Programmes linéaires

Algotithme du simplexe I

Bashar Dudin

May 28, 2018

EPITA

Optimisation linéaire sous contraintes polyédrales

Géométrie de l'espace admissible

On considère le programme linéaire L dans sa forme standard:

$$
\begin{array}{ll}
\text{maximiser} & x_1 + 2x_2 \\
\text{sujet à} & \\
& x_1 + x_2 \leq 5 \\
& -2x_1 + x_2 \leq 3 \\
\text{avec} & x_1, x_2 \geq 0
\end{array}
$$

L'ensemble des solutions admissibles de L est la région du plan dans le premier cadran délimité par les équations $x_2 = 5 - x_1$ and $x_2 = 3 + 2x_1$.

Géométrie de l'espace admissible

L'ensemble des solutions admissibles de L jouit d'une propriété remarquable qu'on appelle la *convexité*.

Géométrie de l’espace admissible

L’ensemble des solutions admissibles de L jouit d’une propriété remarquable qu’on appelle la *convexité*.

Définition

Un sous-ensemble A de $\mathbb{R}^n$ est dit ***convexe*** si tout segment qui relie deux points quelconques de A est inclus dans A.

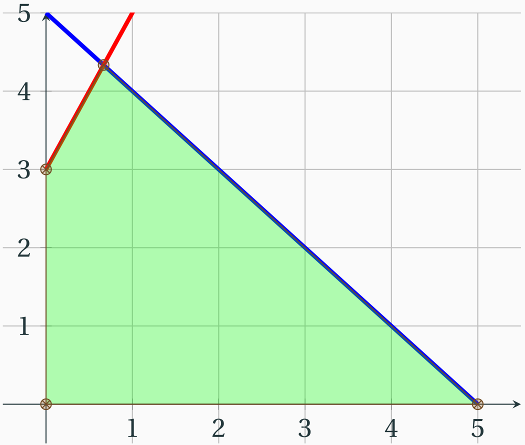
Géométrie de l'espace admissible

L'ensemble des solutions admissibles de L jouit d'une propriété remarquable qu'on appelle la *convexité*.

Définition

Un sous-ensemble A de $\mathbb{R}^n$ est dit ***convexe*** si tout segment qui relie deux points quelconques de A est inclus dans A.

Dans les cas des programme linéaire on travaille avec un cas particulier des parties convexes appelée ***polyèdre***. C'est une intersection de demi-espaces.



Géométrie de l'espace admissible

Proposition

Un programme linéaire a un espace admissible sous-jacent qui est convexe.

Définition

On appelle ***intérieur*** d'une partie convexe A l'ensemble des points $x \in A$ pour lesquels il existe un *disque* de rayon strictement positif centré en x et contenu dans A. Tout point de A qui ne satisfait cette condition est appelé un ***point du bord***. Le ***bord*** de A est l'ensemble des points du bord (potentiellement vide).

Les points du bord d'un polyèdre (défini par des contraintes linéaires), se situent sur les hyperplans obtenus en remplaçant les inégalités des contraintes par des égalités. C'est clairement le cas dans notre exemple bidimensionnel.

Les points optimaux sont des points du bord

On rappelle qu'une fonction $f : A \to \mathbb{R}$ est dite ***bornée*** s'il existe un réel positif M tel que: pour tout $x \in A$, $|f(x)| \leq M$.

Proposition

Soit A l'espace admissible d'un programme linéaire L. Si la fonction objectif est *bornée* sur A alors L a une valeur optimale et elle atteinte sur le bord de A (si non vide).

La justification de ce résultat en dit bien plus!

Les points optimaux sont des points extrémaux

Soit A l'espace admissible d'un programme linéaire L, supposé écrit sous forme standard. Un ***point extrémal*** de A est un point x pour lequel tout *disque* centrée en x ne contient aucun segment (ouvert) centré en x et contenu dans A.

Les points optimaux sont des points extrémaux

Soit A l'espace admissible d'un programme linéaire L, supposé écrit sous forme standard. Un ***point extrémal*** de A est un point x pour lequel tout *disque* centrée en x ne contient aucun segment (ouvert) centré en x et contenu dans A.

De tels points sont nécessairement des points du bord de A ; ils apparaissent comme des intersections d'autant d'hyperplan que la dimension du programme.

