1 Outline

In this lecture, we study

- Convex optimization applications (LASSO, Facility location),
- Classes of convex optimization problems (LP, QP, SDP),
- Convex optimization hierarchy.

2 Convex optimization applications II

2.1 LASSO (least absolute shrinkage and selection operator)

Based on n data points $(x_1, y_1), \ldots, (x_n, y_n)$, we want to find a linear rule

$$y = \beta^\top x$$

that best represents the relationship between x and y. The goal is to find β minimizing the "mean squared error", given by

$$\min_\beta \quad \frac{1}{n}\sum_{i=1}^n (y_i - \beta^\top x_i)^2 \quad = \quad \min_\beta \quad \frac{1}{n}\|y - X\beta\|_2^2$$

where the rows of X are $x_1^\top, \ldots, x_n^\top$.

However, there are issues such as highly collinear covariates and overfitting. Motivated by this, LASSO is a regression method that achieves variable selection and regularization. The LASSO problem is to solve

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n}\|y - X\beta\|_2^2 \\ \text{subject to} \quad & \|\beta\|_1 \le t \end{aligned}$$

where t is a parameter determining the degree of regularization. Basically, the problem induces sparsity in β. The problem is often transformed into the *Lagrangian* form, given as follows.

$$\min_\beta \quad \frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

where λ is set to control the degree of regularization.

2.2 Facility location

Given the locations of n households $x_1, \ldots, x_n \in \mathbb{R}^d$, we wish to build a hospital covering the households. A desired location would minimize the longest distance to a household. The problem can be formulated as

$$\min_x \quad \max_{i=1,\ldots,n} \|x - x_i\|.$$

3 Convex optimization hierarchy

On top of the applications we studied, there are many interesting "classes" of convex optimization problems. In this lecture, we consider them in this section, and specifically, we discuss

- Linear programming (LP),
- Quadratic programming (QP),
- Semidefinite programming (SDP),
- Conic programming,
- Second-order cone programming (SOCP).

In fact, these problems are closely related, and in fact, the problem classes form a hierarchy described in Figure 5.1

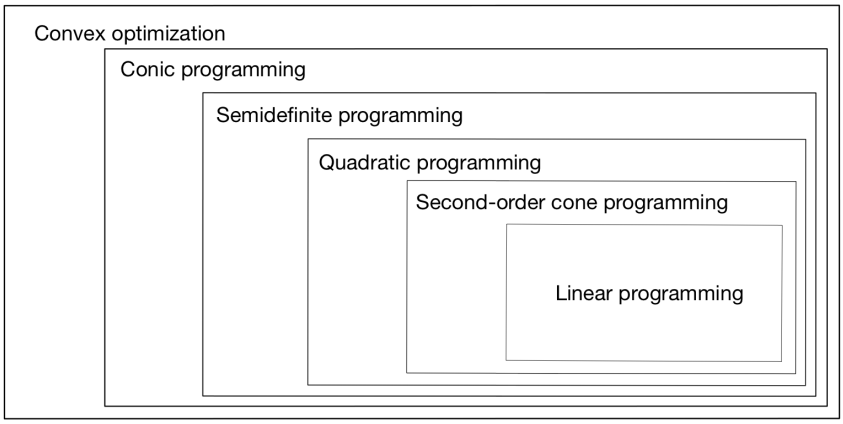

Figure 5.1: Hierarchy of classes of convex optimization problems

4 Linear programming

A linear program (LP) is an optimization problem of the following form.

$$\begin{aligned} \text{minimize} \quad & c^\top x \\ \text{subject to} \quad & Ax \geq b \end{aligned} \tag{P}$$

where $c^\top x$ is the linear objective function and $Ax \geq b$ is the system of linear constraints $a_1^\top x \geq b_1, \ldots, a_n^\top x \geq b_n$, i.e., $a_1^\top, \ldots, a_n^\top$ are the rows of A and $b = (b_1, \ldots, b_n)^\top$. Note that the feasible region $P = \{x \in \mathbb{R}^d : Ax \geq b\}$ is a polyhedron, the intersection of half-spaces $\{x \in \mathbb{R}^d : a_i^\top x \geq b_i\}$ for $i = 1, \ldots, n$. Hence, the linear program is the problem of finding a point in a polyhedron that minimizes a linear function. Figure 5.2 describes a linear program with 2 variables.

