. **Valeur du jeu en stratégie mixte**
   On évalue le gain commun à l'équilibre (par exemple avec $H_{A_2}(y)$) :

$$\mu = 8y + 1 = 8 \cdot \frac{4}{11} + 1 = \frac{32}{11} + \frac{11}{11} = \frac{43}{11}.$$

   Ainsi, le jeu **a une valeur en stratégie mixte**, et cette valeur est :

$$\boxed{\mu = \frac{43}{11}}.$$

   (Vérification possible côté A) À l'équilibre, A choisit x tel que $G_{B_1}(x) = G_{B_2}(x)$:

$$-7x + 9 = 4x + 1 \iff 11x = 8 \iff x = \frac{8}{11}.$$

   Donc

$$(A_1, A_2) = \left(\frac{8}{11}, \frac{3}{11}\right),$$

   et on retrouve bien :

$$\mu = -7 \cdot \frac{8}{11} + 9 = \frac{43}{11}.$$

4 Problèmes de flots

Cette partie n'est plus au programme cette année. Elle concernait le projet de l'année dernière, mais les curieux peuvent toujours la lire.

Vous aurez sans doute à réaliser un projet sur cette partie du cours qui n'est pas vue en CM ou en TD. Pourtant, maîtriser l'exécution de ces méthodes à la main est essentiel pour comprendre les algorithmes afin de les implémenter correctement, et aussi parce que, sait-on jamais, cela pourrait tomber un jour au DE...

Un **problème de flot** consiste à faire circuler une « quantité » (eau, voitures, données, ambulances, etc.) dans un **graphe orienté**. Chaque arête a une **capacité** : on ne peut pas faire passer plus que cette limite. On cherche à envoyer le flot d'une **source** s vers un **puits** t en respectant deux règles : (1) *capacité* sur chaque arête, (2) *conservation* aux sommets intermédiaires (ce qui entre = ce qui sort).

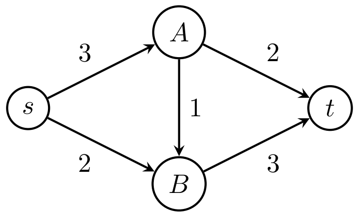


FIGURE 1 – Exemple de réseau de flot (les nombres indiquent les capacités).

Dans l'exemple des ambulances, les rues sont les arêtes (avec leur capacité de circulation) : le **flot maximal** est alors le **nombre maximum d'ambulances** pouvant aller simultanément de l'Hôtel-Dieu (s) au Panthéon (t).

Nous allons détailler deux méthodes de résolution de problèmes de flots (que vous devrez implémenter avec le langage de programmation de votre choix !).

4.1 Méthode de Ford-Fulkerson

L'algorithme d'Edmonds–Karp est une **implémentation efficace et déterministe** de la méthode de Ford–Fulkerson pour résoudre un problème de **flot maximal** dans un réseau orienté capacitaire.

Il repose sur deux idées fondamentales :

- l'utilisation du **réseau résiduel** pour représenter toutes les modifications possibles du flot ;
- la recherche systématique de **chaînes améliorantes les plus courtes** à l'aide d'un **parcours en largeur (BFS)**.

Méthode. Algorithme d'Edmonds–Karp (flot maximal)

1. **Initialisation du flot**
   - On commence avec un flot nul : $f(u, v) = 0$ pour toute arête (u, v).
   - La valeur initiale du flot est donc $|f| = 0$.

2. **Construction du réseau résiduel**
   À partir du flot courant f, on construit le **réseau résiduel** G_f :
   - pour chaque arête (u, v) de capacité $c(u, v)$:
     - une arête (u, v) de capacité résiduelle $c(u, v) - f(u, v)$;
     - une arête (v, u) de capacité résiduelle $f(u, v)$.

   Le réseau résiduel encode :
   - les augmentations possibles du flot ;
   - les diminutions (ou annulations) de flot déjà envoyé.

3. **Recherche d'une chaîne améliorante par parcours en largeur**
   - On applique un **parcours en largeur (BFS)** dans le réseau résiduel, en partant de la source s.
   - Le BFS explore les sommets par distance croissante (nombre d'arêtes).
   - Dès que le puits t est atteint, on a trouvé une **chaîne améliorante** qui est un **plus court chemin** de s vers t dans G_f.

