Optimisation Discrète

Section 1
Bienvenue

Vos formateurs

Professeur

- Friedrich Eisenbrand

Assistants

- Martin Niemeier (assistant principal): anglais, allemand, français
- Adrian Bock: anglais, allemand, français
- Laura Sanità: anglais, italien
- Rico Zenklusen: anglais, allemand, français

Lois du jeu

Notes

- Un examen écrit a la fin du semestre
- On peut obtenir des points de bonus (quelques exercices marqué avec (Δ) et $(*)$ sur les feuilles, vos solutions doivent être déposées à l'heure et seront corrigées par les assistants)
- La formule pour la note finale est

$$1/2 \cdot \lceil 10 \cdot p + 2 + b \rceil$$

  - p: pourcentage des points obtenus a l'examen
  - b: pourcentage des points bonus obtenus

Exercices, consultations

Exercices

- Toutes les 2 semaines: 1 feuille d'exercices
- Profitez bien de vos assistants, demandez des questions, soyez présents pendant les séances d'exercices

Heures de consultation F. Eisenbrand

Tous les lundis à midi

Section 2
Programmation linéaire

Objectives

- Définition d'un programme linéaire
- Un peut d'intuition géométrique
- Modéliser certains problèmes comme programmes linéaires
- Utiliser des logiciels
- Programmation linéaire dans la technologie moderne

Une diète saine et bon marché

But

Acheter de la nourriture à prix minimal tel que les besoins journaliers de certaines vitamines et d'énergie sont satisfaits. Il y a trois types de nourriture. Les carottes, le chou blanc et les céréales.

Vitamine A, B, énergie

- 100g de carottes contiennent: 3.5 mg de vitamine A, 6 mg de vitamine B, 50 kcal d'énergie
- 100g de chou blanc contiennent: 0.1 mg de vitamine A, 30 mg de vitamine B, 70 kcal d'énergie
- 100g de céréales contiennent: 0.02 mg de vitamine A, 0.04mg de vitamine B, 300 kcal d'énergie

Les prix pour 100g de nourriture

- 1 CHF carottes
- 0.5 CHF chou blanc
- 3 CHF céréales

Besoins journaliers

- vitamine A: 0.75 mg
- vitamine B: 0.5 mg
- énergie: 1500 kcal

Un programme linéaire

On introduit les variables x_1, x_2 et x_3, représentant le nombre d'unités de carottes, de choux blancs et de céréales, où une unité correspond à 100g.
Minimiser le coût d'un plat journalier:

$$\min \quad 1 \cdot x_1 + 0.5 \cdot x_2 + 3 \cdot x_3.$$

La contrainte d'au moins 0.75 mg de vitamine A s'écrit comme

$$3.5x_1 + 0.1x_2 + 0.02x_3 \geqslant 0.75.$$

Les variables x_1, x_2 et x_3 doivent être non-négatives.

Le programme linéaire

$$
\begin{array}{ll}
\min & 1 \cdot x_1 + 0.5 \cdot x_2 + 3 \cdot x_3 \\
\text{sous contraintes} & x_1 \geqslant 0 \\
& x_2 \geqslant 0 \\
& x_3 \geqslant 0 \\
& 3.5x_1 + 0.1x_2 + 0.02x_3 \geqslant 0.75 \\
& 6x_1 + 30x_2 + 0.04x_3 \geqslant 0.5 \\
& 50x_1 + 70x_2 + 300x_3 \geqslant 1500.
\end{array}
$$

Notation

Soient $A \in \mathbb{R}^{m \times n}$, $i \in \{1, \ldots, m\}$ et $j \in \{1, \ldots, n\}$

- a_i: $i^{\text{ème}}$ ligne de A
- a^j: $j^{\text{ème}}$ colonne de A
- $A(i, j)$: l'élément de A qui est dans la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne de A

Soient $v \in \mathbb{R}^m$ et $i \in \{1, \ldots, m\}$

- $v(i)$: $i^{\text{ème}}$ élément de v

Programme linéaire: Définition

Programme linéaire (PL)

Soit $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $I_{\geqslant}, I_{\leqslant}, I_{=} \subseteq \{1, \ldots, m\}$ et $J_{\geqslant}, J_{\leqslant} \subseteq \{1, \ldots, n\}$.
Un programme linéaire (PL) est décrit par

i) une fonction objective linéaire:

$$\max c^T x$$
$$\text{ou } \min c^T x$$

ii) des contraintes linéaires

$$a_i^T x \geqslant b(i), i \in I_{\geqslant}$$
$$a_j^T x \leqslant b(j), j \in I_{\leqslant}$$
$$a_k^T x = b(k), k \in I_{=}$$

iii) et des bornes sur les variables

$$x(j) \geqslant 0, j \in J_{\geqslant}$$
$$x(j) \leqslant 0, j \in J_{\leqslant}.$$

