Outils pour la logistique

Cours 2 : Modèles de la programmation linéaire et de la programmation linéaire en nombres entiers

---

ECE – 3EME ANNEE
FILIERE *TRANSPORTS ET MOBILITE*

Cédric BENTZ (CNAM)

La programmation linéaire (PL) en 3 mots et 3 slides (1/3)

- La PL est l'un des domaines de la Recherche Opérationnelle (RO) :
  - Origines militaires de la PL : G. Dantzig, conseiller scientifique à la *U.S. Air Force*, met au point la méthode du simplexe pour résoudre des PL en 1947
  - Le terme « programmation » fait ici référence à l'« organisation » d'opérations militaires et à leur « planification » (et non au sens informatique usuel)
  - Technique déjà utilisée, à l'époque, pour organiser la logistique liée à certaines opérations militaires

La programmation linéaire (PL) en 3 mots et 3 slides (2/3)

- Le domaine de la PL s'est depuis étendu à des applications civiles :
  - Applications “traditionnelles” : télécommunications, transport, logistique, finance, génie industriel, etc.
  - Désormais utilisée dans la plupart des domaines industriels : Orange, SNCF, Air France, EDF, etc.
  - Aussi appelé **Optimisation linéaire**, car on optimise une fonction linéaire sous des contraintes linéaires
  - Cas particulier de la programmation mathématique (fonction à optimiser et contraintes quelconques)

La programmation linéaire (PL) en 3 mots et 3 slides (3/3)

- PL = technique de résolution de problèmes :

Modélisation par un Programme Linéaire (PL) :
- Modèle assez générique,
- Permet de capturer de nombreux problèmes concrets.

↓

Résolution par des algo. dédiés de plus en plus efficaces
- Plus efficaces que des algorithmes génériques de Prog. Math.,
- Par ex. : algorithme du simplexe, points intérieurs, ellipsoïdes, etc.

Solveurs rapides maintenant disponibles, basés sur ces algorithmes
- Par exemple : CPLEX, LPSolve, Xpress, GLPK, Excel...
- Intégration aisée dans tout type d'applications

Un exemple introductif (1/4)

- Achat de matières brutes contenant un minerai à extraire (2 types de MB disponibles) :
  - MB1 : taux moyen de minerai = 6kg par tonne
  - MB2 : taux moyen de minerai = 10kg par tonne
- 3 aspects à prendre en compte :
  - Le temps d'extraction,
  - Le volume total de stockage,
  - Le prix.

Un exemple introductif (2/4)

- Temps d'extraction (16 heures disponibles) :
  - MB1 : 10 minutes par tonne
  - MB2 : 30 minutes par tonne
- Au plus 40 tonnes de MB peuvent être stockées
- Prix unitaires (119 000 euros disponibles) :
  - MB1 : 3.50 euros par kg
  - MB2 : 2 euros par kg
- On veut maximiser la quantité de minerai extraite

==> Quelle quantité acheter de chaque MB ?

Un exemple introductif (3/4)

- Formulation du problème
  - **Variables**
    - $x_1 \geq 0$ est le nombre de tonnes de MP1 à acheter
    - $x_2 \geq 0$ est le nombre de tonnes de MP2 à acheter
  - **Objectif** : maximiser la quantité totale de minerai
    - Quantité totale = quantité via MP1 + quantité via MP2
  - **Contraintes de disponibilité sur les ressources**
    - Temps d'extraction disponible,
    - Poids de stockage disponible,
    - Budget disponible.

Un exemple introductif (4/4)

- Modèle mathématique obtenu (PL)

$\max 6x_1 + 10x_2$ — **Maximiser** une fonction (objectif) **linéaire** en x_1 et x_2

sous contraintes :

$x_1 + 3x_2 \leq 96$ (temps) *{en divisant par 10}*

$x_1 + x_2 \leq 40$ (stockage)

$7x_1 + 4x_2 \leq 238$ (budget) *{en div. par 500}*

— **3 contraintes linéaires** en x_1 et x_2

$\underbrace{x_1 \geq 0, x_2 \geq 0}$

Deux variables x_1 et x_2, contraintes à être **positives**

Plus généralement…

- Une fonction objectif (ou *économique*) linéaire à maximiser/minimiser
  - Transformer un max en min ? Et l'inverse ?
  - "Sortir" un coefficient positif de la fonction objectif ?
- Sous un ensemble de contraintes linéaires (inégalités avec $\leq$ ou $\geq$, ou bien égalités)
- Linéarisation (= écriture sous forme linéaire) ?
  – Par exemple, maximiser le min. de 2 valeurs ?

Petit lexique de la PL

- $x_1, x_2, \ldots, x_n$: variables (de décision)
- Solution **admissible** = affectation de valeurs aux x_i **vérifiant les contraintes**
- Région/ensemble/domaine/polyèdre **admissible** = ensemble des solutions admissibles
- Solution **optimale** = solution admissible qui maximise/minimise la fonction objectif

Exemples ?

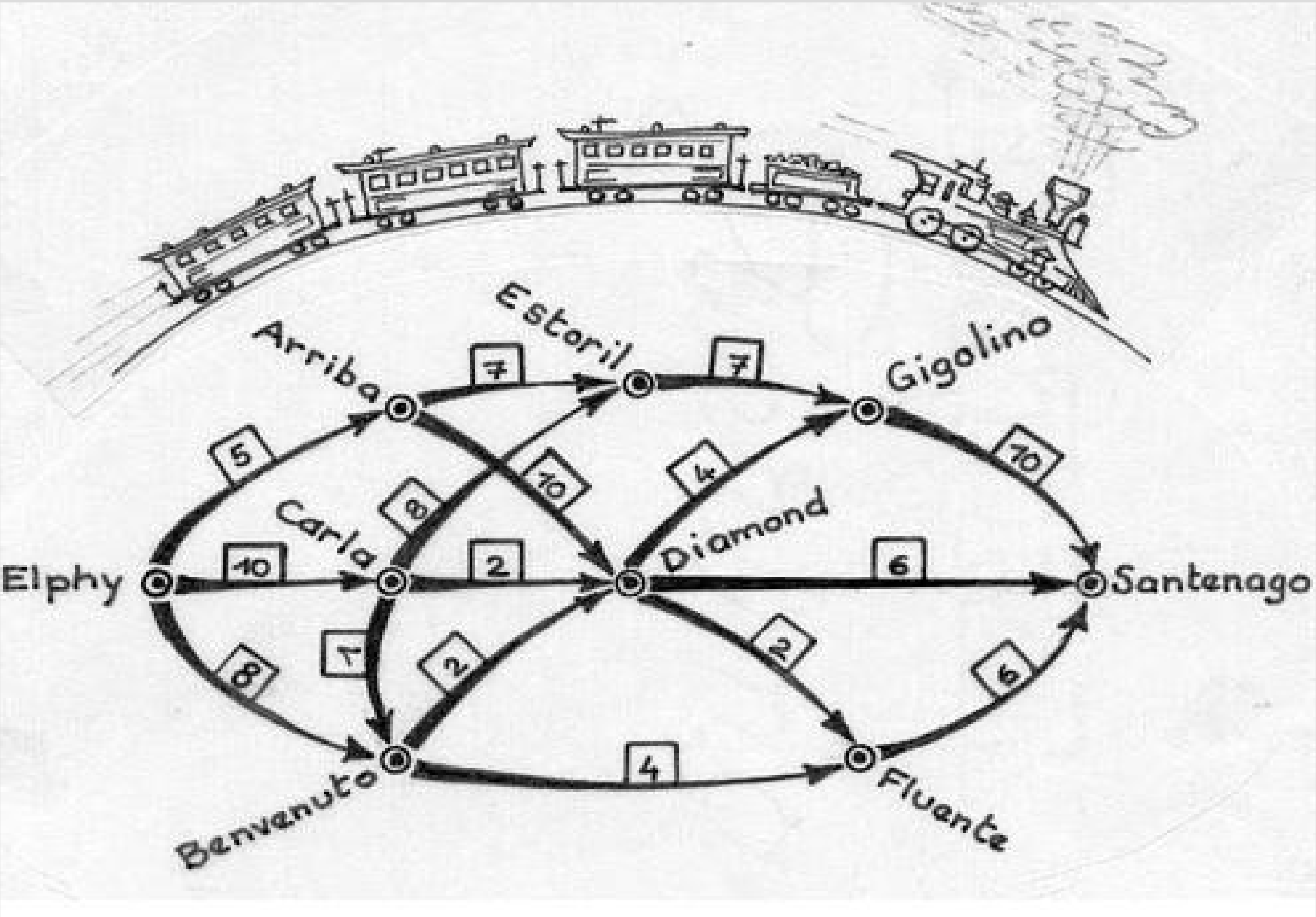
- Problèmes déjà évoqués :
  - Problèmes d’ordonnancement ?
  - Problème du flot maximum ? (cf ex. page suivante)
  - Programmes de transport ? (cf ED)
- Autres problèmes ? planification de production, finances et stratégies d'investissement, etc.