Les points optimaux sont des points extrémaux

Soit A l'espace admissible d'un programme linéaire L, supposé écrit sous forme standard. Un ***point extrémal*** de A est un point x pour lequel tout *disque* centrée en x ne contient aucun segment (ouvert) centré en x et contenu dans A.

De tels points sont nécessairement des points du bord de A ; ils apparaissent comme des intersections d'autant d'hyperplan que la dimension du programme.

Proposition

Si la fonction objectif d'un programme linéaire L est bornée sur A alors L a une valeur optimale atteinte en un point extrémal de A (s'il y en a un).

Les points optimaux sont des points du bord | Recherche Naive

Soit L un programme linéaire de dimension n. Une approche naive pour rechercher les points optimaux est d'étudier l'ensemble des points extrémaux de L, et d'y comparer les valeurs objectifs.

Les points optimaux sont des points du bord | Recherche Naive

Soit L un programme linéaire de dimension n. Une approche naive pour rechercher les points optimaux est d'étudier l'ensemble des points extrémaux de L, et d'y comparer les valeurs objectifs.

Recercher un point extrémal revient à résoudre un système linéaire ayant n équations.

Les points optimaux sont des points du bord | Recherche Naive

Soit L un programme linéaire de dimension n. Une approche naive pour rechercher les points optimaux est d'étudier l'ensemble des points extrémaux de L, et d'y comparer les valeurs objectifs.

Recercher un point extrémal revient à résoudre un système linéaire ayant n équations.

Question

Soit $n \in \mathbb{N}^*$. On consdière les contraintes linéaires

$$\forall i \in \{1, \ldots, n\}, \quad 0 \leq x_i \leq 1.$$

Quel est le nombre de points extrémaux que définissent ces contraintes?

Les points optimaux sont des points du bord | Recherche Naive

Soit L un programme linéaire de dimension n. Une approche naive pour rechercher les points optimaux est d'étudier l'ensemble des points extrémaux de L, et d'y comparer les valeurs objectifs.

Recercher un point extrémal revient à résoudre un système linéaire ayant n équations.

Question

Soit $n \in \mathbb{N}^*$. On considère les contraintes linéaires

$$\forall i \in \{1, \ldots, n\}, \quad 0 \leq x_i \leq 1.$$

Quel est le nombre de points extrémaux que définissent ces contraintes?

Comme le montre cet exemple, la complexité au pire est exponentiellement proportionnelle au nombre de contraintes linéaires.

Les points optimaux sont des points du bord | Une meilleure recherche

Au lieu de lister tous les points optimaux potentiels, l'algorithme du ***simplexe*** est une marche le long des points extrémaux de l'espace admissible qui vérifie la propriété :

> *Une itération du* ***simplexe*** *renvoie uen valeur objectif plus grande que la précédente.*

Les points optimaux sont des points du bord | Une meilleure recherche

Au lieu de lister tous les points optimaux potentiels, l'algorithme du ***simplexe*** est une marche le long des points extrémaux de l'espace admissible qui vérifie la propriété :

> *Une itération du **simplexe** renvoie uen valeur objectif plus grande que la précédente.*

Le simplexe nécessite une étape d'initialisation qu'on laisse de côté pour l'instant en travaillant sous l'hypothèse suivante:

Hypothèse

Étant donné un programme L sous forme *standard*, on suppose que le vecteur nul est admissible.

Les points optimaux sont des points du bord | Une meilleure recherche

Au lieu de lister tous les points optimaux potentiels, l'algorithme du ***simplexe*** est une marche le long des points extrémaux de l'espace admissible qui vérifie la propriété :

> *Une itération du **simplexe** renvoie uen valeur objectif plus grande que la précédente.*

Le simplexe nécessite une étape d'initialisation qu'on laisse de côté pour l'instant en travaillant sous l'hypothèse suivante:

Hypothèse

Étant donné un programme L sous forme *standard*, on suppose que le vecteur nul est admissible.

Dans l'exemple à venir, en commençant par la solution nulle on marche le long des points extrémaux pour chercher la valeur optimale.