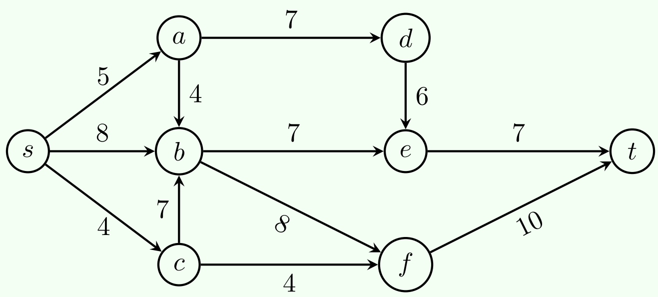
4. **Augmentation du flot**
   - On détermine la capacité résiduelle minimale le long du chemin trouvé :
   $$c_\Gamma = \min\{c_f(u,v) \mid (u,v) \in \Gamma\}.$$
   - On augmente le flot de c_Γ le long du chemin :
     - on ajoute c_Γ sur les arêtes directes ;
     - on retranche c_Γ sur les arêtes inverses.

   La valeur totale du flot augmente alors strictement :
   $$|f| \leftarrow |f| + c_\Gamma.$$

5. **Itération**
   - On reconstruit le réseau résiduel.
   - On relance un parcours en largeur pour chercher une nouvelle chaîne améliorante.
   - Tant qu'un chemin de s vers t existe dans G_f, le flot n'est pas maximal.

6. **Condition d'arrêt**
   - Lorsque le parcours en largeur ne permet plus d'atteindre t depuis s, il n'existe plus de chaîne améliorante.
   - Le flot courant est alors **maximal**.

Exemple. Résoudre le problème de flot suivant avec la méthode de Ford-Fulkerson.



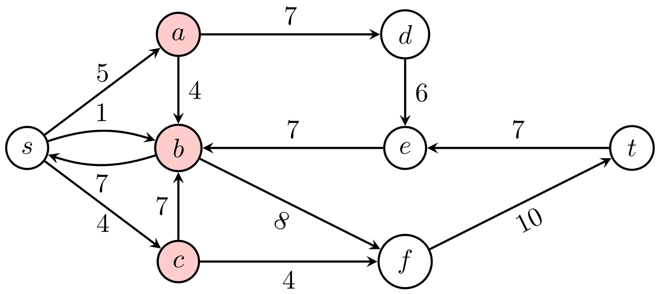
Solution.

1. **Parcours en largeur :**

```
s
a b c ; pi(a)=s, pi(b)=s, pi(c)=s
b c d ; pi(d)=a
c d e f ; pi(e)=b, pi(f)=b
f t ; pi(t)=e
```

   Chaîne améliorante 1 : $s \to b \to e \to t$ et **flot** 7.

2. **Graphe résiduel :**

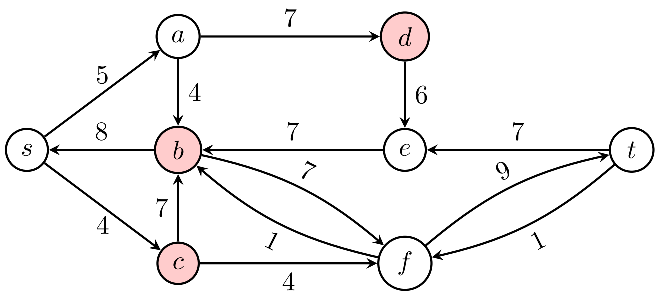


Parcours en largeur :

```
s
a b c ; pi(a)=s, pi(b)=s, pi(c)=s
b c d ; pi(d)=a
c d f ; pi(f)=b
f e ; pi(e)=d
e t ; pi(t)=f
```

Chaîne améliorante 2 : $s \to b \to f \to t$ et **flot** 1.

3. **Graphe résiduel :**

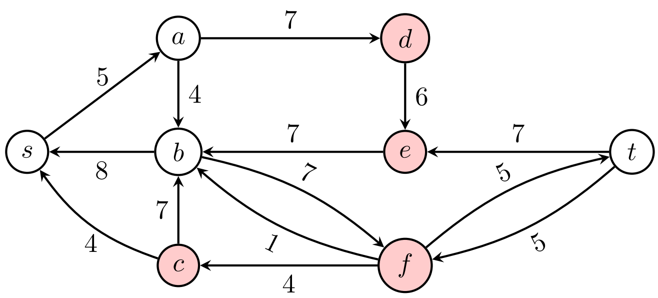


Parcours en largeur :

```
s
a c ; pi(a)=s, pi(c)=s
c b d ; pi(b)=a, pi(d)=a
b d f ; pi(f)=c
f e ; pi(e)=d
e t ; pi(t)=f
```

Chaîne améliorante 3 : $s \to c \to f \to t$ et **flot** 4.

