



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Universität
Zürich^{UZH}

Mathematical Optimization, Autumn Semester 2017

Summary of the lectures in 2017

Vanessa Leite

Github (git/svn) repository page:

<https://github.com/ssinhaleite/eth-mathematical-optimization-summary>

Contact vrclite@gmail.com if you have any questions.

Sunday 1st October, 2017

Professor:
Prof. Dr. Robert Weismantel
D-MATH, ETH, UZH

Summary of the lectures about Mathematical Optimization - Prof. Dr. Robert Weismantel.
You can find a referece list in the end of each chapter.

Contents

Contents		i
1	Introduction	1
1.1	Complexity Theory - short revision	1
	Decision Problems	1
1.2	Optimization problems in finite dimensional space	2
1.3	Basic definitions	3
1.4	Classification of Optimization Problems	5
2	Linear Optimization and Extreme Points I	6
3	Glossary	7
4	TODO	7

Introduction

Complexity Theory - short revision

Definition: an algorithm is a finite list of instructions. The running time of an algorithm is the number of operations needed to complete it.

Simplifications

- We measure the “complexity” of an algorithm with respect to the length of the input, n . $T_A(n)$ = complexity of the algorithm = worst case for same input size = calculated/defined as the maximum over all inputs of length n .
- We are only interested in how $T_A(n)$ increases w.r.t. n .
 $T_A(n) = O(n)$: “linear”
 $T_A(n) = O(n^k)$: “polynomial”
 $T_A(n) = O(2^n)$: “exponential”
- We assume that every operation of the algorithm requires the same cost (“unit cost model”).

Note We need to assume that the number of bits of any stored member must be bounded by a (constant) multiple of the number of bits of the largest input value. Example: if $a \in \mathbb{Z}$ is part of the input, its input size is $O(\log|a|)$. An algorithm with running time of $O(a)$ is not polynomial-time! ($O(e^{\log a})$: “pseudo polynomial time” $\rightarrow O((\log m \times A_{ij}) \times m \times n)$).

Decision Problems

Let $\Sigma = \{0, 1\}$ and Σ^* be a finite binary sequences (sequences on Σ). We assume that all problems are encoded as elements of Σ^* .

Definition \mathcal{P} problems: For a set $S \subseteq \Sigma^*$, the decision problem defined by S is the problem whether a given $x \in \Sigma^*$ is in S or not.

It is a polynomial-time solvable ($S \in \mathcal{P}$) if there exists an algorithm \mathcal{A} s.t. $\forall x \in \Sigma^*, \mathcal{A}(x) = 1 \iff x \in S$.

If you can solve a decision problem, usually you can solve the optimization problem.

Definition \mathcal{NP} problems: The decision problem defined by $S \subseteq \Sigma^*$ is in \mathcal{NP} if there is a polynomial time algorithm $\mathcal{A} : \Sigma^* \times \Sigma^* \mapsto \{0, 1\}$ and a polynomial p , s.t.:

- $\forall x \in S, \exists y \in \Sigma^*$ s.t. $|y| \in O(p(|x|))$ and $\mathcal{A}(x, y) = 1$
- $\forall x \in S, \forall y \in \Sigma^*, \mathcal{A}(x, y) = 0$

Example of \mathcal{P} problems: Let $G = (V, E)$ be an undirected graph. “does G have a cycle?”; “does G have a perfect matching?”. Example of \mathcal{NP} problems: “does there exist a cycle covering all vertices

(hamiltonian cycle)?" . This problem is easy to check, given a cycle, if it covers all vertices, however, it is not that easy find the cycle.

NP-Hard vs NP-Complete A decision problem S is reducible to S' ($S \subseteq pS'$) \iff there is a function $f : \Sigma^* \mapsto \Sigma^*$ compatible in polynomial s.t. $x \in S \iff f(x) \in S'$.

Lemma Let S, S' be decision problems. If $S' \in \mathcal{P}$ (is polynomial-time solvable). Then, $S \subseteq pS' \Rightarrow S \in \mathcal{P}$.

Definition NP-Complete: A decision problem is called NP-complete if $S \in \mathcal{NP}$ and $\forall S' \in \mathcal{NP}, S' \subseteq pS$. If you have a solution for S , you could solve all other S problems in polynomial time.

Definition NP-Hard: A decision problem S is NP-Hard if $\forall S' \in \mathcal{NP}, S' \subseteq pS$.

Optimization problems in finite dimensional space

Definition: Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function. $dom(f) = \{x \in \mathbb{R}^n \mid |f(x)| < \infty\}$.