On peut récrire

- $\min c^T x$ comme $\max -c^T x$
- $a_i^T x \geqslant b(i), i \in I_{\geqslant}$ comme $-a_i^T x \leqslant -b(i), i \in I_{\geqslant}$
- $a_k^T x = b(k), k \in I_{=}$ comme $a_k^T x \leqslant b(k), -a_k^T x \leqslant -b(k), k \in I_{=}$
- $x(i) \geqslant 0$ comme $-e_i^T x \leqslant 0$, où e_i est le $i^{\text{ème}}$ vecteur unitaire
- $x(i) \leqslant 0$ comme $e_i^T x \leqslant 0$

Forme d'inégalité standard

Un programme linéaire peut toujours être écrit comme

$$\max\{c^T x \colon Ax \leqslant b, x \in \mathbb{R}^n\}$$

avec une matrice $A \in \mathbb{R}^{m \times n}$ et des vecteurs $b \in \mathbb{R}^m$ et $c \in \mathbb{R}^n$. Cette représentation est appelée la forme d'inégalité standard.

Admissible, borné, solution optimale

- Un point $x^* \in \mathbb{R}^n$ est appelé admissible, si x^* satisfait toutes les contraintes et les bornes sur les variables. S'ils existent des solutions admissibles pour un programme linéaire alors le programme lui-même est appelé admissible.
- Un programme linéaire est borné s'il existe une constante $M \in \mathbb{R}$ telle que pour tout $x^* \in \mathbb{R}^n$ qui est admissible, on a soit $c^T x^* \leqslant M$, si le programme linéaire est un problème de maximisation, ou soit $c^T x^* \geqslant M$, si le programme linéaire est un problème de minimisation.
- Une solution admissible x^* est une solution optimale si $c^T x^* \geqslant c^T y^*$ pour tout y^* admissible, si le programme linéaire est un problème de maximisation, ou $c^T x^* \leqslant c^T y^*$, si le programme linéaire est un problème de minimisation.

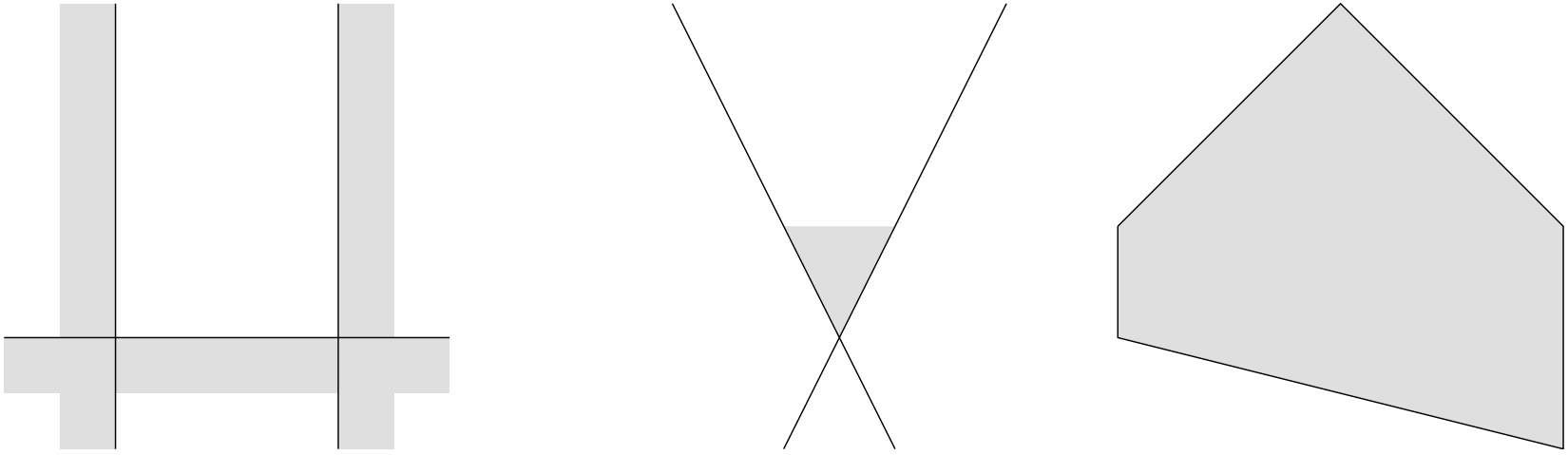


Figure: Ayant comme fonction objective de trouver le point le plus haut, nous avons de gauche à droite un programme linéaire inadmissible, un programme linéaire non-borné et un programme linéaire borné.

