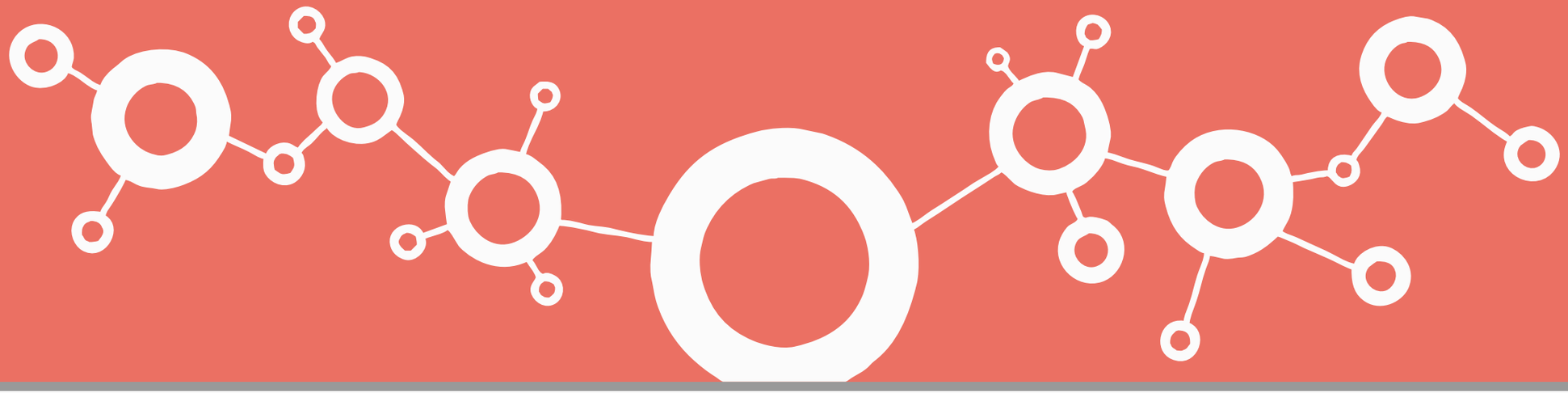


Ch. 13 : Programmation linéaire

2 Un premier problème…

Un constructeur informatique fabrique des cartes mères de deux types :

- Le modèle A nécessite 4 puces de type P1 et 1 puce de type P2
- Le modèle B nécessite 2 puces de type P1 et 3 puces de type P2

Cependant, elle ne dispose que de 1600 puces P1 et 1500 puces P2 par jour.

La carte mère de type A est vendue 350€ et la carte mère de type B est vendue 220 €.

Combien faut-il fabriquer quotidiennement de cartes mères de chaque type pour maximiser les profits de l'entreprise ?

3 Formulation mathématique

Notons x le nombre de cartes mères de type A et y le nombre de cartes mères de type B à produire

- *La carte mère de type A est vendue 350€ et la carte mère de type B est vendue 220 €.*

⇒ on cherche à *maximiser* les profits, càd : $\max 350x + 220y$

- Cependant, on a des contraintes sur les quantités de puces disponibles !
  - *modèle A : 4 puces P1 et 1 puce P2*
  - *modèle B : 2 puces P1 et 3 puces P2*
  - *stock :1600 puces P1 et 1500 puces P2*

$$4x + 2y \leq 1600$$

$$x + 3y \leq 1500$$

$$x \geq 0, y \geq 0$$

Programme mathématique

De manière très générale, un *programme mathématique* est un problème d'optimisation, consistant à trouver le *minimum* ou le *maximum* d'une *fonction objectif* sous certaines *contraintes* :

$$\min c(x, y, z \dots) \quad (\text{ou } \max c(x, y, z \dots)) \qquad \textit{(fonction coût ou fonction objectif)}$$

$$\text{sous contraintes :}$$

$$f_1(x, y, z \dots) \leq 0$$

$$\vdots$$

$$f_n(x, y, z \dots) \leq 0$$

- Historiquement, le terme *programme* provient ici de la notion de *programme militaire*

- Pourquoi peut-on se limiter aux inégalités de type « inférieur ou égal à » ?

Programme linéaire

Etant données des *variables* x, y, z …, une *équation linéaire** est une expression de la forme :

$$ax + by + cz + \cdots + k = 0$$

où les coefficients $a, b, c, \ldots, k$ sont des *constantes*.