Algorithme du simplexe : premier essai

Algorithme du simplexe : premier exemple pratique

Voici la forme *slack* du programme L de départ:

$$
\begin{array}{lrl}
\text{maximiser} & & x_1 + 2x_2 \\
\text{sujet à} & & \\
& x_3 & = 5 - x_1 - x_2 \\
& x_4 & = 3 + 2x_1 - x_2 \\
\text{avec} & & x_1,\, x_2,\, x_3,\, x_4 \geq 0
\end{array}
$$

La solution admissible nulle de la forme standard de L correspond à la solution $(0,0,5,3)$ de cette forme-ci ; elle est de valeur objectif nulle. Les variables d'écarts x_3 et x_4 quantifie ô combien la solution (x_1, x_2) est loin de faire des inégalités

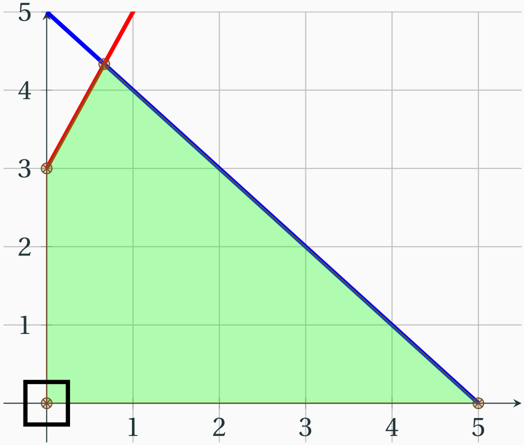
$$x_1 + x_2 \leq 5$$

$$-2x_1 + x_2 \leq 3$$

des égalités.

Algorithme du simplexe : premier exemple pratique

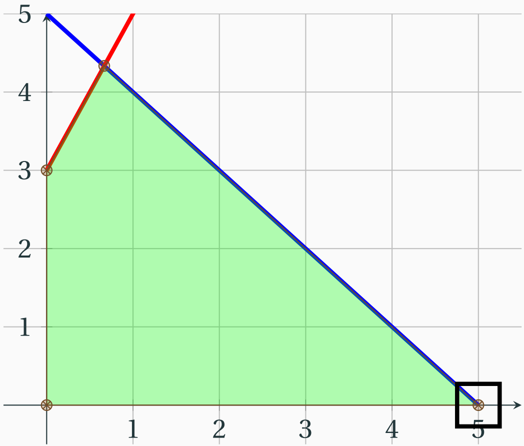
La soluion *slack* $(0,0,5,3)$ a une solution *standard* l'origine du plan. Pour se balader le long du bord du lieu admissible on doit choisir par où aller ; verticalement laissant x_1 à 0 ou horizontalement avec x_2 à 0. Tout choix est bon tant qu'on augmente la valeur objectif. On sera plus précis sur ce choix plus tard. Pour l'instant on choisit d'augmenter x_1, x_2 restant constant.



Algorithme du simplexe : premier exemple pratique

$$
\begin{array}{lll}
\text{maximiser} & x_1 + 2x_2 & \\
\text{sujet à} & & \\
& x_3 = 5 - x_1 - x_2 & (E_1) \\
& x_4 = 3 + 2x_1 - x_2 & (E_2) \\
\text{avec} & x_1, x_2, x_3, x_4 \geq 0 &
\end{array}
$$

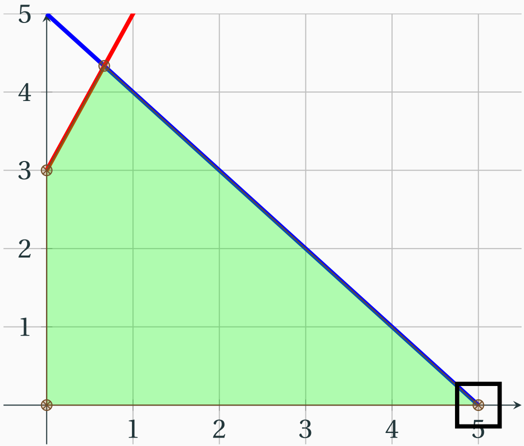
x_1 est contrainte par E_1: l'augmenter indéfiniment violerait les contrainte de positivité. Il n'y a pas de telle contrainte de E_2. La plus grande valeur possible pour x_1 est donc obtenue quand x_3 vaut 0. La solution obtenue ainsi est $(5, 0, 0, 13)$ de valeur objectif 5. Elle satisfait l'égalité definie par la première contrainte.