4. **Graphe résiduel :**

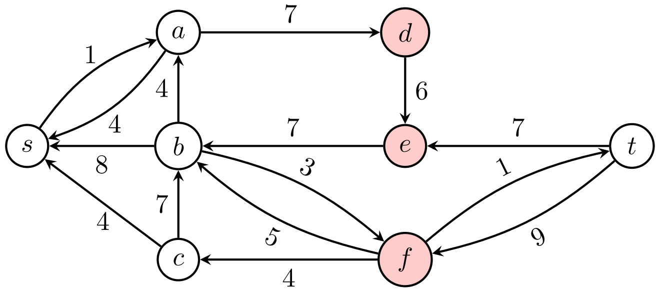


Parcours en largeur :

```
s
a ; pi(a)=s
b d ; pi(b)=a, pi(d)=a
d f ; pi(f)=b
f e ; pi(e)=d
e c t ; pi(c)=f, pi(t)=f
```

Chaîne améliorante 4 : $s \to a \to b \to f \to t$ et **flot** 4.

5. **Graphe résiduel :**

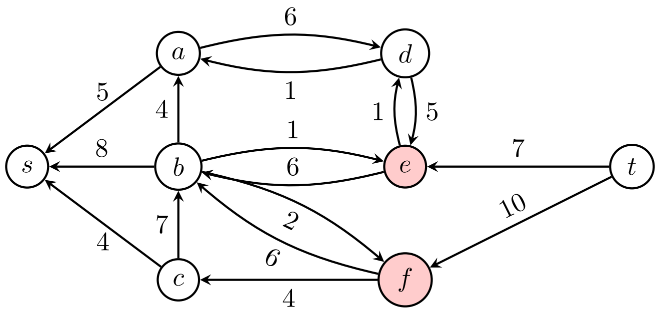


Parcours en largeur :

```
s
a ; pi(a)=s
d ; pi(d)=a
e ; pi(e)=d
b ; pi(b)=e
f ; pi(f)=b
c t ; pi(c)=f, pi(t)=f
```

Chaîne améliorante 5 : $s \to a \to d \to e \to b \to f \to t$ et **flot** 1.

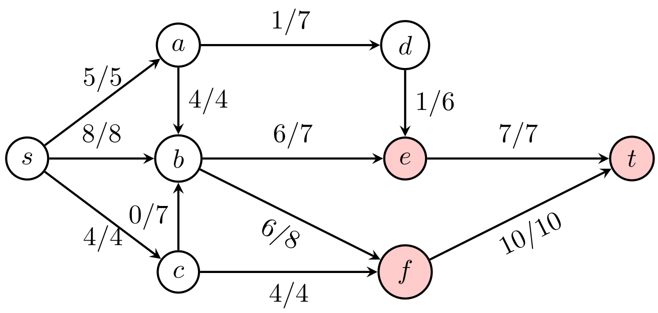
6. **Graphe résiduel :**



Pas de chaîne améliorante : on conclut que le flot est **maximal**. Sa valeur vaut

$$|f| = 7 + 1 + 4 + 4 + 1 = 17.$$

Solution (flots à envoyer sur le graphe initial) : On prend les flots backwards du graphe résiduel précédent.



4.2 Algorithme Pousser-Réétiqueter

L'algorithme **pousser–réétiqueter** est une méthode efficace pour calculer un **flot maximal** dans un réseau. Contrairement aux algorithmes classiques comme Ford–Fulkerson, il ne cherche pas directement des chemins

complets entre la source et le puits.
Son idée centrale est la suivante :

- on commence par envoyer *autant de flot que possible* depuis la source, sans se soucier immédiatement de l'équilibre aux autres sommets ;
- certains sommets reçoivent alors trop de flot et se retrouvent en *situation de débordement* ;
- l'algorithme consiste à **évacuer progressivement ce surplus** en le poussant vers d'autres sommets, jusqu'à ce que l'équilibre soit rétabli partout.

Pour permettre ce fonctionnement, l'algorithme autorise temporairement des situations non équilibrées : on parle alors de **pré-flot**. Un sommet (sauf la source et le puits) peut recevoir plus de flot qu'il n'en envoie ; cet excès devra ensuite être redistribué.