Débit d'un réseau ferroviaire (rappels)

- Sur un réseau ferroviaire, on a indiqué sur chaque tronçon entre 2 villes le nombre maximum de trains qui peuvent passer par jour dans le sens indiqué.
  - Aller de Elphy à Santenago prend moins d'un jour.
  - Chaque jour, il peut partir au plus 23 trains d'Elphy.
- Combien de ces trains, au maximum, peuvent parvenir dans la journée à Santenago ?

(D'après « La vie du rail », 1998.)

Exercice de modélisation PL : flot maximum Elphy-->Santenago



Résolution graphique d'un PL (1/2)

- Méthode dite de **« résolution graphique »**
  - Exploite le fait qu'on a 2 variables : $x_1 \geq 0$ et $x_2 \geq 0$
- Toute inégalité est vérifiée par une moitié du plan $\Rightarrow$ contrainte = demi-plan (dimension 2)
  - Tracé du domaine admissible dans le plan
    - Tracé de la frontière (= *une droite*) du demi-plan défini par chaque contrainte,
    - Pour chaque droite, on sélectionne un demi-plan (celui qui contient ou ne contient pas l'origine, selon le cas),
    - Région admissible = intersection de tous ces demi-plans.
  - Déterminer la *meilleure* des solutions admissibles ?

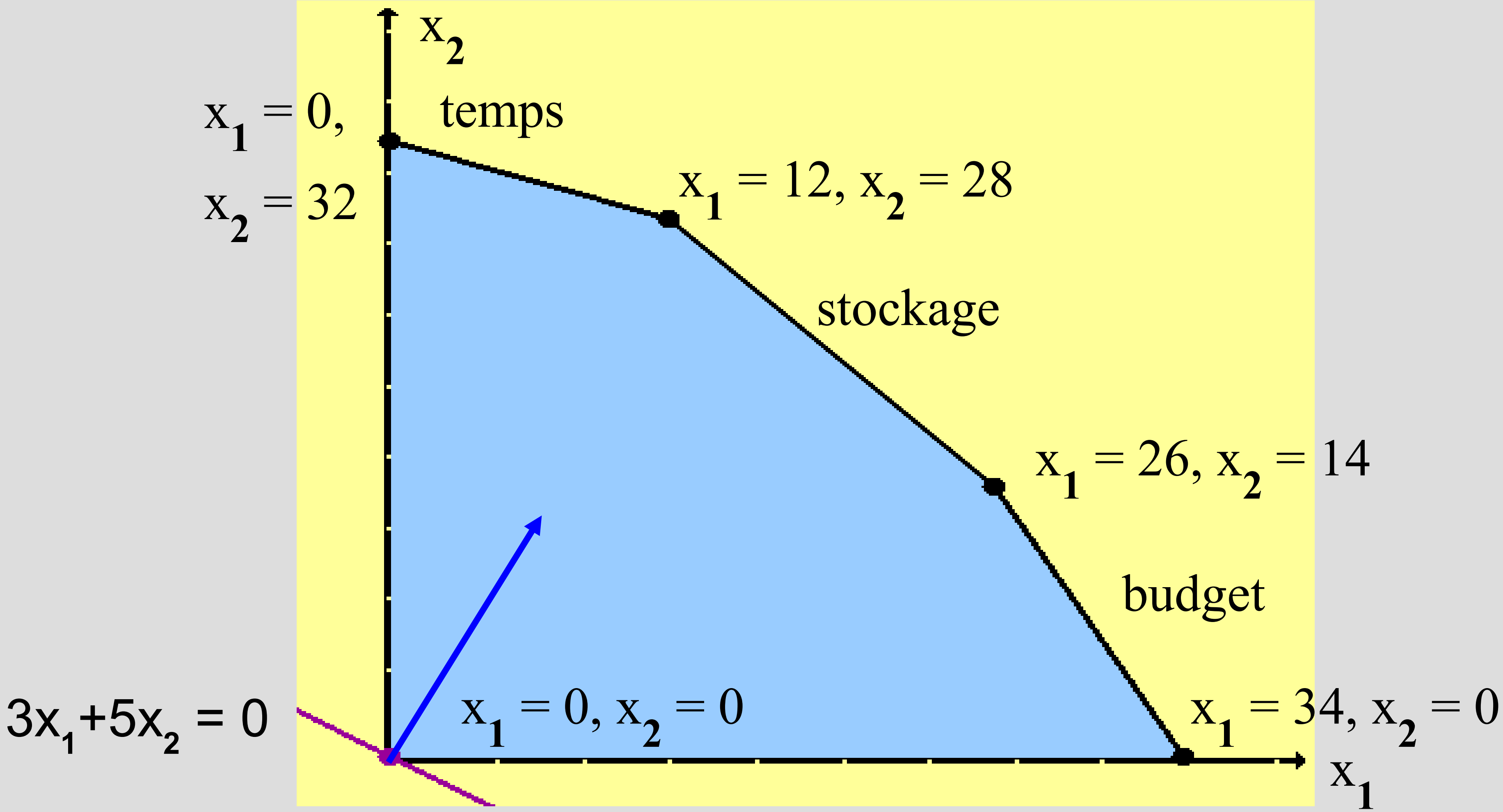
Résolution graphique d'un PL (2/2)

- Prise en compte de la fonction objectif f ?
  - Si la valeur optimale est f*, on appelle **droite optimale** la droite d'équation f(x)=f* :
    - Toute droite d'équation f(x)=constante lui est parallèle,
    - Tracé d'une droite f(x)=v passant par la région admissible,
    - Tracé de la droite f(x)=0 (parallèle à f(x)=v),
    - Si v > 0 et qu'on maximise f, il faut se déplacer de f(x)=0 vers f(x)=v en restant parallèle à f(x)=0, et continuer dans cette direction le plus possible tout en restant admissible,
    - Les 3 autres cas sont similaires...
  - Donc : recherche « visuelle » d'une solution optimale !

Résolution graphique du problème d'achat de matières brutes (1/3)