💡 Autrement dit, les *variables* ne sont jamais multipliées entre elles (il n'y a pas de $x^2, y^3, xz \ldots$)

Un *programme linéaire* est un programme mathématique où :

- La fonction objectif est linéaire
- Les contraintes sont des inéquations linéaires (💡 jamais d'inéquations *strictes*)

* Il serait plus correct de dire « équation affine »

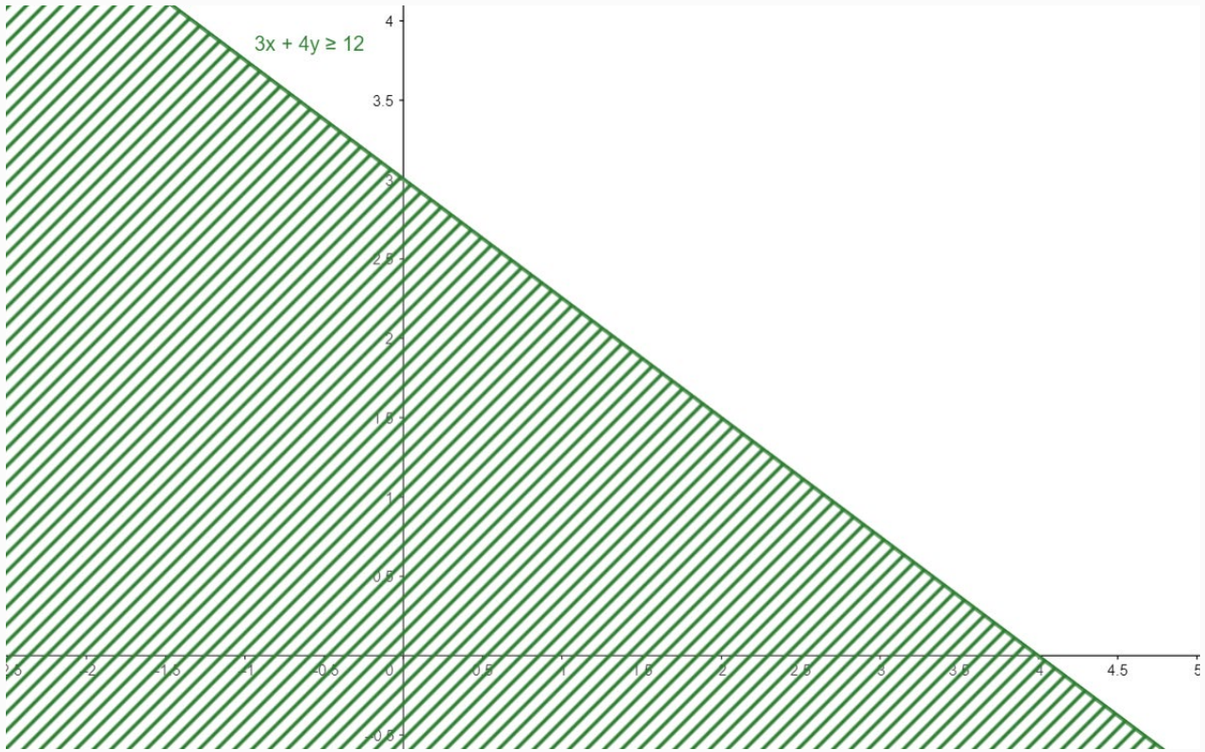
Représentation graphique

Quand on n'a que deux variables, on peut représenter le programme linéaire *graphiquement*