Afin de guider le déplacement du flot, chaque sommet est associé à une **hauteur**, que l'on peut interpréter comme une altitude :

- le flot ne peut être poussé que *vers le bas*, c'est-à-dire vers des sommets de hauteur plus faible ;
- si aucun voisin plus bas n'existe, le sommet est *relevé* (réétiqueté) afin de créer une nouvelle direction possible pour le flot.

Enfin, comme dans la méthode de Ford–Fulkerson, l'algorithme s'appuie sur le **réseau résiduel**, qui indique à chaque instant :

- combien de flot peut encore être envoyé sur une arête ;
- et dans quel sens un flot déjà envoyé peut éventuellement être annulé.

L'algorithme alterne alors deux opérations simples : **pousser** le flot lorsqu'une direction descendante est possible, et **réétiqueter** les sommets bloqués. Lorsque plus aucun sommet ne présente de surplus, le pré-flot devient un flot équilibré et celui-ci est **maximal**.

Méthode. Algorithme pousser–réétiqueter
Pour appliquer l'algorithme, on maintient en permanence, pour chaque sommet u :

- une **hauteur** $h(u)$, qui indique le "niveau" du sommet ;
- un **excédent** $e(u)$, qui mesure la quantité de flot en trop stockée au sommet.

L'excédent $e(u)$ se calcule toujours de la même manière :

$$\textit{excédent} = \textit{flot entrant - flot sortant.}$$

1. **Initialisation du pré-flot**
   - On fixe la hauteur de la source à une valeur très grande :
     $$h(s) = |V|,$$
     et la hauteur de tous les autres sommets (y compris le puits) à 0.
   - On met tous les flots à zéro.
   - On **sature immédiatement toutes les arêtes sortant de la source**.

   Concrètement, pour chaque arête (s, u) :
   - on met sur l'arête le flot égal à sa capacité ;
   - le sommet u reçoit un excédent égal à cette capacité ;
   - la source devient déficitaire (mais on n'en tient pas compte).

2. **Identifier les sommets actifs**
   Un sommet est dit **actif** s'il vérifie :
   - ce n'est ni la source ni le puits ;
   - son excédent est strictement positif.

   Intuitivement, un sommet actif est un sommet qui *débord e* et doit se débarrasser de son excédent.

3. **Tenter une opération de poussée**
   Soit u un sommet actif. On regarde ses arêtes sortantes dans le **réseau résiduel**. Une **poussée** de u vers un voisin v est possible si :
   - il reste de la capacité disponible sur l'arête (u, v) ;
   - le sommet v est exactement **un niveau plus bas** que u.

Si ces conditions sont réunies :

- on envoie le maximum possible de flot,
- c'est-à-dire le minimum entre l'excédent de u et la capacité restante de l'arête ;
- l'excédent de u diminue ;
- l'excédent de v augmente.

Cette opération correspond à *faire couler l'eau vers le bas*.

4. **Réétiqueter si aucune poussée n'est possible**
   Si un sommet actif u ne peut pousser son excédent vers *aucun* voisin, alors il est bloqué.

   Dans ce cas :

   - on augmente la hauteur de u ;
   - on choisit la plus petite hauteur parmi ses voisins accessibles ;
   - on fixe $h(u)$ juste au-dessus de cette hauteur.

   Le réétiquetage permet de *créer une pente* afin qu'une poussée devienne possible à l'étape suivante.

5. **Ordre de traitement**
   Lorsque plusieurs sommets sont actifs :

   - on choisit en priorité celui de plus grande hauteur ;
   - en cas d'égalité, on applique un ordre fixé (par exemple alphabétique).

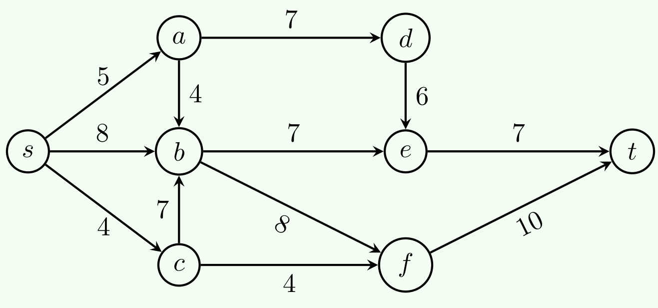
6. **Condition d'arrêt**
   L'algorithme s'arrête lorsqu'il n'y a plus de sommet actif. À ce moment-là :

   - tous les excédents ont été éliminés ;
   - le pré-flot respecte la conservation du flot ;
   - le flot obtenu est **maximal**.