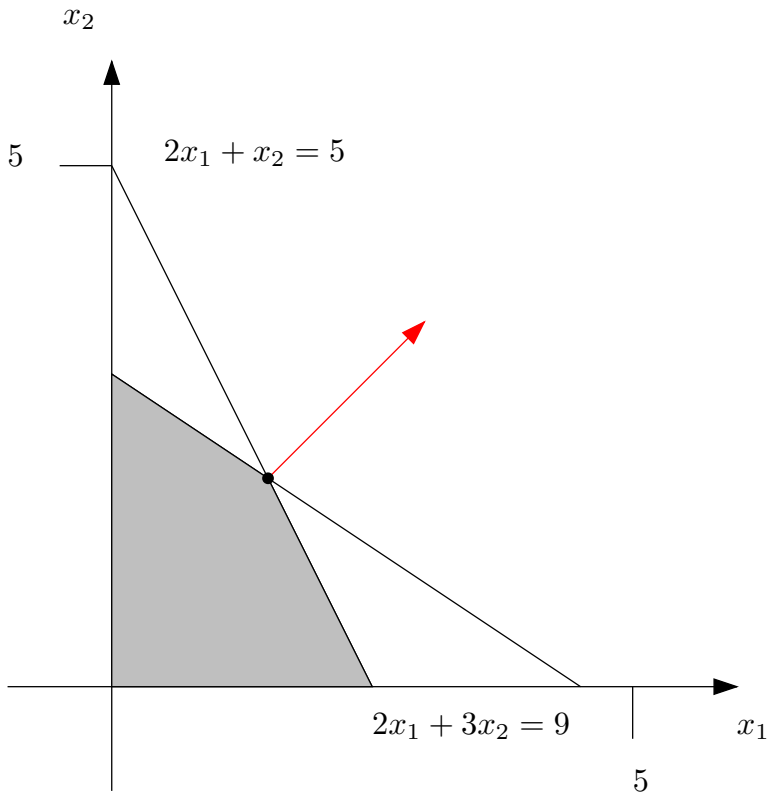
Nous allons montrer dans ce cours qu'un programme linéaire admissible et borné admet une solution optimale.

Programmes linéaires à deux variables

Les programmes linéaires à deux variables peuvent être résolus à l'aide graphique. Considérons par exemple le programme linéaire suivant:

$$
\begin{array}{lcl}
\max x_1 + x_2 & & \\
2x_1 + 3x_2 & \leqslant & 9 \\
2x_1 + x_2 & \leqslant & 5 \\
x_1, x_2 \geqslant 0. & &
\end{array}
$$

La figure ci-dessous indique les solutions admissibles (région grise). Le vecteur rouge représente la fonction objective $(1,1)$. Ce programme linéaire est admissible et borné. La solution optimale est l'intersection entre les deux droites $2x_1 + x_2 = 5$ et $2x_1 + 3x_2 = 9$. Cette intersection correspond à $x^* = (3/2, 2)$.



Ajuster une droite

Un problème bien connu en statistique:

On observe des points $(x_i, y_i) \in \mathbb{R}^2 \quad i = 1, \ldots, n$ et on s'intéresse à trouver une fonction linéaire $y = a \cdot x + b$ qui reflète l'échantillon

Une manière de faire ceci est de minimiser:

$$\sum_{i=1}^{n} (ax_i + b - y_i)^2, \tag{1}$$

où $a, b \in \mathbb{R}$ sont les paramètres de la droite qui est cherchée.

$(ax_i + b - y_i)^2$: Carré de la distance verticale du point (x_i, y_i) à la droite $y = ax + b$

Au lieu d'utiliser la méthode des moindres carrés on pourrait aussi minimiser la fonction suivante un peu plus robuste face à des valeurs déviantes:

$$\sum_{i=1}^{n} |ax_i + b - y_i|. \tag{2}$$

L'astuce:

Introduire une variable de plus qui modélise la valeur absolue de $ax_i + b - y_i$.

Le programme linéaire

$$\begin{array}{lll} \min \sum_{i=1}^{n} h_i & & \\ h_i & \geqslant & ax_i + b - y_i,\ i = 1, \ldots, n \\ h_i & \geqslant & -(ax_i + b - y_i),\ i = 1, \ldots, n \end{array} \tag{3}$$

Les variables sont h_i, $i = 1, \ldots, n$, a et b. Pour $a \in \mathbb{R}$ et $b \in \mathbb{R}$ fixés, les h_i optimaux seront $h_i = |ax_i + b - y_i|$ vu que la fonction objective minimise la somme des h_i. Si un des h_i était strictement plus grand que $|ax_i + b - y_i|$ alors la fonction objective pourrait être améliorée en diminuant ce h_i.