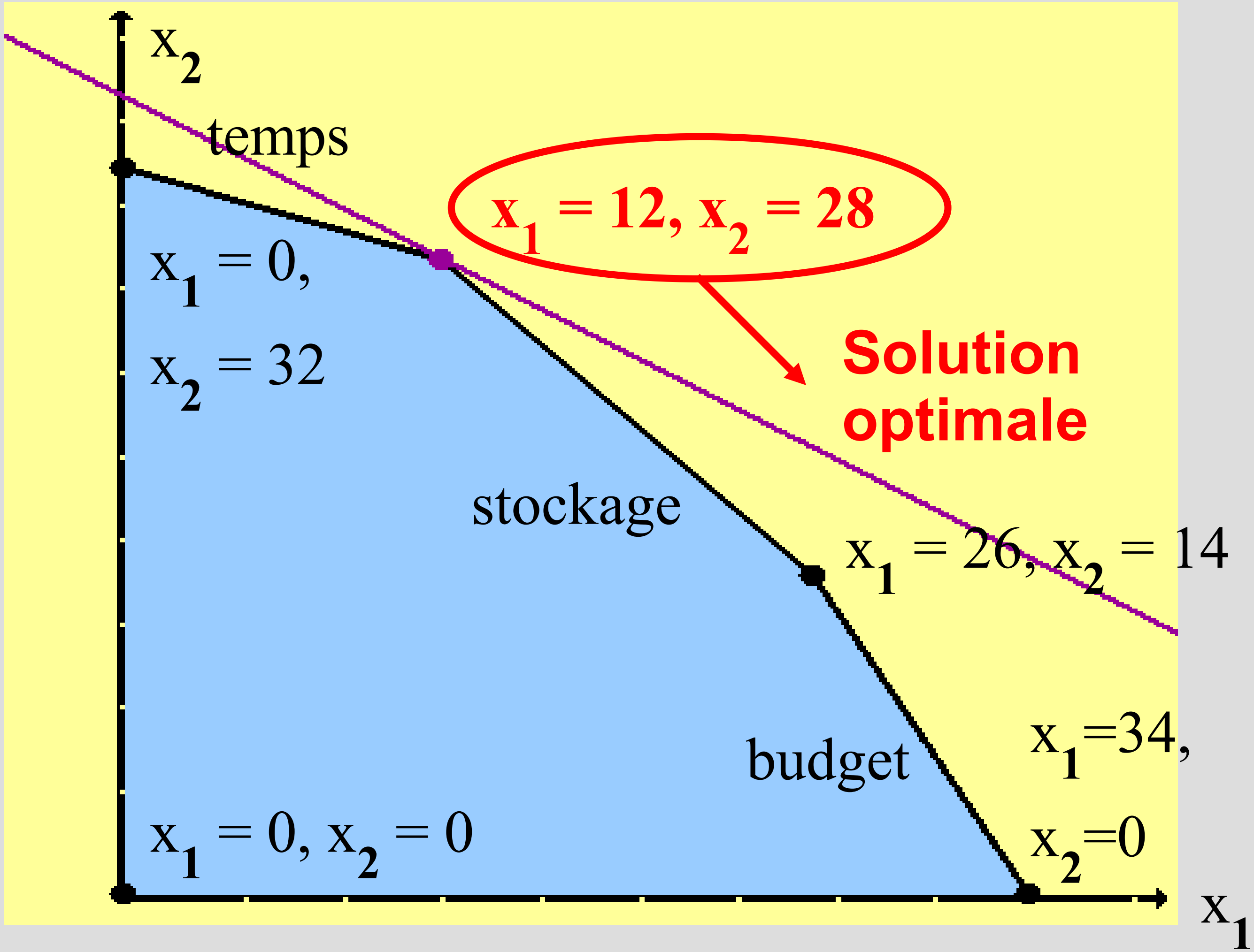
- Dans le domaine admissible (en bleu), on cherche une solution qui maximise $3x_1 + 5x_2$
  - On trace la droite $3x_1+5x_2=0$ (en violet)
    - La droite optimale est $3x_1+5x_2=$*valeur optimale*
    - Cette droite est donc **parallèle** à la droite $3x_1+5x_2=0$
  - Idée (visuelle) : on « déplace » la droite $3x_1+5x_2=0$
    - En la faisant « glisser », le plus possible, dans le bon sens ($3x_1+5x_2=8$ passe par le point $x_1=1$ et $x_2=1$),
    - Tout en restant dans le domaine admissible.

Résolution graphique du problème d'achat de matières brutes (2/3)



Résolution graphique du problème d'achat de matières brutes (3/3)

$\underbrace{6x_1+10x_2 = 352}_{\text{Droite optimale}}$



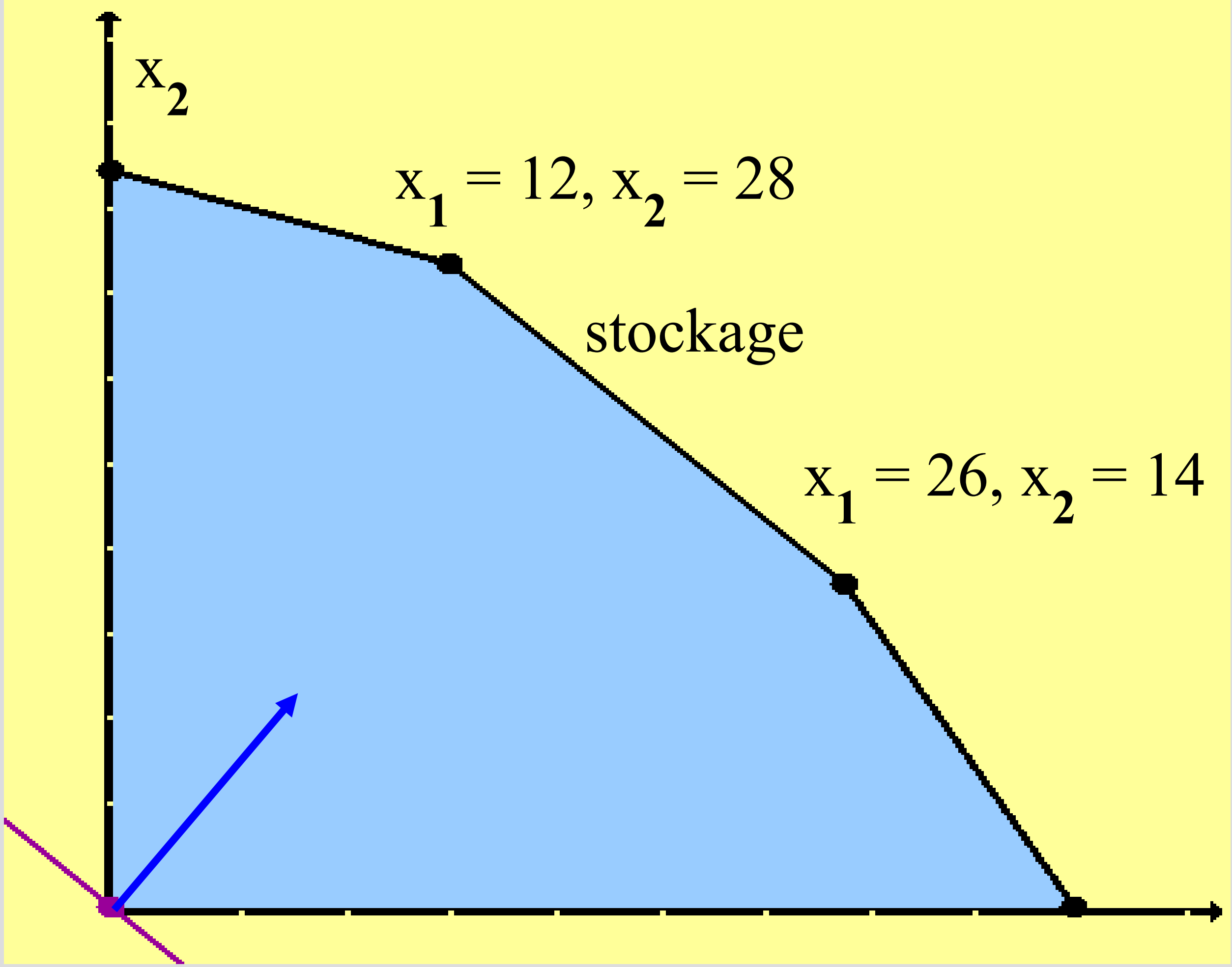
Plan d'achat optimal (unique) :
12 tonnes de MP1,
28 tonnes de MP2,
temps et stockage disponibles utilisés intégralement,
352 kg de minerai

Une deuxième variante (1/2)

- Si taux moyen de minerai dans les MB = 8 kg/t ?

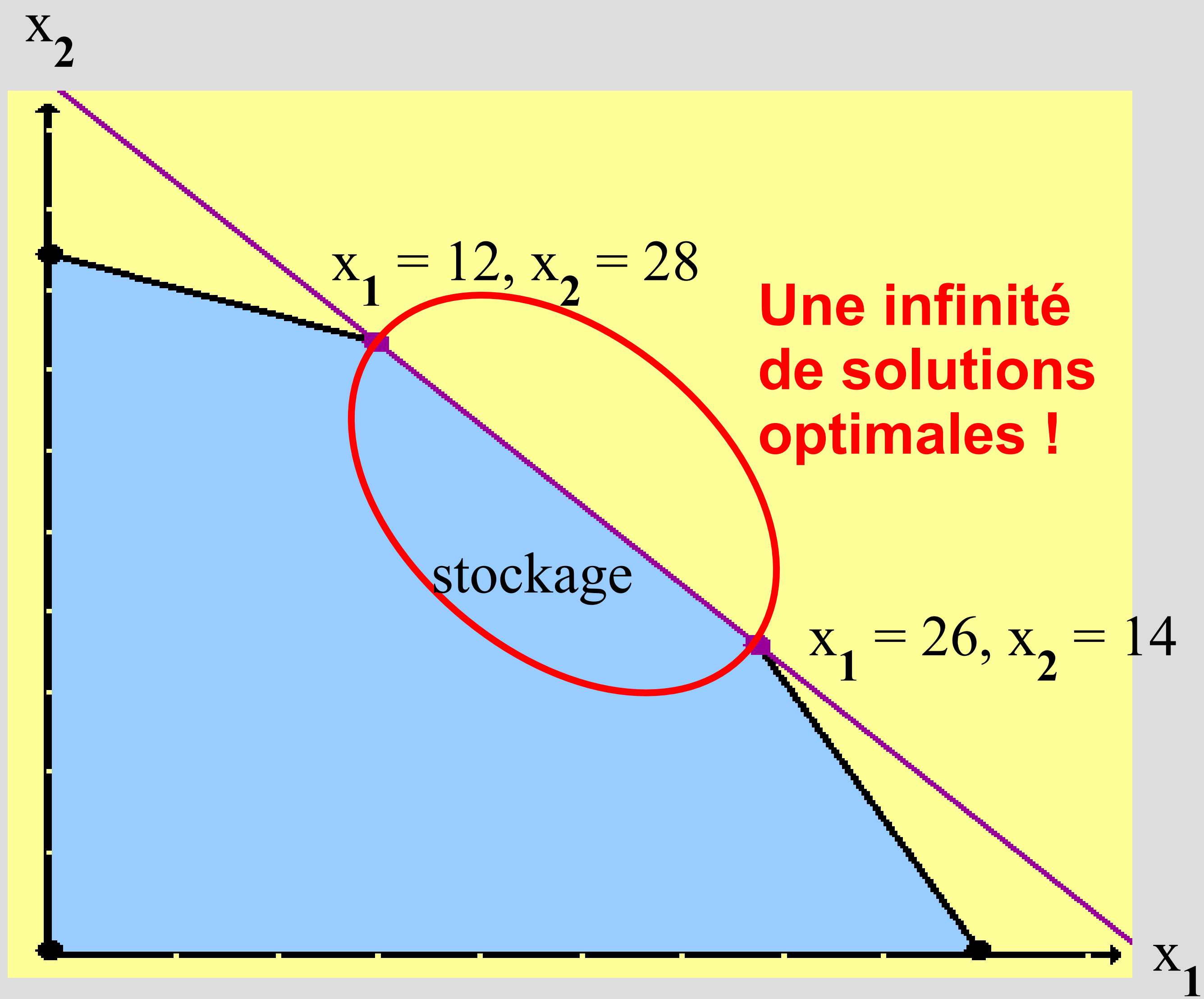
Nouvelle fonction économique :
8 max $x_1 + x_2$