We need a function to optimize. An optimization problem is of the form:

$$\min/\max_{x \in \mathcal{F}} f(x)$$

Where $\mathcal{F} \subseteq \mathbb{R}^n$ is the feasible domain given implicitly or by a "membership oracle". The function $f : \mathbb{R}^n \leftarrow \mathbb{R}$ is presented implicitly (i.e., is a domain you can't describe) or by an "evaluation oracle", for instance, given by a query: $x \in \mathbb{R}^n$: true ($x \in \mathcal{F}$) or false ($x \notin \mathcal{F}$).

Implicitly means that you need evaluate by a given point; continuum spectrum of points can be scaled and produce a minimum more negative than any natural.

What does $\min_{x \in \mathcal{F}} f(x)$ mean for us? This problem can be assigned to one of three meanings:

- nothing: the problem is *infeasible*; it means that the domain is empty, i.e., $\mathcal{F} = \emptyset$, there is nothing to optimize.
- the problem is *unbounded*, i.e., there exists a sequence of points $x^1, x^2, \dots \in \mathcal{F}$ and $\forall n \in \mathbb{N}, f(x^1) < -n \Rightarrow \lim_{i \rightarrow \infty} f(x^i) = -\infty$. There exists a sequence of points that for each natural number, $f(x^i) < -n$, so we cannot bound the sequence.
- the problem has an *optimal solution*, i.e., $\exists x^* \in \mathcal{F}$ such that $f(x^*) \leq f(z) \forall z \in \mathcal{F}$.

Target of the course Derive conclusions(meanings from above 1.2) for optimization problem $\min_{x \in \mathcal{F}} f(x)$ and give ideally a proof that our conclusion is correct. It is necessary to proof (find a proof) to hold the conclusions. What to do when the se \mathcal{F} is empty? It is necessary a mathematical theory.

Basic definitions

organize the enumeration of figures in the text

- A *half space* is a set of the form $\{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}$ for $a \in \mathbb{R}^n, \alpha \in \mathbb{R}$.
A *half space rational* is a set of the form $\{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}$ for $a \in \mathbb{Q}^n, \alpha \in \mathbb{Q}$. In this case the boundary is a rational line.
- A *polyhedron* is the intersection of a **finite** number of halfspaces (Figure 1.3). The intersection of halfspaces gives a region of finite points. In this case, it is called *polytope*: a bounded polyhedron. We need to use geometry and algebraic forms. In algebraic form: $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\} = \{x \in \mathbb{R}^n \mid a_1^T x \leq b_1, \dots, a_n^T x \leq b_n\}$ where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. A is a matrix and $Ax \leq b$ represents each row of A .
Rational polyhedron: $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m$.
one half space is also a polyhedron.
all points inside a polyhedron are called feasible.
- A set \mathcal{Q} is *convex* (Figure 1.3) if $\forall x, y \in \mathcal{Q}$ and $\lambda \in (0, 1)$ then $\lambda x + (1 - \lambda)y \in \mathcal{Q}$.
- A function f is *linear* if $f(x) = c^T x$ for $c \in \mathbb{R}^n$.
- A function f is *convex* if $f : \mathbb{R}^n \rightarrow \mathbb{R} \Rightarrow \emptyset \neq \text{dom}(\mathcal{F})$ is a convex set and $f(\lambda x + (1 - \lambda)y) \leq \lambda(f(x)) + (1 - \lambda)(f(y)) \forall x, y \in \text{dom}(\mathcal{F})$ and $\forall \lambda \in (0, 1)$.
- A function f is *strictly convex* if $f(\lambda x + (1 - \lambda)y) < \lambda(f(x)) + (1 - \lambda)(f(y)) \forall x, y \in \text{dom}(\mathcal{F})$ and $\forall \lambda \in (0, 1)$.
- If f be continuously differentiable, then f is convex if, and only if, $f(y) \geq f(x) + \Delta f(x)^T(yx) \forall y, x \in \text{dom}(f)$. And strictly convex if, and only if, $f(y) > f(x) + \Delta f(x)^T(yx) \forall y, x \in \text{dom}(f)$ and $y \neq x$.
- If f is twice differentiable, then f is convex if, and only if, $\Delta^2 f(x) \geq 0, \forall x \in \text{dom}(f)$ (positive semidefinite).

Convex set Given vectors $(x^1, \dots, x^t \in \mathbb{R}^n)$ the convex hull of this vectors are defined as $\text{conv}(x^1, \dots, x^t) := \{x \in \mathbb{R}^n \mid \exists \lambda_1, \dots, \lambda_t \geq 0\}$ such that $\mathbf{1}^T \lambda = 1$ (or $\sum_{i=1}^t \lambda_i = 1$): $x = \sum_{i=1}^t \lambda_i x^i$.