7. **Valeur du flot maximal**
   La valeur du flot maximal est simplement :

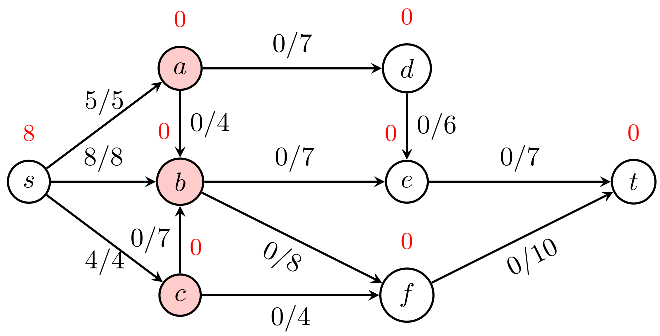
   le flot total sortant de la source (ou entrant dans le puits).

Exemple. Résoudre le problème de flot suivant avec la méthode Pousser-Réétiqueter.



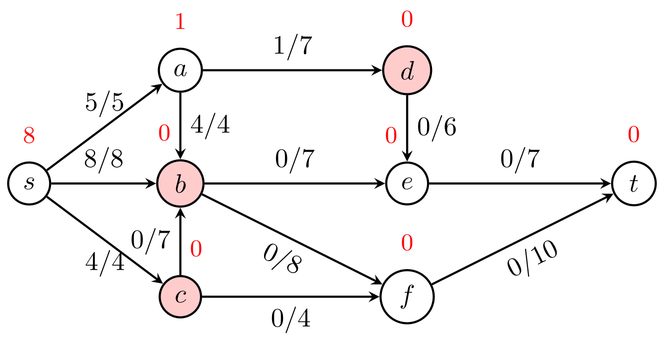
Solution.

1. **Initialisation (pré-flot).** On sature les arêtes sortantes de s.



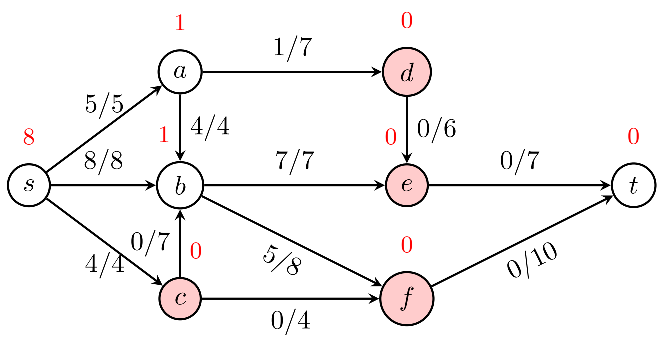
Excès : $e = (0, 5, 8, 4, 0, 0, 0, 0)$.

2. **Itération 1 : décharge de a (ordre alphabétique).**



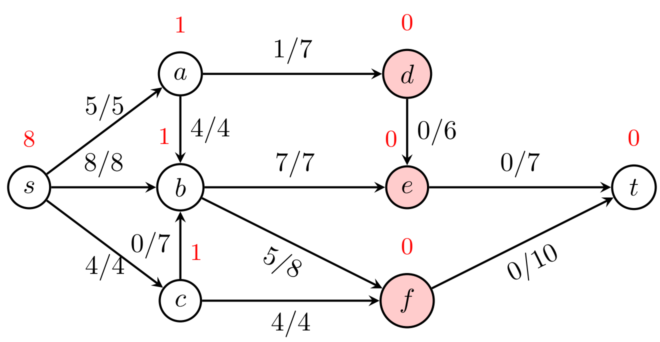
Excès : $e = (0, 0, 12, 4, 1, 0, 0, 0)$.

3. **Itération 2 : décharge de b (ordre alphabétique).**



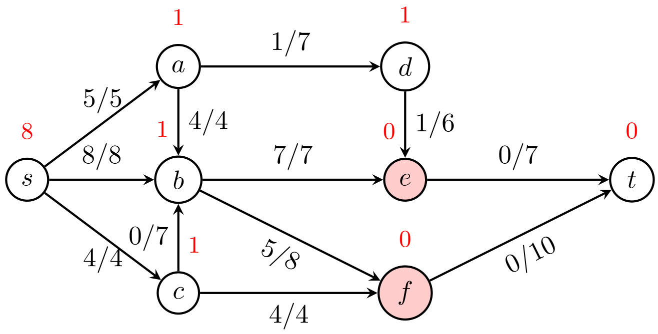
Excès : $e = (0, 0, 0, 4, 1, 7, 5, 0)$.