Rappels : en 2 dimensions

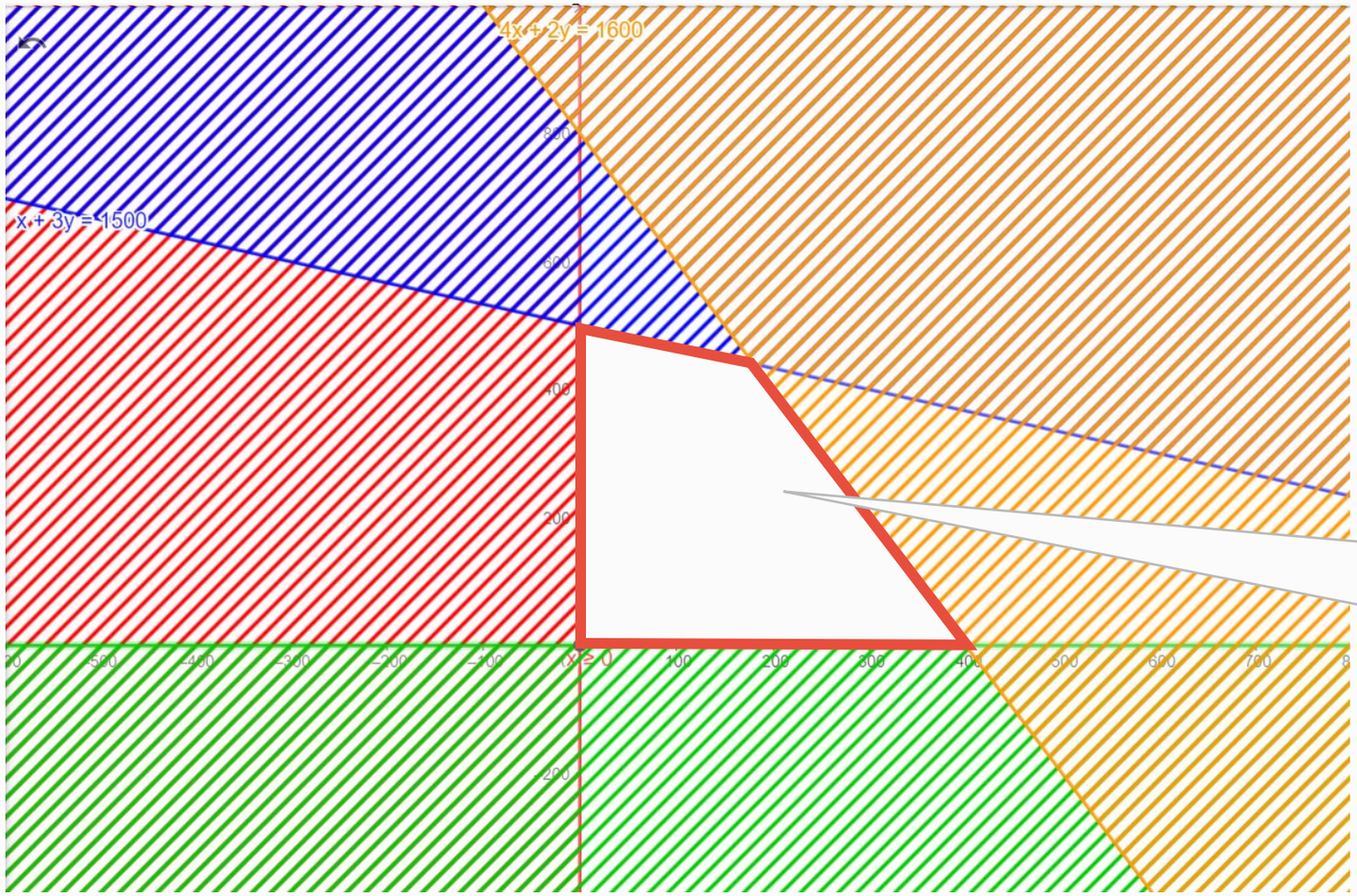
- une équation linéaire définit une *droite*
- une inéquation linéaire définit un *demi-plan*

Exemple : la partie *non hachurée* correspond au demi-plan d'équation $3x + 4y \geq 12$



Représentation graphique

Quand on n'a que deux variables, on peut représenter le programme linéaire *graphiquement* :