Algorithme du simplexe : premier exemple pratique

$$
\begin{array}{lll}
\text{maximiser} & x_1 + 2x_2 & \\
\text{sujet à} & & \\
& x_3 = 5 - x_1 - x_2 & (E_1) \\
& x_4 = 3 + 2x_1 - x_2 & (E_2) \\
\text{avec} & x_1, x_2, x_3, x_4 \geq 0 &
\end{array}
$$

La solution $(0,0,5,3)$ peut être comprise comme mettant x_1 et x_2 à 0. Pour $(5,0,0,13)$ il s'agit d'en faire autant pour x_3 et x_2. Ce fait est général : un point extrémal correspond au cas où 2 variables exactement sont à 0. Pour conserver la pattern *annuler une variable à gauche* on échange x_1 et x_3 de côtés.

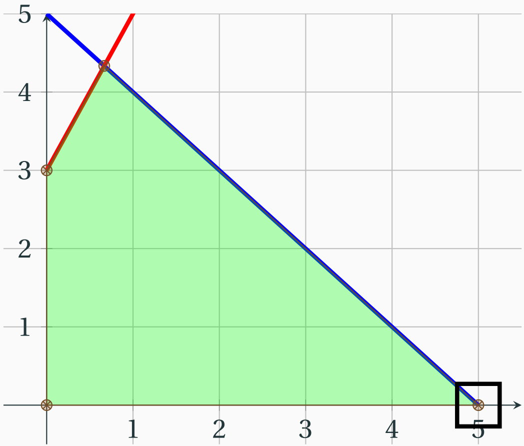


Algorithme du simplexe : premier exemple pratique

Pour obtenir la solution $(5, 0, 0, 13)$ on peut exprimer x_1 dans E_1 en fonction du reste. Cette expression remplace x_1 dans les autres contraintes. D'où le programme équivalent

$$
\begin{array}{lrcll}
\text{maximiser} & & & 5 - x_3 + x_2 & \\
\text{sujet à} & & & & \\
& x_1 & = & 5 - x_3 - x_2 & (E_1) \\
& x_4 & = & 13 - 2x_3 - 3x_2 & (E_2) \\
\text{avec} & & & x_1, x_2, x_3, x_4 \geq 0 &
\end{array}
$$

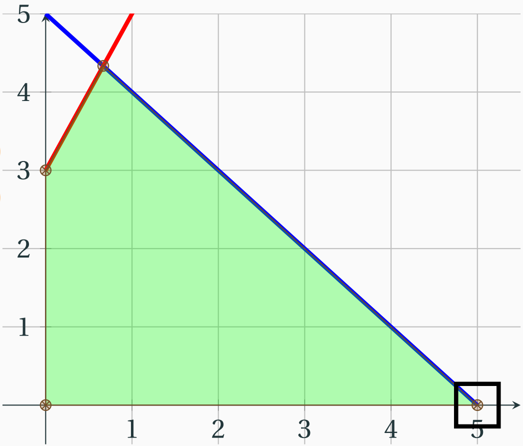
Si l'on annule x_3 et x_2 on retrouve la solution admissible attendue.



Algorithme du simplexe : premier exemple pratique

$$
\begin{array}{lrl}
\text{maximiser} & & \dfrac{28}{3} - \dfrac{5}{3}x_3 - \dfrac{1}{3}x_4 \\
\text{sujet à} & & \\
 & x_1 & = 2/3 - (1/3)x_3 + (1/3)x_4 \qquad (E_1) \\
 & x_2 & = 13/3 - (2/3)x_3 - (1/3)x_4 \qquad (E_2) \\
\text{avec} & & x_1, x_2, x_3, x_4 \geq 0
\end{array}
$$

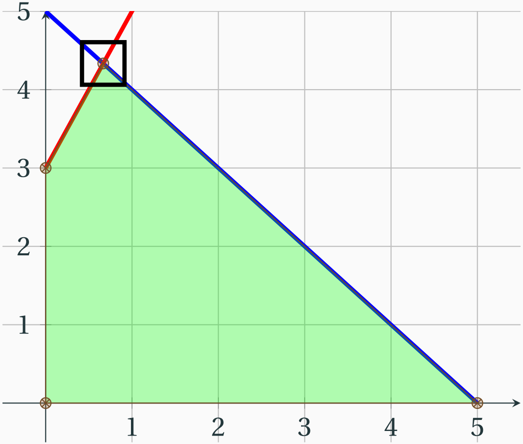
En réitérant la démarche précédente avec x_2 et la seconde équation on obtient le programme ci-dessus. En mettant x_3 et x_4 à 0, on obtient une valeur objectif de 28/3.