4. **Itération 3 : décharge de c (ordre alphabétique).**



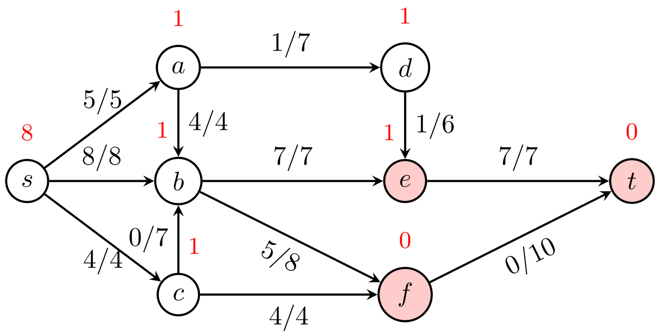
Excès : $e = (0, 0, 0, 0, 1, 7, 9, 0)$.

5. **Itération 4 : décharge de d (ordre alphabétique).**



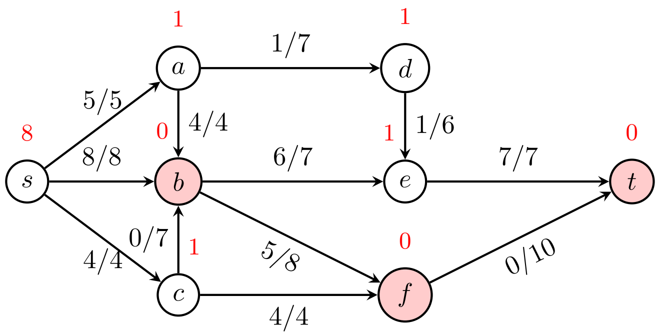
Excès : $e = (0, 0, 0, 0, 0, 8, 9, 0)$.

6. **Itération 5 : décharge de e (ordre alphabétique).**



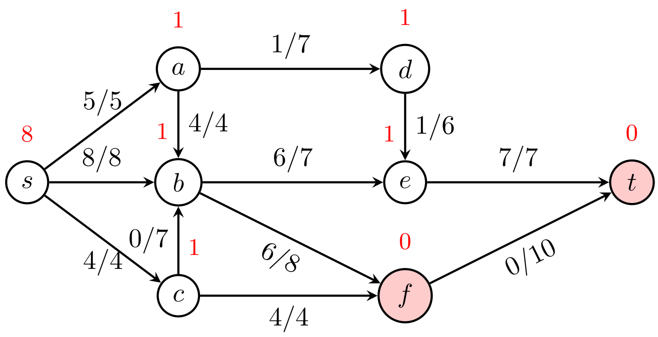
Excès : $e = (0, 0, 0, 0, 0, 1, 9, 7)$.

7. **Itération 6 : décharge de e (plus haut).**



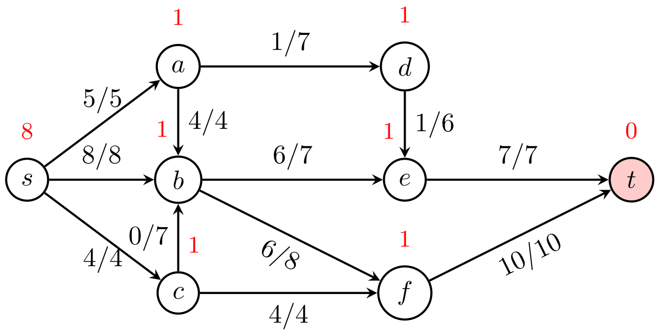
Excès : $e = (0, 0, 1, 0, 0, 0, 9, 7)$.

8. **Itération 7 : décharge de b (plus haut).**



Excès : $e = (0, 0, 0, 0, 0, 0, 10, 7)$.

9. **Itération 8 : décharge de f (seul excédant).**



Excès : $e = (0, 0, 0, 0, 0, 0, 0, 17)$, donc **FIN**.
Le **flot maximal** vaut donc

$$|f| = 17.$$

4.3 Problème de flot à coût minimal

En construction...