$8\,x_1 + 8\,x_2 = 0$



Une deuxième variante (2/2)

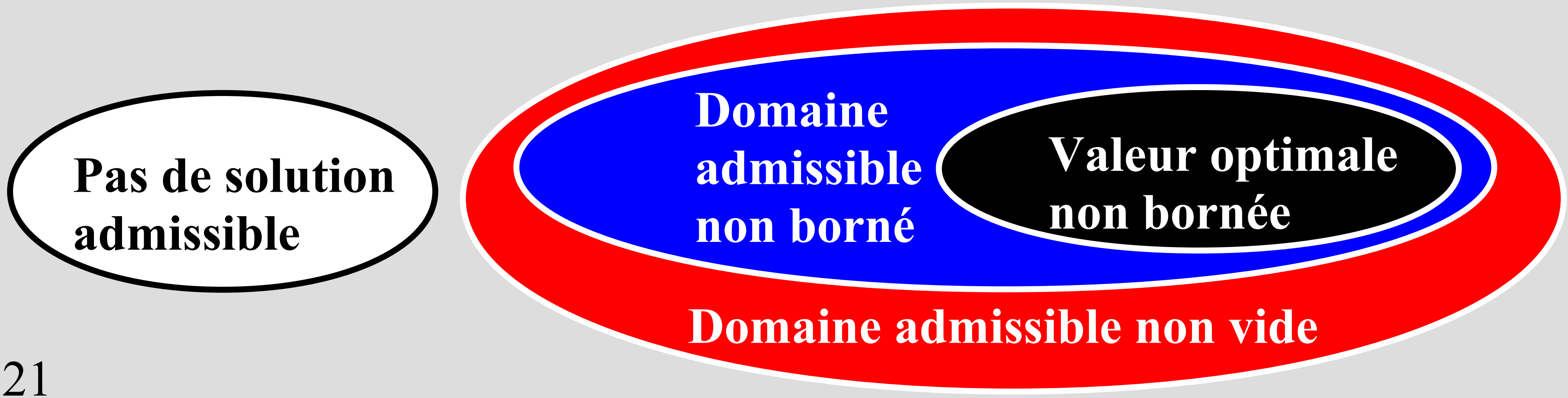
$$\underbrace{8x_1 + 8x_2 = 320}_{\text{Droite optimale}}$$



Plans d'achat optimaux :
stockage disponible intégralement utilisé, 320 kg de minerai.
(Par exemple :
12 tonnes de MP1,
28 tonnes de MP2)

Un premier bilan sur la PL (1/2)

- Ensemble admissible d'un PL ?
  - En dimension 2 (n=2), région du plan bornée par des droites, c'est-à-dire polygone
  - En dimension quelconque, région de IR^n bornée par p **hyperplans** (**polyèdre**)
- Solutions optimales d'un PL ?



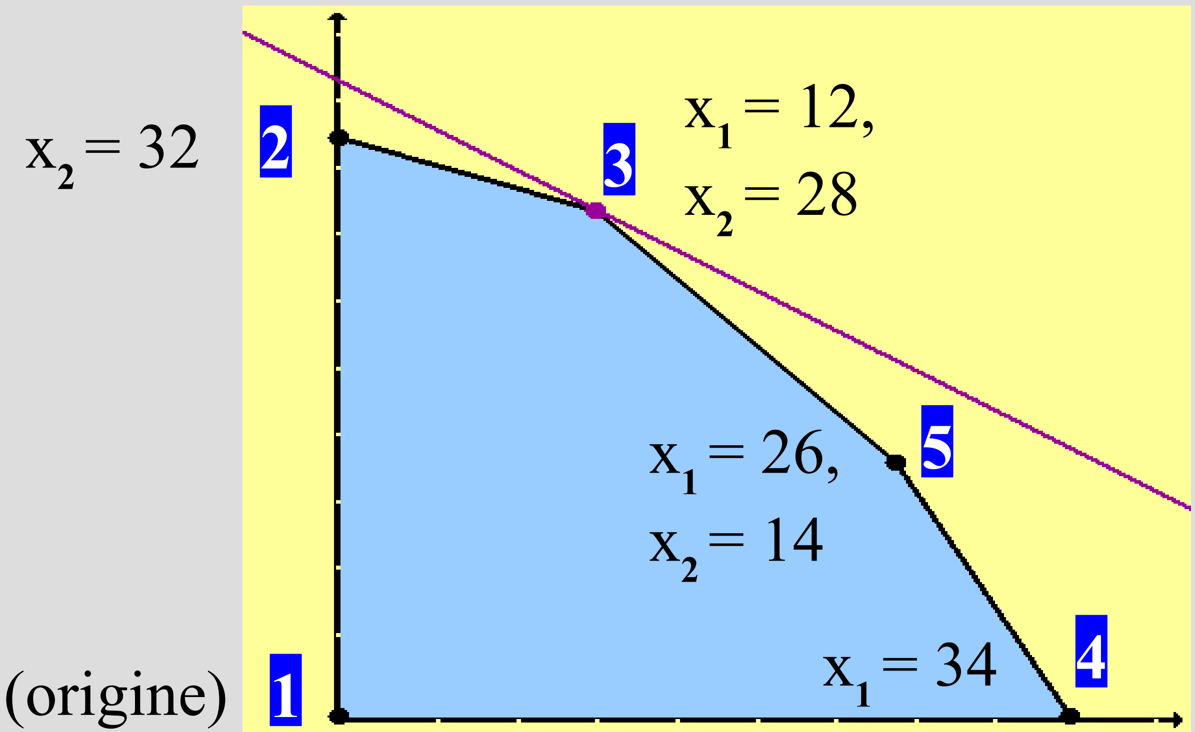
Un premier bilan sur la PL (2/2)

- Si la valeur optimale est bornée, alors, que le domaine admissible soit borné ou non, on a :
  - Soit **une seule solution optimale**
    - En un sommet du polyèdre (polygone si n=2) constituant la région admissible
  - Soit **une infinité de solutions optimales**
    - Si n=2, fonction objectif parallèle à l'une des contraintes ou nulle (mais si n>2, ce n'est plus vrai)
- Dernier cas : l'inverse est-il vrai ?
  - Non, il peut n'y avoir qu'une solution optimale, même si la fonction objectif est parallèle à une contrainte !

Et avec plus de variables ?

- On sait maintenant résoudre des PL si $n = 2$
  - On pourrait imaginer le faire avec $n=3$ variables (résolution graphique en dimension 3)
  - Et avec $n \geq 4$ variables ?
    - Résolution graphique **inapplicable**
    - Nécessité d'une **méthode algébrique** (et automatisée)
    - Différents algorithmes qui peuvent être implémentés dans des solveurs (simplexe, points intérieurs, ellipsoïdes), mais qui dépassent le cadre de ce cours…

Algorithme du simplexe pour la PL



Principe :

On part de l'origine, puis on se déplace de sommet en sommet le long des arêtes du polygone/polyèdre des contraintes, en améliorant à chaque fois la valeur de la fonction objectif, jusqu'à trouver un sommet optimal.