Algorithme du simplexe : premier exemple pratique

$$\begin{array}{lrl} \text{maximiser} & & 5 - x_3 + x_2 \\ \text{sujet à} & & \\ & x_1 & = 5 - x_3 - x_2 \qquad (E_1) \\ & x_4 & = 13 - 2x_3 - 3x_2 \qquad (E_2) \\ \text{avec} & & x_1,\, x_2,\, x_3,\, x_4 \geq 0 \end{array}$$

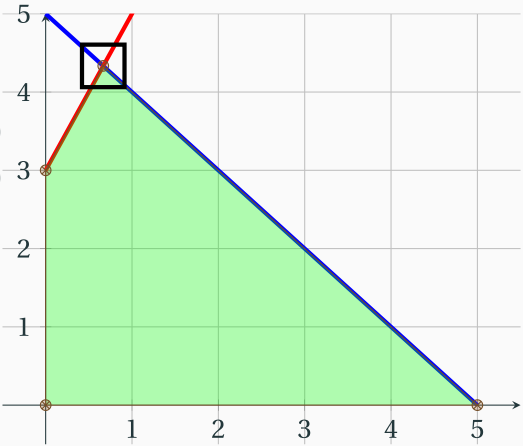
On peut espérer augmenter la valeur objectif via x_2. Toute augmentation de x_3 donnerait une plus petite valeur objectif. L'équation la plus restrictive pour x_2 est E_2.



Algorithme du simplexe : premier exemple pratique

$$
\begin{array}{lll}
\text{maximiser} & \dfrac{28}{3} - \dfrac{5}{3}x_3 - \dfrac{1}{3}x_4 & \\
\text{sujet à} & & \\
& x_1 = 2/3 - (1/3)x_3 + (1/3)x_4 & (E_1) \\
& x_2 = 13/3 - (2/3)x_3 - (1/3)x_4 & (E_2) \\
\text{avec} & x_1, x_2, x_3, x_4 \geq 0 &
\end{array}
$$

On ne peut, désormais, plus espérer augmenter la valeur objectif. Tout augmentation de x_3 ou x_4 diminuerait la valeur objectif. La valeur objectif maximal serait donc $28/3$, atteinte en $(2/3, 13/3, 0, 0)$.



Algorithme du simplexe : un peu de formalisation

Notation et terminologie

On considère le programme donné sous forme standard par

$$\begin{array}{lc} \text{maximiser} & z = v + \sum_{j=1}^{n} c_j x_j \\ \text{sujet á} & \\ & \forall i \in \{1, \dots, m\}, \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \\ \text{avec} & \forall j \in N, \quad x_j \geq 0 \end{array}$$

L'ensemble N est l'ensemble $\{1, \dots, n\}$ des indices de variables. Notez que v est la valeur objectif de la solution de base ; ce n'est pas une solution admissible en général, elle peut avoir des entrées négatives.

Notation et terminologie

La forme *slack* du programme précédent s'écrit

$$
\begin{array}{lc}
\text{maximiser} & z = \nu + \sum_{j=1}^{n} c_j x_j \\
\text{sujet á} & \\
 & \forall i \in \{1, \dots m\}, \quad x_{i+n} = b_i - \sum_{j=1}^{n} a_{ij} x_j \\
\text{avec} & \forall j \in N \cup B, \quad x_j \geq 0
\end{array}
$$

L'esnsemble des variables indexées par $B = \{n+1, \dots, n+m\}$ (à gauche) sont dites de ***base***. Les variables indexées par N (à droite) sont dites ***hors base***. La ***solution de base*** d'un tel programme est obtenue en mettant toutes les variables hors bases à 0.

Meta-étapes du simplexe

L’hypothèse temporaire sous laquelle nous travaillons peut être phrasée comme suit:

Hypothèse (BF)

La solution de base du programme initiale sous-forme standard est admissible.