Solveurs et langages de modélisation

But

Trouver une droite ajustée comme décrit avant pour les points

$$(1,3),(2.8,3.3),(4,2),(5.5,2.1),(6,0.2),(7,1.3),(7.5,1),(8.5,0.8).$$

Il y a deux formats bien-connus pour des problèmes de programmation linéaire qui sont couramment utilisés par des solveurs de programmes linéaires. Ce sont le format lp et le format mps. Tous les deux ne sont pas faciles à lire. Pour faciliter la modélisation d'un programme linéaire, des langages de modélisation sont utilisés. Nous montrons l'utilisation du logiciel de modélisation à code source ouvert appelé zimpl [2].

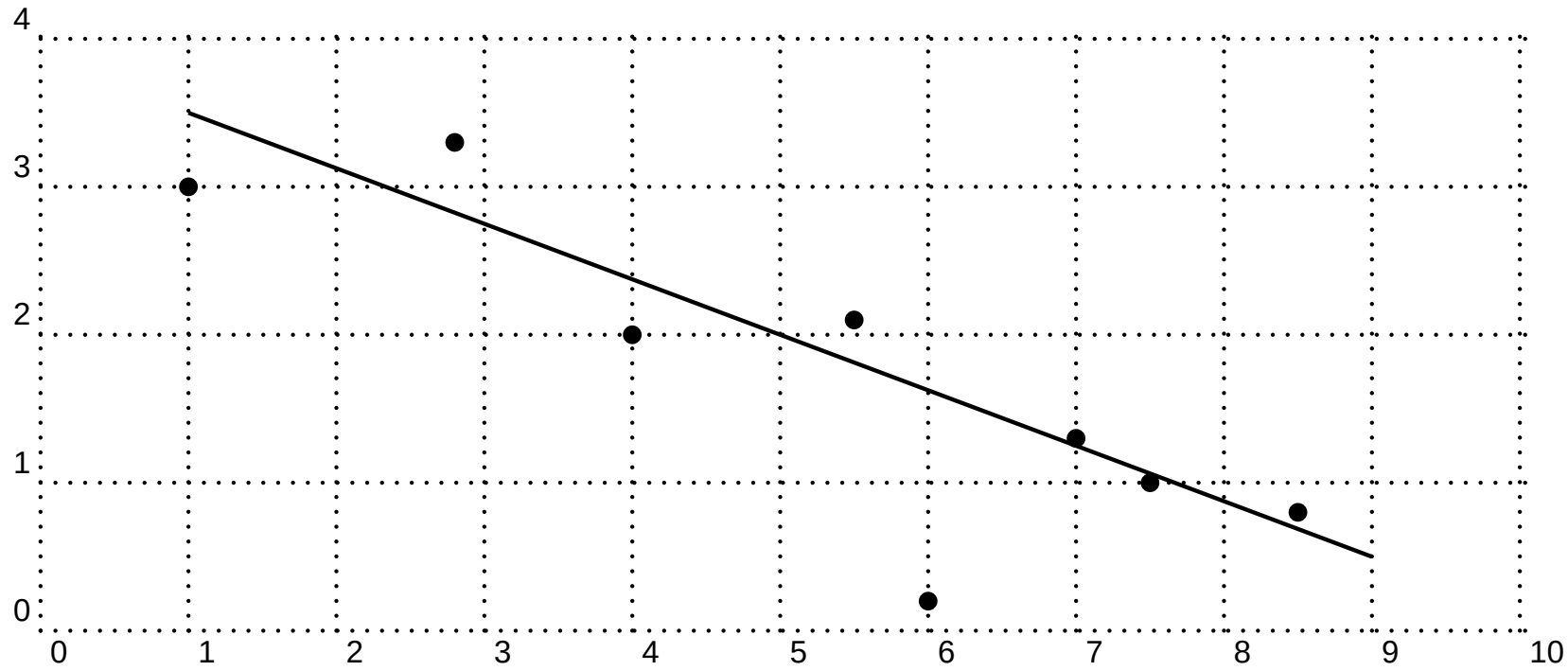
Langages de modélisation

```
set I := { 1 to 8};
param X[I] :=  <1> 1, <2> 2.8, <3> 4, <4> 5.5,
               <5> 6, <6> 7, <7> 7.5, <8> 8.5 ;
param Y[I] :=  <1> 3, <2> 3.3, <3> 2, <4> 2.1,
               <5> .2, <6> 1.3, <7> 1, <8> .8 ;
var h[I] >= -infinity <= infinity;
var a  >= -infinity <= infinity ;
var b  >= -infinity <= infinity ;

minimize cost: sum <i> in I: h[i];

subto c1: forall <i> in I:      h[i] >=   ( a * X[i] + b -Y[i]);
subto c2: forall <i> in I:      h[i] >= - ( a * X[i] + b -Y[i]);
```

