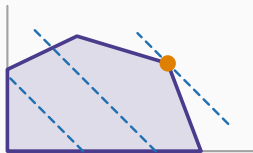


Recherche opérationnelle

Chapitre 2 : Modéliser un problème de décision



Traduire une phrase en inégalité

Les trois formes de contraintes

Le modèle complet de l'atelier

Les hypothèses cachées de la programmation linéaire

Ce qu'il faut retenir

Traduire une phrase en inégalité

À se poser dans cet ordre

1. Qu'est-ce que je choisis? — ce sont les variables.
2. Qu'est-ce que je veux? — c'est l'objectif.
3. Qu'est-ce qui m'en empêche? — ce sont les contraintes.
4. Qu'est-ce qui va de soi? — positivité, entiers, capacités nulles.

La quatrième question est la plus oubliée

Un modèle qui autorise une production négative donnera une solution absurde, parfaitement optimale.

Le solveur ne sait pas qu'on ne fabrique pas moins de zéro sac. Il faut le lui dire.

Une règle simple

Une variable par **décision élémentaire**, avec son unité écrite noir sur blanc.

x = nombre de sacs A produits par semaine

y = nombre de sacs B produits par semaine

La faute la plus fréquente

Écrire « x = les sacs A » sans dire ni **nombre**, ni **par semaine**.

Le modèle tourne, le résultat est un chiffre — mais on ne sait pas si c'est une quantité, un chiffre d'affaires ou une durée.

Les trois formes de contraintes

Ce qu'une ressource impose

Contrainte de capacité

Consommation totale $\leq$ ressource disponible.

$$2x + y \leq 100 \quad (\text{heures de decoupe})$$

Contrainte d'exigence

Production totale $\geq$ minimum requis.

$$x \geq 10 \quad (\text{commande ferme d'un client})$$

Contrainte d'équilibre

Ce qui entre est égal à ce qui sort.

$$x_{\text{entree}} = x_{\text{sortie}}$$

Trois mots à repérer dans l'énoncé

« On dispose de », « au plus », « la capacité est de » → **capacité**, sens $\leq$.

« Il faut au moins », « le contrat impose » → **exigence**, sens $\geq$.

« Tout ce qui arrive repart », « le stock final est nul » → **équilibre**, sens $=$.

Une vérification qui ne trompe pas

Les deux membres d'une contrainte doivent avoir la même unité.

$2x + y \leq 100$: à gauche, des heures ; à droite, des heures. Si l'unité ne colle pas, la contrainte est fausse — sans qu'on ait besoin de réfléchir au fond.

Le modèle complet de l'atelier

Le programme

$$\max Z = 30x + 25y$$

$$\text{sous } \begin{cases} 2x + y \leq 100 & (\text{decoupe}) \\ x + y \leq 80 & (\text{couture}) \\ x \geq 0, y \geq 0 \end{cases}$$

Lire le modèle à voix haute

Maximiser la marge, sachant que la découpe demande deux heures par sac A et une par sac B pour cent heures disponibles, et que la couture demande une heure par sac.

Si la lecture à voix haute redonne l'énoncé, le modèle est juste.

Trois tests de bon sens

Une solution évidente est-elle admissible? $x = y = 0$ respecte tout : le domaine n'est pas vide.

Une solution absurde est-elle exclue? $x = -5$ est interdit par $x \geq 0$.

Une contrainte est-elle inutile? Si l'une n'est jamais atteinte, elle ne sert à rien — et le signaler est une information.

Un cas où la vérification sauve

Si l'on avait écrit $2x + y \leq 100$ pour la couture au lieu de la découpe, la solution serait $x = 20$, $y = 60$ tout de même — mais l'interprétation serait fausse.

Un modèle peut être numériquement correct et logiquement faux.

Les hypothèses cachées de la programmation linéaire

Ce que « linéaire » suppose

Proportionnalité — deux fois plus de sacs demandent exactement deux fois plus d'heures.

Additivité — les deux modèles n'interagissent pas : pas d'économie à les produire ensemble.

Divisibilité — on peut produire 20,5 sacs.

Quand elles tombent

La **proportionnalité** tombe s'il y a des rendements d'échelle ou un temps de réglage fixe.

La **divisibilité** tombe pour un avion, une usine, un employé. C'est le sujet du chapitre 15.

La contrainte oubliée

On modélise les heures et l'on oublie le cuir. Le solveur propose de produire trois fois plus que le stock ne permet.

Une solution optimale d'un modèle incomplet n'est optimale de rien.

L'objectif mal choisi

Maximiser le **chiffre d'affaires** au lieu de la **marge** conduit à produire le bien le plus cher, même s'il coûte plus qu'il ne rapporte.

L'objectif doit être ce qu'on veut vraiment, pas ce qui est le plus facile à mesurer.

Ce qu'il faut retenir

Six points

1. Quatre questions dans l'ordre : **que choisis-je**, **que veux-je**, **qu'est-ce qui m'en empêche**, **qu'est-ce qui va de soi**.
2. Une variable est définie avec son **unité** et sa **période**.
3. Trois formes de contraintes : **capacité** $\leq$, **exigence** $\geq$, **équilibre** $=$.
4. Les deux membres d'une contrainte ont **la même unité** : c'est le test le plus rapide.
5. La programmation linéaire suppose **proportionnalité**, **additivité**, **divisibilité**.
6. Contrainte oubliée et objectif mal choisi sont les **deux erreurs coûteuses**.

La suite

Le chapitre 3 met ce modèle en forme normalisée — celle que tout algorithme attend — et introduit le vocabulaire du polyèdre des solutions.