Sous cette hypothèse aucun prétraitement n’est nécessaire pour lancer les pivots du simplexe. On traite en un premier temps le cas satisfaisant ($\boldsymbol{BF}$), avant d’attaquer le cas général.

Méta-étapes du simplexe

Le simplexe est une série d'opérations pivots échangeant une variable de base avec une autre hors-base.

Méta-étapes du simplexe

Le simplexe est une série d'opérations pivots échangeant une variable de base avec une autre hors-base.En voici les étapes essentielles sous $(\boldsymbol{BF})$.

- Choisir une variable hors-base x_e qui augmente la valeur objectif.

Méta-étapes du simplexe

Le simplexe est une série d'opérations pivots échangeant une variable de base avec une autre hors-base.En voici les étapes essentielles sous ($\boldsymbol{BF}$).

- Choisir une variable hors-base x_e qui augmente la valeur objectif.
- Utiliser la contrainte la plus restrictiveℓ sur x_e pour exprimer x_e en fonction du reste.

Méta-étapes du simplexe

Le simplexe est une série d'opérations pivots échangeant une variable de base avec une autre hors-base.En voici les étapes essentielles sous $(\boldsymbol{BF})$.

- Choisir une variable hors-base x_e qui augmente la valeur objectif.
- Utiliser la contrainte la plus restrictiveℓ sur x_e pour exprimer x_e en fonction du reste.
- Remplacer x_e dans le reste des équations et la fonction objectif par l'expression précédente. La variable x_e devient de base et la variable x_ℓ hors-base.

Méta-étapes du simplexe

Le simplexe est une série d'opérations pivots échangeant une variable de base avec une autre hors-base.En voici les étapes essentielles sous ($\boldsymbol{BF}$).

- Choisir une variable hors-base x_e qui augmente la valeur objectif.
- Utiliser la contrainte la plus restrictiveℓ sur x_e pour exprimer x_e en fonction du reste.
- Remplacer x_e dans le reste des équations et la fonction objectif par l'expression précédente. La variable x_e devient de base et la variable x_ℓ hors-base.
- En mettant toutes les variables hors-base à 0 on obtient un tuple dont les n premières entrées forment une solution admissible de notre programme linéaire.

Méta-étapes du simplexe

Le simplexe est une série d'opérations pivots échangeant une variable de base avec une autre hors-base.En voici les étapes essentielles sous ($\boldsymbol{BF}$).

- Choisir une variable hors-base x_e qui augmente la valeur objectif.
- Utiliser la contrainte la plus restrictiveℓ sur x_e pour exprimer x_e en fonction du reste.
- Remplacer x_e dans le reste des équations et la fonction objectif par l'expression précédente. La variable x_e devient de base et la variable x_ℓ hors-base.
- En mettant toutes les variables hors-base à 0 on obtient un tuple dont les n premières entrées forment une solution admissible de notre programme linéaire.
- Réitérer les étapes précédentes jusqu'à ne plus pouvoir augmenter la valeur objectif.

KO sur la non bornitude[1]

Il n'y aucune garantie que l'algorithme précédent nous donne bien la valeur optimale du programme de départ. On sait, en général, que si la fonction objectif est bornée sur la région admissible, alors il y a une valeur optimale atteinte sur le bord de cette région. Il y a cependant des cas où un programme linéaire n'a pas une fonction objectif bornée sur le l'espace admissible.

Non bornitude

Trouver un programme linéaire qui n'est pas borné!

[1] Du verbe bornituder.

Méta-étapes du simplexe

Les étapes précédentes sont précisément ce qu'on a effectué lors de l'étude de notre premier exemple. Cette approche ne se limite cependant pas au de dimension 2.

Un cas de dimension 3

En suivant la démarche précédente, tenter de résoindre le programme suivant:

$$
\begin{array}{ll}
\text{maximiser} & 3x_1 + x_2 + 2x_3 \\
\text{sujet à} & \\
& x_1 + x_2 + 3x_3 \leq 30 \\
& 2x_1 + x_2 + 5x_3 \leq 24 \\
& 4x_1 + x_2 + x_3 \leq 36 \\
\text{avec} & x_1, x_2, x_3 \geq 0
\end{array}
$$

C’est tout pour aujourd’hui!