$x \geq 0$

$y \geq 0$

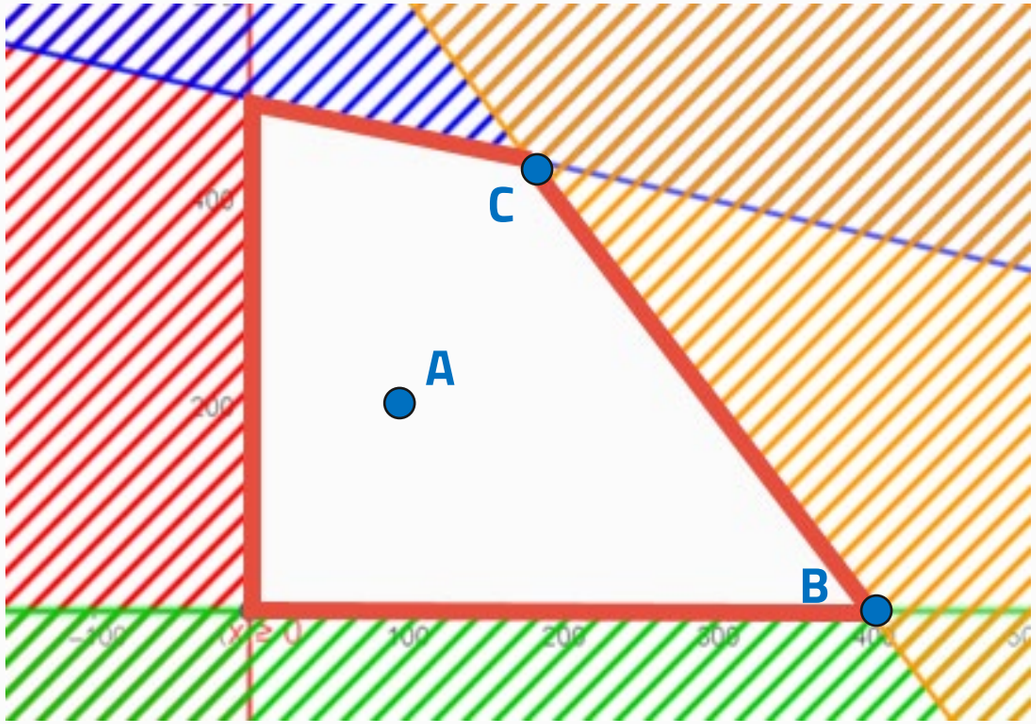
$4x + 2y \leq 1600$

$x + 3y \leq 1500$

Polygone des solutions réalisables

Représentation graphique

Chaque point du polygone correspond à une *solution réalisable* (càd qui satisfait toutes les contraintes) et donne une valeur à la fonction objectif $c(x, y) = 350x + 220y$



- Point A(100, 200)

$c(100, 200) = 35000 + 44000 = 79\,000$

- Point B(400, 0)

$c(400, 0) = 350 \times 400 = 140\,000$

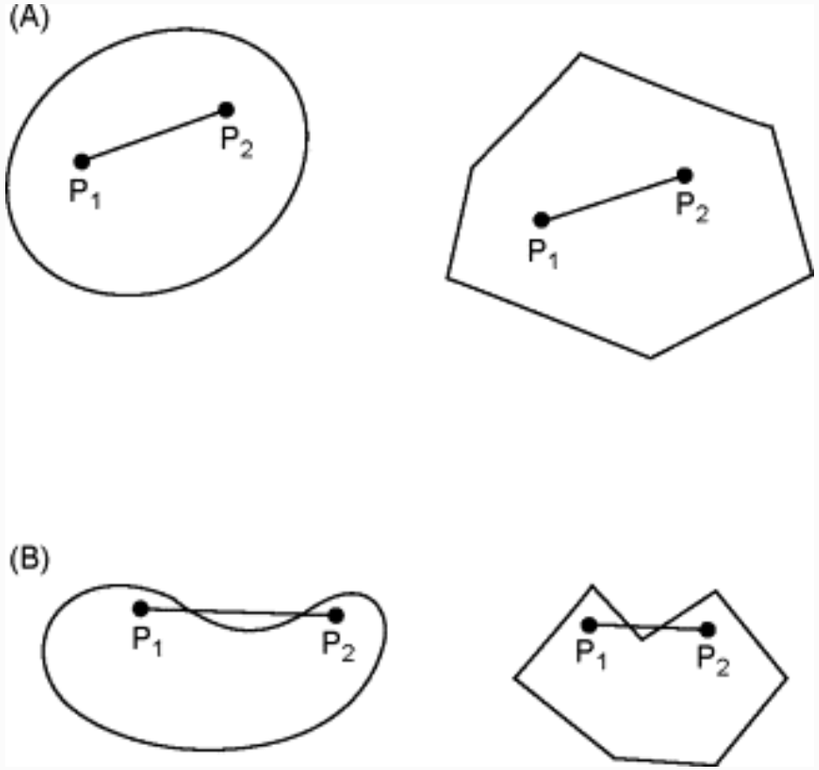
- Point C(180, 440)

$c(180, 440) = 350 \times 180 + 220 \times 440$

$= 159\,800$

Ensemble convexe

Un ensemble *E* est *convexe* si pour tous points P_1 et P_2 de E, tous les points du segment $[P_1, P_2]$ appartiennent à E :



Ensembles convexes

Ensembles non convexes

Ensemble convexe

2 propriétés importantes :

Prop. : un demi-plan est un ensemble convexe

Prop. : l'intersection d'ensembles convexes est un ensemble convexe

Or, le polygone des solutions réalisables est l'intersection de plusieurs demi-plans.