Zimpl crée un programme linéaire qui est lisible par des solveurs linéaires de programmation comme QSopt or SoPlex.

Le résultat

Une droite ajustée optimale qui respecte la mesure de distance (2) est la droite $y = -0.293333 \cdot x + 3.293333$.



Programmation linéaire pour une durée de vie de DELO plus longue

Les diodes électroluminescentes organiques (DELOs)

- La technologie d'écran du future
- Les produits commerciaux sont de plus en plus munis avec de tels écrans
- Or, la technologie DELO la moins chère souffre d'une courte durée de vie. Quoi faire?

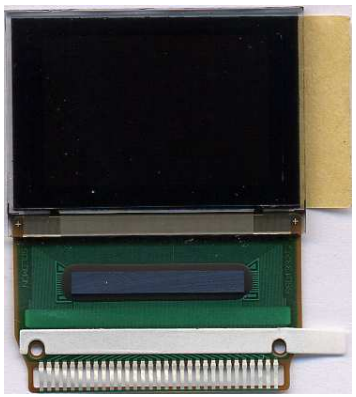


Figure: Échantillon d'un appareil commercial DELO avec driveur à puce intégrée

Un peu de technique

- DELO a une structure de matrice consistant de n lignes et de m colonnes
- L'image: Matrice non-négative de taille $n \times m$ dont les valeurs RGB sont représentées par $(r_{ij}) \in [0, \ldots, \varrho]^{n \times m}$
- Pour l'instant, l'interrupteur de la ligne i et de la colonne j est fermé, un courant électrique passe par la diode du pixel (i, j) et rend le pixel (i, j) éclairé
- À une séquence d'images suffisamment haute, par exemple 50 Hz, la perception de l'oeil est la valeur moyenne de la lumière émis par le pixel et on voit l'image
- Le schéma traditionnel d'adressage est ligne par ligne
- Avec ce schéma d'adressage, les pixels ne sont pas utilisé la plupart du temps et doivent donc briller avec une très haute intensité. Ceci rend les diodes tendus et est une cause majeure de la courte durée de vie des écrans

Sauver du temps / diminuer l'intensité

109	238	28	=	0	82	25	+	0	0	0	+	109	156	3
112	237	28		0	82	25		112	155	3		0	0	0
150	234	25		0	41	22		112	155	3		38	38	0
189	232	22		0	41	22		189	191	0		0	0	0
227	229	19		0	0	0		189	191	0		38	38	19

Figure: Un exemple de décomposition

Le programme linéaire

$$\min \sum_{i=1}^{n} u_i^{(1)} + \sum_{i=1}^{n-1} u_i^{(2)}$$

$$\text{s.c.} \quad f_{ij}^{(1)} + f_{i-1,j}^{(2)} + f_{ij}^{(2)} = r_{ij} \qquad \text{pour tout } i,j \qquad (4)$$

$$f_{ij}^{(\alpha)} \leqslant u_i^{(\alpha)} \qquad \text{pour tout } i,j,\alpha \qquad (5)$$

$$f_{ij}^{(\alpha)} \in \mathbb{R}_{\geqslant 0} \qquad \text{pour tout } i,j,\alpha$$

En décomposant les images de cette manière la durée de vie moyenne d'un écran DELO peut être augmentée de 100% environ, voir [1].

Objectives

- Définition d'un programme linéaire ✔
- Un peu d'intuition géométrique ✔
- Modéliser certains problèmes comme programme linéaire ✔
- Utiliser des logiciels ✔
- Programmation linéaire dans la technologie moderne ✔


F. Eisenbrand, A. Karrenbauer, and C. Xu.
Algorithms for longer oled lifetime.
In C. Demetrescu, editor, *6th International Workshop on Experimental Algorithms, WEA07*, volume 4525 of *Lecture Notes in Computer Science*, pages 338–351. Springer, 2007.
27


T. Koch.
Rapid Mathematical Programming.
PhD thesis, Technische Universität Berlin, 2004.
ZIB-Report 04-58.
21