En fonction des choix effectués durant l'algorithme on visite les 3 solutions admissibles (sommets) 1 2 3, ou bien on visite les 4>3 solutions admissibles (sommets) 1 4 5 3 ⇒ en général, quel est le meilleur chemin ? Sa taille ?

Justification « géométrique » de l'algorithme du simplexe (1/4)

- En réalité, on peut montrer que : domaine admissible d'un PL (si borné) = *enveloppe convexe* de ses sommets
- **Ensemble convexe** C : si deux points sont dans C, alors tout le segment qui relie ces deux points est dans C
- **Enveloppe convexe** d'un ensemble de points = plus petit ensemble convexe qui contient tous ces points
- <u>Par exemple</u> : segment pour 2 points, triangle pour 3 points non alignés, etc.

Justification « géométrique » de l'algorithme du simplexe (2/4)

- Un autre résultat essentiel est alors que tout point x d'une enveloppe convexe de points peut s'exprimer comme une combinaison convexe de ses sommets $z_1, \ldots, z_s$
- On a donc : $x=\Sigma_i \lambda_i z_i$, avec $\lambda_i \geq 0$ pour tout i et $\Sigma_i \lambda_i=1$
- Ceci implique que, si l'on maximise une fonction linéaire (ou même convexe) sur le domaine admissible d'un PL, alors au moins un sommet de ce domaine est optimal !

Justification « géométrique » de l'algorithme du simplexe (3/4)

- Pour résoudre un PL, on peut donc théoriquement restreindre l'ensemble (**infini**) des solutions admissibles aux sommets du polyèdre (ensemble **fini**) : on peut tous les énumérer, calculer leur valeur, puis garder le meilleur
- On peut montrer que, s'il y a n variables et m contraintes, il peut y avoir jusqu'à $C^{n}_{n+m} = (n+m)!/(n!m!)$ sommets
- En pratique, c'est donc infaisable : même si m=n=50, on a déjà $C^{50}_{100} \approx 10^{29} = 10^{9}*(3.33*10^{7})*(3\,000*10^{9})$

(Soit 3 000 milliards d'années pour les énumérer tous sur un ordinateur avec un processeur à 1 GHz !!!)

Justification « géométrique » de l'algorithme du simplexe (4/4)

- L'algorithme du simplexe est une façon plus astucieuse d'utiliser ces idées : plutôt que d'énumérer TOUS les sommets, on en parcourt seulement une partie !
- Concrètement, à chaque itération, on passe d'un sommet du polyèdre admissible à un autre, dont la valeur est meilleure (vis-à-vis de la fonction objectif)
- Finalement, quand ce n'est plus possible, on peut montrer qu'on obtient bien une solution optimale...
- Mais à quel point cette stratégie est-elle efficace ?

Efficacité du simplexe

- Si plusieurs « cheminements » possibles pour aboutir à la solution optimale par simplexe
  - Comment trouver le meilleur ? Et sa longueur ?
    - Cf PL précédent : deux, de longueurs respectives 3 et 4
  - Problème : en théorie, l'algorithme peut être obligé de visiter un nb de sommets non polynomial en m et n
    - Cf instances de Klee & Minty
    - Mais il finit : à chaque itération, on explore un sommet meilleur, donc au pire on explore tous les sommets (nb fini)
    - En pratique, il est très utilisé car très efficace : entre 3m/2 et 3m itérations en moyenne (m = nombre de contraintes)

Limites de la PL : un exemple (1/3)

- On peut acheter deux types de boissons, notées b1 et b2, et conditionnées en bouteilles :
  - Chaque bouteille de b1 contient 2 UV de vitamine C, et chaque bouteille de b2 en contient 10 UV.
  - Chaque bouteille de b1 coûte 3 euros, et chaque bouteille de b2 coûte 5 euros.
  - Le budget disponible est de 6 euros.
- On cherche le plan d'achat qui maximise la quantité totale de vitamine C ainsi acquise.

Limites de la PL : un exemple (2/3)

- Solution optimale (méthode graphique) :
  - $x_1=0$, $x_2=1{,}2$ (de valeur 12)
- Impossible : on ne peut pas acheter un cinquième d'une bouteille de b2 !!!
  - On veut donc que les deux variables x_1 et x_2 prennent des valeurs **entières** !
- Solution optimale (par énumération) :
  - $x_1=0$, $x_2=1$ (de valeur 10)
  - Pas un sommet du domaine admissible !

Limites de la PL : un exemple (3/3)

- On doit donc parfois exiger des variables **entières**
- Difficultés ?
  - 1) Le PL {2 max(x_1+5x_2) avec $x_1+2x_2 \leq 2$, $x_1 \geq 0$, $x_2 \geq 0$} contient les mêmes solutions entières que le précédent, mais sa solution optimale est entière !
  - 2) S'il y a 5 UV de vitamine C dans toute bouteille de b1 et 1 UV dans toute bouteille de b2, alors la solution optimale (x_1=2, x_2=0) est entière !
- En général, ce genre de problèmes est beaucoup plus difficile à résoudre (cf suite du cours)...

Quelques familles de PL en variables entières

- Si toutes les variables sont entières, on parle de PLNE = Programme (ou Programmation) Linéaire en Nombres Entiers
- Si les variables entières sont dans {0,1}
  - Variables (bivalentes) 0-1, booléennes, binaires...
  - Equivalence avec des contraintes quadratiques
- Si certaines variables sont réelles/continues = PL en variables mixtes (PLVM)
  - Un PL est donc un PLVM particulier !

Modéliser des alternatives en PLNE

- Le « ET » est *facile* à modéliser en PL
- Mais le « OU » est intrinsèquement non linéaire
- Pourtant, avec des variables 0-1, c'est possible !
- Exemple avec 2 contraintes $x \leq a$ OU $x \geq b$:
  - On suppose x borné : $|x| \leq c$
  - On écrit 2 contraintes (à respecter simultanément), à l'aide d'une variable 0-1 y :
    - $x \leq a + y(c-a)$
    - $x \geq b + (1-y)(-c-b)$

Modélisation PLNE de problèmes classiques en logistique

- Problème de stockage optimal (dit sac à dos)
- Un autre problème de stockage optimal (dit « bin packing »)
- Problème d'affectation de ressources (dit affectation linéaire)
- Problème de tournée (dit TSP, ou voyageur de commerce – VC)

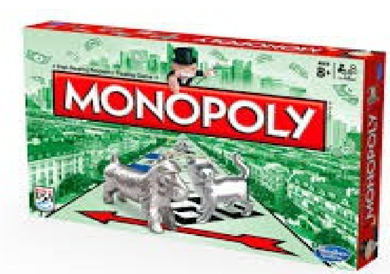
Problème du « sac à dos » (stockage optimal dans un container)

- **Contexte** : on doit décider, parmi plusieurs marchandises, lesquelles stocker dans un container, de façon à maximiser le revenu qui y est associé
- **Formalisation** : chaque marchandise est munie d'un encombrement v_i et d'un revenu r_i, et le volume du container est V
- **Modèle PLNE** ?
  - Une variable 0-1 par marchandise
  - Dépendance couplée ?

V m^3

?