Par conséquent :

le polygone des solutions réalisables est un ensemble convexe

Conséquence : la fonction objectif atteint son minimum / maximum en un *sommet* du polygone

Ensemble convexe

Remarque : dans ce cours, on restera principalement en 2D, mais toutes ces définitions se généralisent aux dimensions > 2, où les contraintes définissent des *demi-espaces*.

Prop. : un *demi-espace* est un ensemble convexe

Prop. : l'intersection d'ensembles convexes est un ensemble convexe

Or le *polyèdre* des solutions réalisables est l'intersection de plusieurs demi-espaces.

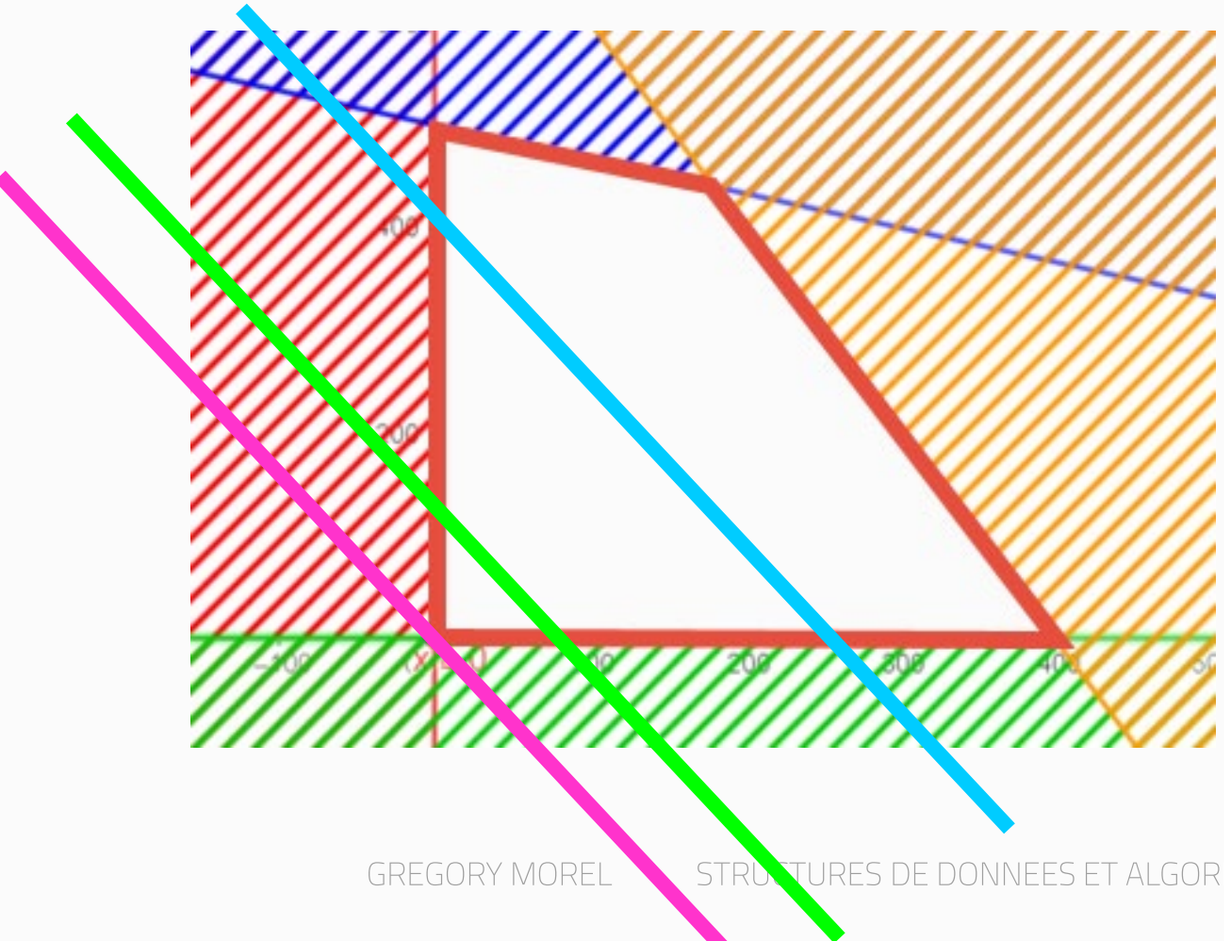
Par conséquent :