Note $\text{conv}(x^1, \dots, x^t) = \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^t \lambda_i x^i, \lambda \geq 0, \sum \lambda = 1\}$ is a special case of a polyhedron (bounded polyhedron) and every bounded polyhedron is of this form.

Lemma

1. a polyhedron is a convex set
2. $\text{conv}(x^1, \dots, x^t)$ is a convex set

Figure 1: A polyhedron/polytope.

$$Ax \leq b$$

A : $m \times n$ matrix

b : $m \times 1$ vector

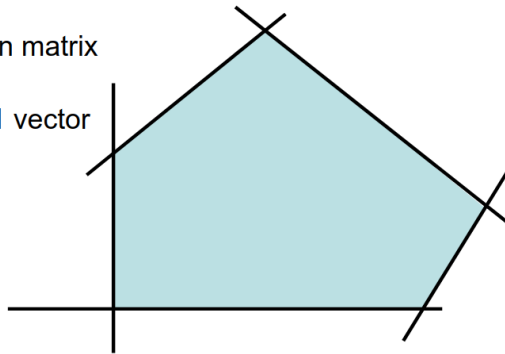
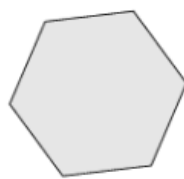
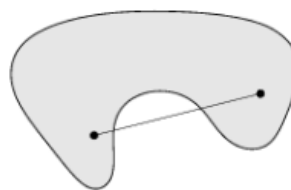


Figure 2: Convex and non-convex set.



convex



not convex

Proof of 1 Let $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Let $x, y \in \mathcal{P}, \lambda \in (0, 1)$. Show $\lambda x + (1 - \lambda)y \in \mathcal{P}$.

$$A[\lambda x + (1 - \lambda)y] = \lambda Ax + (1 - \lambda)Ay$$

$$\lambda Ax \leq \lambda b \text{ and } (1 - \lambda)Ay \leq (1 - \lambda)b$$

$$\lambda Ax + (1 - \lambda)Ay \leq \lambda b + (1 - \lambda)b = b$$

$$\Rightarrow \lambda x + (1 - \lambda)y \in \mathcal{P}.$$

Proof of 1 Let $x^1, \dots, x^t \in \mathbb{R}^n$. Let $x, y \in \text{conv}(x^1, \dots, x^t)$ and $0 < \lambda < 1$. Show $\lambda x + (1 - \lambda)y \in \text{conv}(x^1, \dots, x^t)$.

x lies in $\text{conv}(x^1, \dots, x^t)$. What does that mean? That implies $x = \sum_{i=1}^t \mu_i x^i, \mu_i \geq 0 \forall i$ and $y = \sum_{i=1}^t \sigma_i x^i, \sigma_i \geq 0 \forall i$

$$\begin{aligned}
\lambda x + (1 - \lambda)y &= \sum_{i=1}^t \lambda \mu_i x^i + \sum_{i=1}^t (1 - \lambda) \sigma_i x^i \\
&= \sum_{i=1}^t (\lambda \mu_i + (1 - \lambda) \sigma_i) x^i \\
\tau_i &= (\lambda \mu_i + (1 - \lambda) \sigma_i), \tau_i \geq 0, \forall i \\
&= \sum_{i=1}^t \tau_i = \sum_{i=1}^t (\lambda \mu_i + (1 - \lambda) \sigma_i) \\
&= \lambda \sum_{i=1}^t \mu_i + (1 - \lambda) \sum_{i=1}^t \sigma_i \\
&\quad \sum_{i=1}^t \mu_i = 1 \text{ and } \sum_{i=1}^t \sigma_i = 1 \\
&= \lambda + (1 - \lambda) = 1
\end{aligned}$$

Classification of Optimization Problems

Definition of optimization problem: For $f : \mathbb{R}^n \mapsto \mathbb{R}$ (any function), solve $\min_{x \in \mathcal{F}} f(x)$ (objective value) or $\operatorname{argmin}_{x \in \mathcal{F}} f(x)$ (value of x that minimize $f(x)$).

Note $\max_{x \in \mathcal{F}} f(x) = \min_{x \in \mathcal{F}} -f(x)$

- *linear optimization problem*: given $A \in \mathcal{Q}^{m \times n}, b \in \mathcal{Q}^m, c \in \mathcal{Q}^n$. $\min_{x \in \mathcal{P}} c^T x$ where \mathcal{P} is a polyhedron defined by $\{x \in \mathbb{R}^n \mid x_i \geq 0 \forall i, Ax = b\}$ or $\max_{x \in \mathcal{P}} c^T x$ where $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$

$c^T x$ is the objective (linear) function.