?



r_1 $

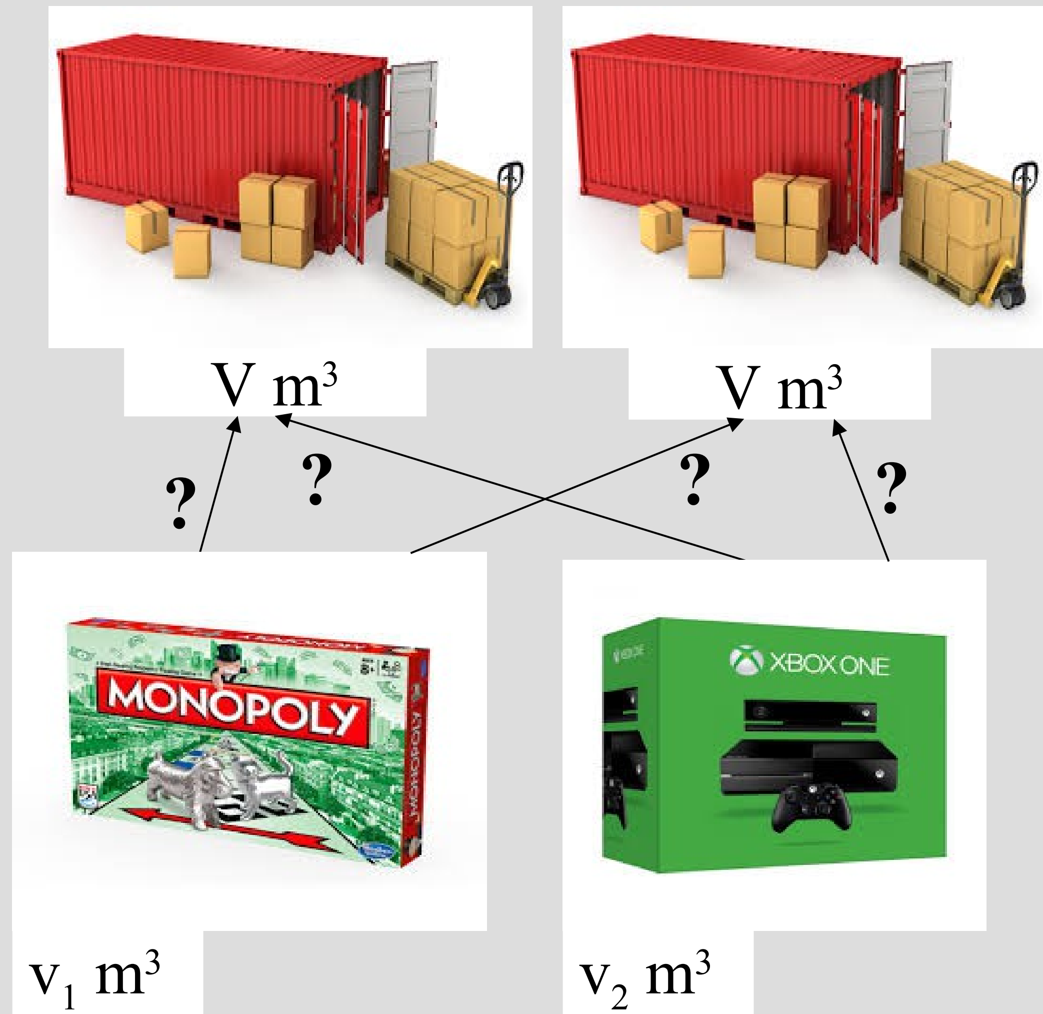
v_1 m^3

r_2 $

v_2 m^3

Problème du « bin packing » (stockage dans un nombre minimum de containers)

- **Contexte** : on a un ensemble de marchandises à stocker, et on souhaite utiliser le moins possible de containers identiques
- **Formalisation** : chacune des n marchandises est munie d'un encombrement v_i, et le volume disponible par container est V
- **Modèle PLNE** ?
  - Une variable 0-1 par paire marchandise-container
  - Une variable 0-1 par container potentiel



Problème d'affectation linéaire

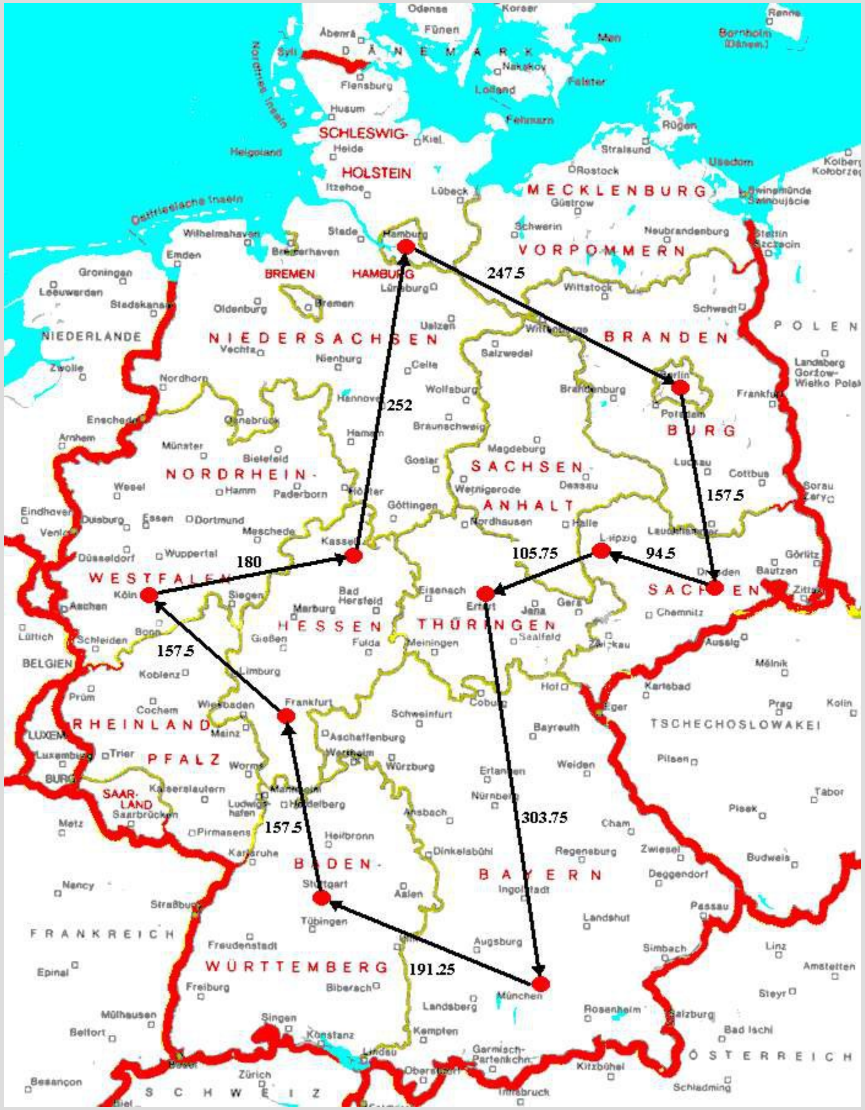
- **Contexte** : dans un entrepôt, n tâches doivent être affectées à n agents de façon à minimiser le temps total (= somme des temps) mis pour effectuer toutes les tâches, sachant que le temps mis par un agent pour effectuer une tâche dépend de la tâche ET de l'agent
- **Formalisation** : chaque paire (tâche, agent) est munie d'une valeur (temps en minutes), et 1 tâche = 1 agent
- **Modèle PLNE** ?
  - Une variable 0-1 par paire (tâche, agent)

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

Agents

Problème de tournée dit TSP ou voyageur de commerce (VC) (1/2)

- **Contexte** : un véhicule de livraison doit passer une et une seule fois par chacun des n clients de sa tournée
- **Objectif** : minimiser la distance totale parcourue
- **Modèle** (graphe) : graphe G=({gares}, A), dont les arêtes A sont pondérées par des distances ; on cherche alors dans G un cycle hamiltonien (CH) de poids minimum



Problème de tournée dit TSP ou voyageur de commerce (VC) (2/2)

- Modèle PLNE ?
  - Une variable 0-1 par arête de G
  - Minimiser la longueur totale du CH, c'est-à-dire la somme des longueurs des arêtes du CH

Deux types de contraintes :