le polyèdre des solutions réalisables est un ensemble convexe

Conséquence : la fonction objectif atteint son minimum / maximum en un *sommet* du polyèdre

Interprétation graphique

La droite rose est la droite d'équation $350x + 220y = 0$: elle passe par le point (0,0) et correspond donc à une production de 0 carte mère de type A et 0 carte mère de type B… pour un bénéfice total de 0 €…



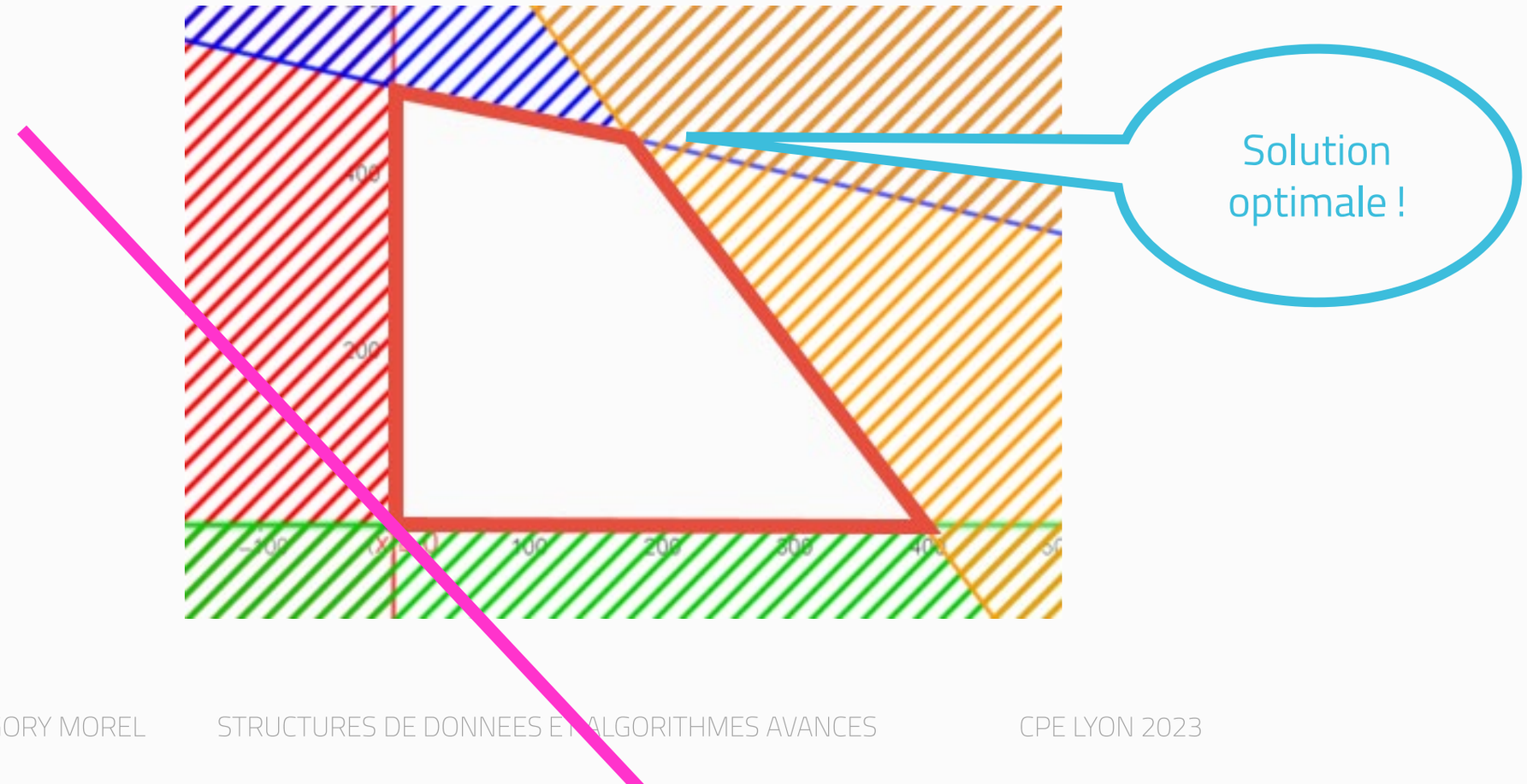
💡 Pour augmenter le bénéfice, par exemple obtenir 30 000 €, il faut tracer la droite d'équation $350x + 220y = 30000$