$Ax \leq b$ is the constraint/feasible region.

Can be solved quickly and in polynomial time.

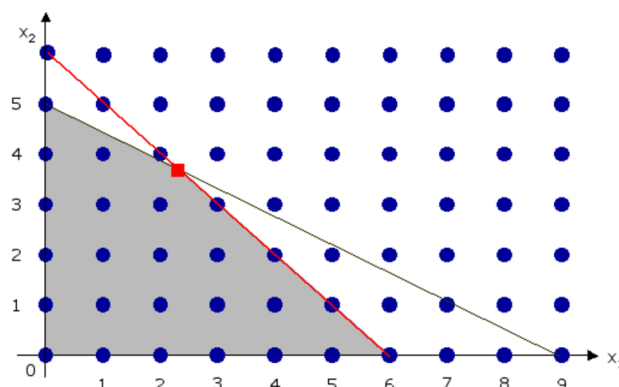
integer linear optimization problem (Figure 1.4). $\max/\min\{c^T x \mid Ax \leq b, x \in \mathbb{Z}^n\}$. Solve a integer optimization problem can lead to a better result than just round linear solutions.

mixed integer linear optimization problem $\max/\min\{c^T x \mid Ax \leq b, x \in \mathbb{Z}^{n-d} \times \mathbb{R}^d\}$.

- *convex optimization problem*: \mathcal{F} is a convex set, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function and $\mathcal{F} \subseteq \operatorname{dom}(f)$
 $f(x)$ convex and \mathcal{F} convex set. $\forall x, y \in \mathbb{R}^n, \lambda \in [0, 1], f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$.
 $\forall x, y \in \mathcal{F}, \lambda x + (1 - \lambda)y \in \mathcal{F}$.

Example of implicitly: knapsack problem: collection are all feasible solutions of knapsack problem.

Figure 3: ILP



- *combinatorial optimization problem*: given finite groundset \mathcal{E} and a collection \mathcal{I} of subsets of \mathcal{E} (typically, implicitly given: you cannot write it down). For $c : \mathcal{E} \mapsto \mathcal{R}$ find a member in the collection ($I \in \mathcal{I}$) such that $\sum_{i \in I} c_i$ is minimal/maximal. Example: $G = (V, E), E \subseteq V \times V$
 $S \subseteq V$ stable implies that for all pairs in $S (\forall i, j \in S)$, the corresponding edge i, j is not present ($i \neq j, (i, j) \notin E$). Groundset: V
 $\mathcal{I} \subseteq 2^V$
 $I \in \mathcal{I} \iff I$ is stable in G (implicitly given)
 $c : V \mapsto \mathcal{R}$; suppose $c(v) = 1, \forall v \in V$
 find a maximal (w.r.t. cardinality) stable set in G
- *general integer optimization problem*:
 $\max_{x \in \mathcal{P} \cap \mathbb{Z}^n} c^T x, \mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ or $\min_{x \in \mathcal{P} \cap \mathbb{Z}^n} c^T x, \mathcal{P} = \{x \in \mathbb{R}^n \mid x_i \geq 0 \forall i, Ax = b\}$

Linear Optimization and Extreme Points I

Geometry can be misleading. We try to get some intuitions from geometry but what we want to do is came up with the algebra.

Consider $\max_{s.t. x \in \mathcal{P}} c^T x, \mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}, A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n$

From linear algebra we know:

$$A \in \mathbb{Q}^{m \times n}$$

$n = \dim(\text{Ker}(A)) + \dim(\text{Im}(A))$, where $\text{Ker}(A)$ is the kernel space and $\text{Im}(A)$ is the image space.

$$\text{Ker}(A) = \{x \in \mathbb{R}^n \mid Ax = 0\}$$

$$\text{Im}(A) = \{x \in \mathbb{R}^n \mid x^T = y^T A, \text{ for some } y \in \mathbb{R}^m\}$$

$\text{Ker}(A) \perp \text{Im}(A)$ (kernel space is orthogonal/perpendicular to image space).

Glossary

Term	Brief explanation of the term
A very important term	The much longer explanations breaking over several lines here and more text to see the line breaking again and again and again and again.

TODO

This is the chapter on what still has to be improved in the summary. Please update this list by writing the todos directly into the summary where necessary by writing:

```
\todo{Something that still has to be done}
```

For a missing figure, please write:

```
\missingfigure{A missing figure}
```

Everyone that picks up the work on this summary will thank you for keeping this up-to-date properly.

Todo list

organize the enumeration of figures in the text 3

Example of implicitly: knapsack problem: collection are all feasible solutions of knapsack problem. 5