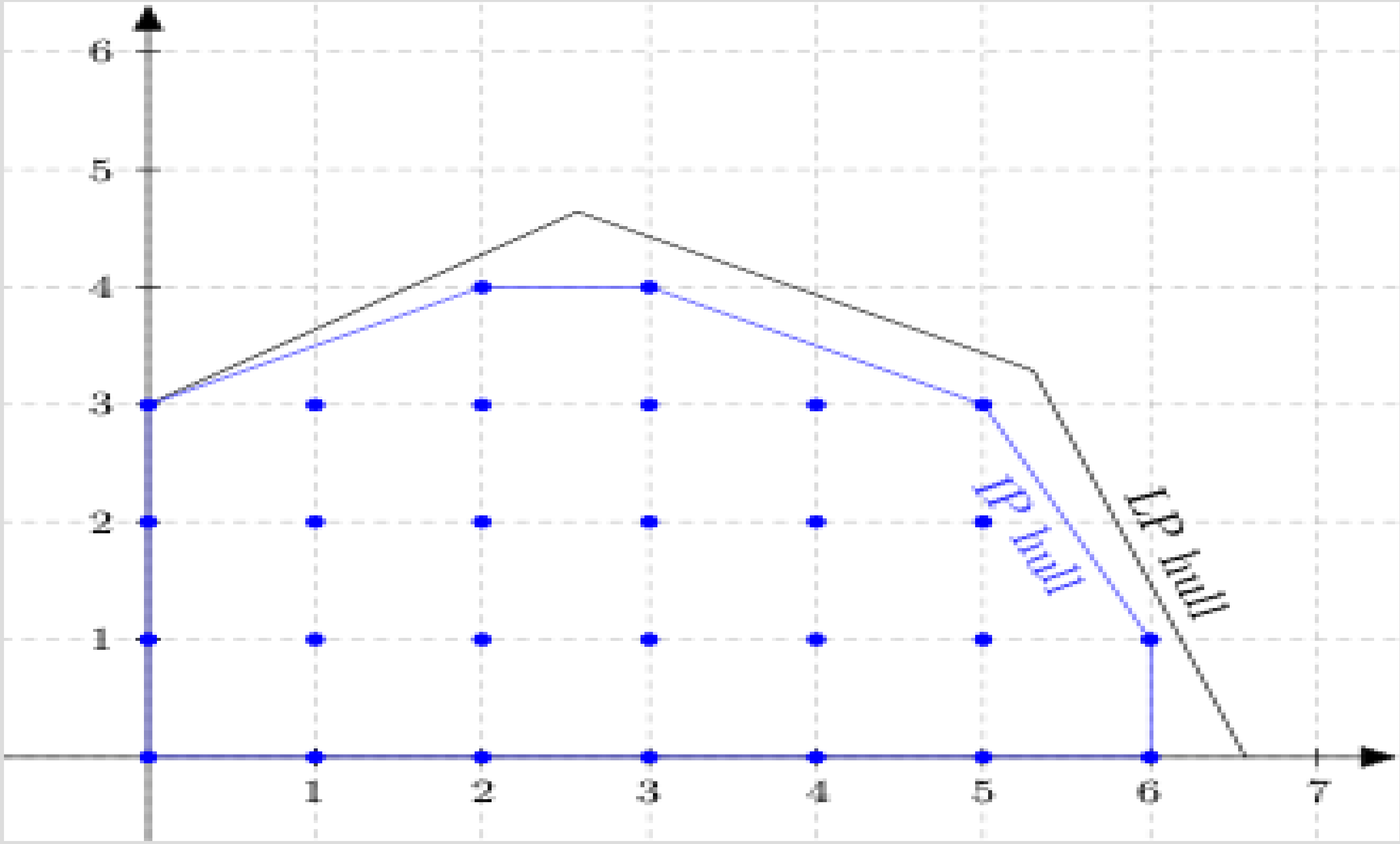
1. Tout sommet est adjacent à deux arêtes du CH
2. Elimination des sous-tours ?

Pour tout sous-ensemble non vide de gares, il y a au moins une arête ayant une extrémité « à l'extérieur »

➔ Nombre de contraintes à ajouter ?

Quelques remarques sur la résolution des PLNE (1/4)

- Problème : le domaine admissible d'un PLNE est un ensemble discret de points, donc non convexe



Quelques remarques sur la résolution des PLNE (2/4)

- Résoudre le PL associé et arrondir la valeur des variables continues de la solution ? ➔ NON !
- 1er contre-exemple :

$\max\{x+10000y, \text{ s.c. } 2x+4y\leq 3, x\geq 0, y\geq 0\}$

  - 3 sommets : (0,0), (3/2,0), (0, ¾)
  - Solution continue optimale : y=3/4, de valeur 7500
  - Solution entière optimale (énumération sur les 2 solutions (0,0) et (1,0)) : x=1, de valeur 1

Quelques remarques sur la résolution des PLNE (3/4)

- 2e contre-exemple :

max $4x+5y$, sous contraintes :

$x+y \leq 4.5$

$2x+y \geq 5.5$

$x \in [0,3]$, $y \geq 1/2$

  - 4 sommets : (1,7/2), (3,1/2), (3,3/2), (5/2,1/2)
  - Solution continue optimale : $x=1$, $y=7/2$ de valeur 21.5
  - Solution entière optimale (énumération sur les 2 solutions (3,1) et (2,2)) : $x=2$, $y=2$ de valeur 18

Quelques remarques sur la résolution des PLNE (4/4)

- Comment résoudre un PLNE ?
  - On ne peut utiliser aucun algorithme de PL (simplexe, points intérieurs, ellipsoïdes, etc.) ➔ plus difficile !
  - Enumérer les solutions admissibles (nombre fini) ?
    - Pour un PLNE avec 100 variables 0-1 : $2^{100} \approx 10^{30}$ solutions potentielles à énumérer (**explosion combinatoire**) !
  - En fait, aucun algorithme efficace connu (NP-difficile) !
- Plusieurs approches d'efficacité variable, complémentaires et imparfaites, peuvent être utilisées (compromis temps/optimalité)

Résoudre, en pratique, des PLNE

- Constat : beaucoup plus difficiles à résoudre que des PL
- Résolution « exacte » basée sur des méthodes de « recherche arborescente » (séparation et évaluation)
  - Lentes, mais permettent d'obtenir une solution optimale
- Résolution « approchée » (on cherche une « bonne » solution, mais pas optimale) à l'aide d'« heuristiques »
  - (+) Plus rapides en pratique
  - (+) Donnent parfois de très bonnes solutions
  - (-) Beaucoup d'heuristiques disponibles, donc le choix de l'heuristique appropriée peut parfois être compliqué à faire
  - (-) Qualité des solutions non garantie *a priori*

Cas « faciles » en PL avec des variables entières (1/3)

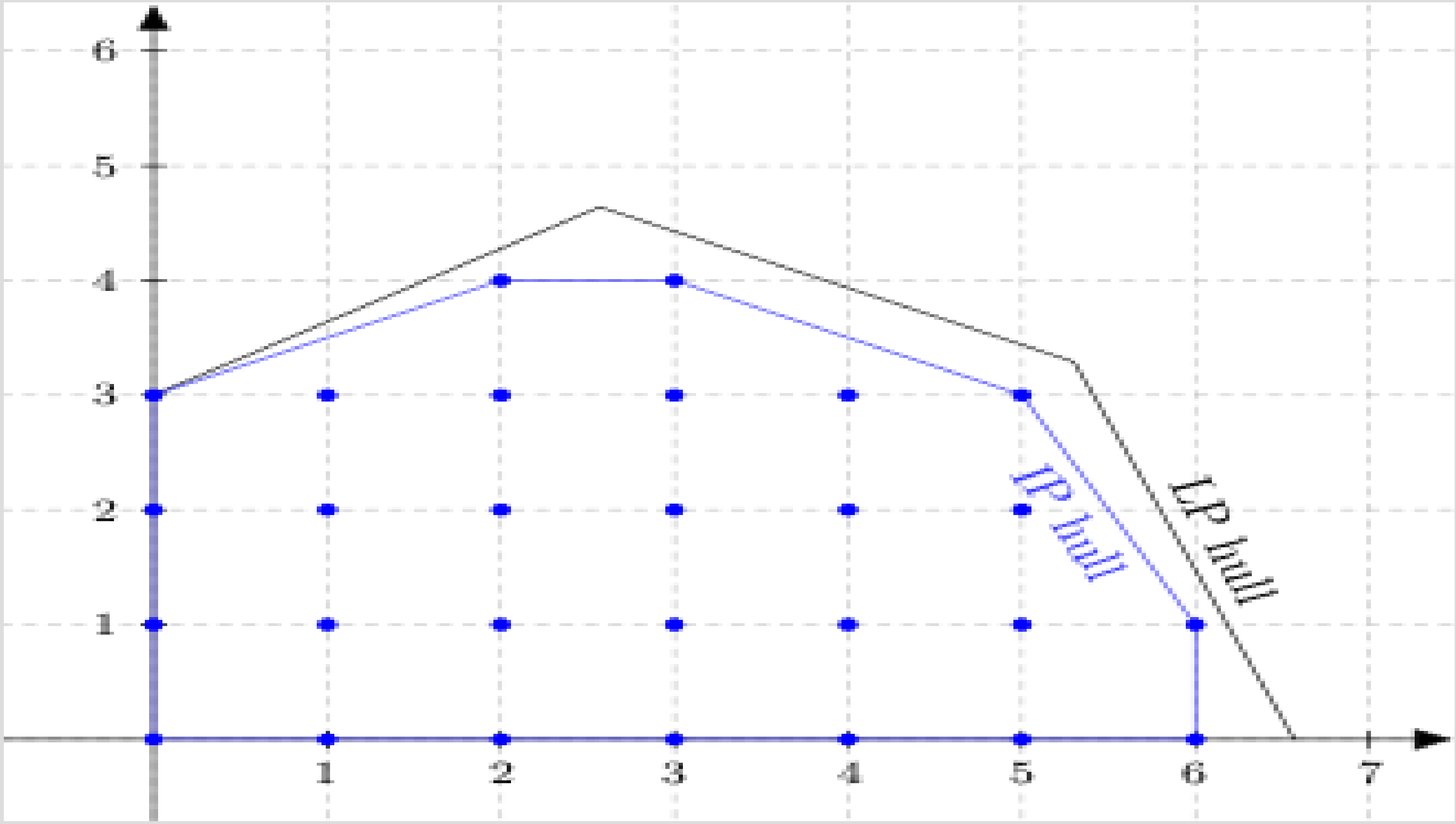
- Certains PL sont “faciles” à résoudre, même si on exige des variables entières :
  - Flot maximum,
  - Plus court chemin avec longueurs ≥ 0,
  - Sélection et répartition de tâches (cf la suite).
- Pourquoi ? Raison possible : la solution optimale du PL associé est en fait toujours entière !