Idem si on veut 90 000 € : il faut tracer la droite d'équation $350x + 220y = 90000$

Toutes ces droites, d'équation $350x + 220y = c$, sont *parallèles* à la droite $350x + 220y = 0$

Interprétation graphique

Graphiquement, on trace la fonction objectif avec une valeur de 0, puis on la déplace parallèlement à elle-même tant qu'on touche un point du polygone des solutions réalisables



Algorithme du simplexe

💡 Pour trouver l'optimum, il « suffit » donc de calculer la valeur de la fonction objectif en chacun des sommets du polygone et de garder la meilleure solution !

⚠ Problème : il peut y avoir BEAUCOUP de sommets (un nombre *exponentiel*)

💡 En 1947, George Dantzig a mis au point l'*algorithme du simplexe*, qui explore les sommets de manière « intelligente » en passant d'un sommet à un sommet voisin par une règle de *pivotage*

En 2000, la revue *Computing in Science & Engineering* a placé l'algorithme du simplexe dans la liste des 10 algorithmes qui ont eu la plus grande influence en science et ingénierie au XXe siècle

Complexité de la programmation linéaire

L'algorithme du simplexe est très efficace en pratique !

Mais en 1973, Klee et Minty ont construit un « cube » qui oblige le simplexe à parcourir les 2^n sommets : autrement dit, le simplexe est *exponentiel dans le pire cas* :-(

Existe-t-il un autre algorithme, capable de résoudre un programme linéaire en temps polynomial ?

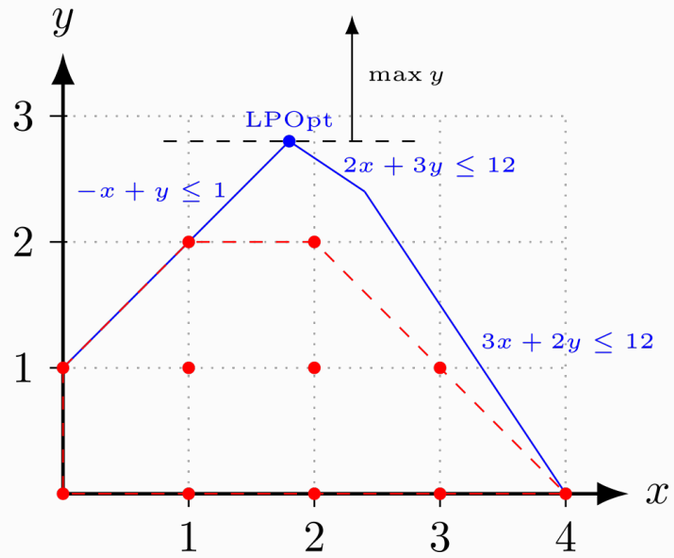
Oui ! En 1979, un chercheur Russe, Leonid Khachiyan, a créé un algorithme *polynomial*

Depuis, de nombreuses améliorations ont été trouvées… mais l'algorithme du simplexe marche si bien en pratique qu'il est encore couramment utilisé !

Programmation linéaire en nombre entiers

Dans le problème des de la fabrication des cartes mères, on cherche une *solution entière* (on ne peut pas fabriquer une fraction de carte mère !)

Pb : les sommets du polyèdre des solutions réalisables n'ont pas toujours des coordonnées entières !



- Contraintes linéaires

LPOpt : solution optimale du programme linéaire

- solution réalisable *entière*

On peut construire des polyèdres dont la solution optimale entière est aussi éloignée qu'on veut de la solution optimale non entière :-(

La programmation en nombre entiers est un problème plus difficile que la programmation linéaire !