Cas « faciles » en PL avec des variables entières (2/3)

- Quand/comment cela peut-il se produire ?
  - Par exemple, si tous les sommets du polygone (ou polyèdre) admissible ont des coordonnées entières...
- Ce cas correspond en fait à la situation où le domaine admissible du PL est l'enveloppe convexe des points entiers !
  - C'est par exemple le cas si la matrice des contraintes vérifie certaines propriétés (**totale unimodularité**).

Cas « faciles » en PL avec des variables entières (3/3)

- Donc, si la frontière du domaine admissible du PL est en traits bleus et non noirs ==> PLNE “facile”



Une application : exécution de tâches sur des machines (1/6)

- Un problème de sélection de tâches (éléments de production) sur une machine :
  - Un ensemble de n tâches : chaque tâche est représentée par un intervalle de temps $[d_i,f_i]$
  - On dispose d'une seule machine pour les exécuter
  - 2 tâches qui se chevauchent sont **incompatibles**
  - On veut sélectionner (pour les exécuter) un nombre maximum de tâches compatibles
    - Modélisation du problème ?
    - Résolution du problème ?

Une application : exécution de tâches sur des machines (2/6)

- Modélisation PL ?
  - Une variable 0-1 par tâche
  - Une contrainte par paire de tâches incompatibles ?
  - Alternative : découpage du temps en créneaux
    - On note $e_1,...,e_k$ les d_i,f_i classés par ordre croissant ($k \leq 2n$)
    - Une contrainte (au plus une tâche sélectionnée) par créneau $[e_i,e_{i+1}] \Rightarrow$ matrice avec au plus 2n-1 lignes
    - Une tâche traverse des créneaux consécutifs !
  - La matrice des contraintes obtenue est totalement unimodulaire, donc résoudre le PL fournit une solution optimale entière !

Une application : exécution de tâches sur des machines (3/6)

- Méthode de résolution alternative ?
  - En fait, comme beaucoup de problèmes se modélisant par un PL avec une matrice totalement unimodulaire, ce problème admet une méthode de résolution simple
    - Une matrice totalement unimodulaire est donc souvent un « indice » que le problème considéré est « facile »
  - Algorithme glouton spécifique :
    - On suppose les tâches triées par f_i croissants
    - Tant qu'il reste des tâches à sélectionner/exécuter faire
      - Sélectionner la tâche de plus petit indice qui soit compatible avec les tâches déjà sélectionnées
    - Fin tant que

Une application : exécution de tâches sur des machines (4/6)

- A partir de là, on peut en fait résoudre le « vrai » problème, celui de la répartition des tâches :
  - On a un ensemble de tâches (étapes ou éléments de production) à exécuter, à répartir sur le moins de machines possible : comment faire ?
  - Bien sûr, chaque machine ne peut exécuter que des tâches compatibles entre elles...
  - La meilleure stratégie est alors très simple : il s'agit d'affecter les tâches, machine par machine, à l'aide de la méthode précédente !

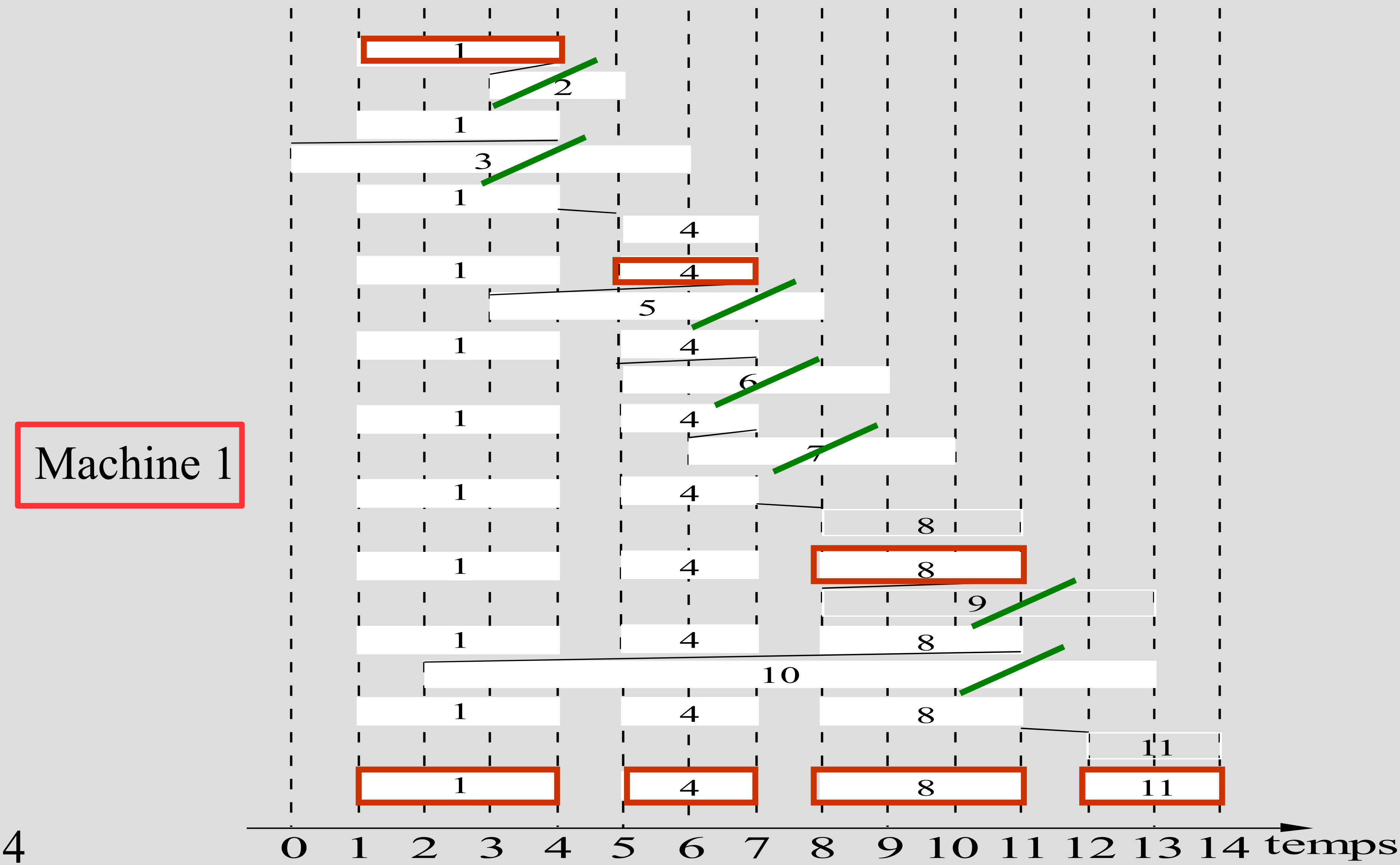
Une application : exécution de tâches sur des machines (5/6)

- Exemple avec 11 tâches :

i	1	2	3	4	5	6	7	8	9	10	11
d_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14

Machine 1 | Machine 2 | Machine 3 | Machine 4 | Machine 5

Une application : exécution de tâches sur des machines (6/6)



Bilan sur les modèles PL et PLNE

- Résolution graphique de PL si n=2 variables, avec différents cas possibles
- Résolution de PL par *solveurs* pour n>2
- PL = modèle puissant, mais ayant des limites :
  - Capture de nombreux problèmes concrets (flots, ordonnancement, transport, etc.)
  - Difficulté pour gérer les variables entières
    - Quelques cas faciles : plus court chemin à longueurs positives, répartition de tâches, etc.
    - Cas plus général de la PL avec variables entières ? Résolution détaillée dans la suite du cours...


"That's all Folks!"