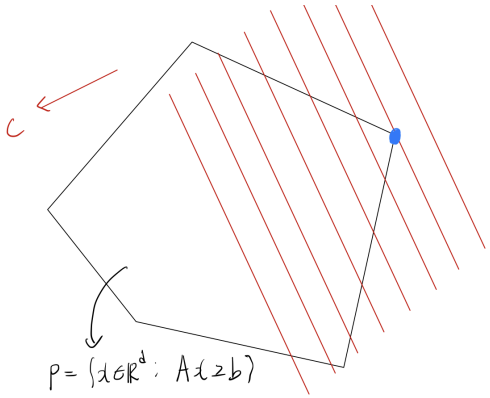

Figure 5.2: Geometric picture of a linear program

There are several equivalent forms of (P). First, we can transform (P) to a linear program all of whose variables are nonnegative.

$$\begin{array}{rl} \text{minimize} & c^\top x \\ \text{subject to} & Ax \geq b \\ & x \geq 0. \end{array} \tag{P'}$$

How? For each variable x_i, we separate its positive part and negative part, by setting $x_i = x_i^+ - x_i^-$ with $x_i^+, x_i^- \geq 0$. Then (P) is equivalent to

$$\begin{array}{rl} \text{minimize} & c^\top x^+ - c^\top x^- \\ \text{subject to} & Ax^+ - Ax^- \geq b \\ & x^+, x^- \geq 0. \end{array} \quad = \quad \begin{array}{rl} \text{minimize} & \begin{bmatrix} c \\ -c \end{bmatrix}^\top \begin{bmatrix} x^+ \\ x^- \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} A \\ -A \end{bmatrix} \begin{bmatrix} x^+ \\ x^- \end{bmatrix} \geq b \\ & \begin{bmatrix} x^+ \\ x^- \end{bmatrix} \geq 0. \end{array}$$

Second, (P) can be further transformed to a linear program all of whose constraints are equlities as the following.

$$\begin{array}{rl} \text{minimize} & c^\top x \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array} \tag{P''}$$

This is often called the *standard form*. We obtain the form by adding *slack variables* to (P') as follows.

$$\begin{array}{rl} \text{minimize} & c^\top x \\ \text{subject to} & Ax - s = b \\ & x, s \geq 0. \end{array}$$

$$\begin{array}{lll} \text{minimize} & c^\top x & \\ \text{subject to} & Ax - s = b & = \\ & x, s \geq 0. & \end{array} \qquad \begin{array}{ll} & \text{minimize} \quad \begin{bmatrix} c \\ 0 \end{bmatrix}^\top \begin{bmatrix} x \\ s \end{bmatrix} \\ & \text{subject to} \quad \begin{bmatrix} A \\ -I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = b \\ & \qquad\qquad \begin{bmatrix} x \\ s \end{bmatrix} \geq 0. \end{array}$$

4.1 LP duality

The dual linear program of (P) is given by

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & A^\top y = c, \\ & y \geq 0. \end{array} \tag{D}$$

The dual linear program of the standard form, given in (P'') is

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & A^\top y \leq c, \end{array} \tag{D''}$$

The *weak LP duality* states that $\text{OPT}(P) \geq \text{OPT}(D)$, where $\text{OPT}(P)$ and $\text{OPT}(D)$ denote the optimal values of (P) and (D), respectively. The *strong LP duality* states that if (P) is feasible and bounded, then (D) is feasible and bounded and $\text{OPT}(P) = \text{OPT}(D)$.

4.2 Example: facility location

Recall that the facility location problem can be modeled as follows.

$$\begin{array}{ll} \text{minimize} & \max\limits_{i=1,\ldots,n} \|x - x^i\|_1 \\ \text{subject to} & x \in \mathbb{R}^d \end{array}$$

where $x^1, \ldots, x^n \in \mathbb{R}^d$ are the locations of households and we use the ℓ_1 norm for the norm $\|\cdot\|$. In fact, the problem is equivalent to a linear program. Why? We first introduce an auxiliary variable to replace the objective. Then the problem is equivalent to

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & t \geq \max\limits_{i=1,\ldots,n} \|x - x^i\|_1. \end{array}$$

Moreover, $t \geq \max_{i=1,\ldots,n} \|x - x^i\|_1$ is equvialent to imposing $t \geq \|x - x^i\|_1$ for $i = 1, \ldots, n$. The next step is to replace $t \geq \|x - x^i\|_1$ by a set of linear inequalities. Note that

$$\|x - x^i\|_1 = \sum_{j=1}^d |x_j - x_j^i|.$$

By introducing an auxiliary variable s_{ij} for each absolute value term $|x_j - x_j^i|$, we replace $t \geq \sum_{j=1}^d |x_j - x_j^i|$ by $t \geq \sum_{j=1}^d s_{ij}$ and $s_{ij} \geq |x_j - x_j^i|$. Moreover, $s_{ij} \geq |x_j - x_j^i|$ is equivalent to

$s_{ij} \geq x_j - x_j^i \geq -s_{ij}$. Therefore, we obtain

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & t \geq \sum_{j=1}^{d} s_{ij} && \text{for } i = 1, \ldots, n, \\ & s_{ij} \geq x_j - x_j^i \geq -s_{ij} && \text{for } i = 1, \ldots, n,\ j = 1, \ldots, d \end{aligned}$$

which is a linear program.

4.3 History

Linear programming was first devised by Kantorovich in 1939 [Kan39]. Then LP was used to model problems in military operations research duing World War II, and during that time, Dantzig was part of Project SCOOP (Scientific Computation of Optimum Programs) arranged for Pentagon [CET16]. Dantzig developed the famous "Simplex method" for solving linear programs[1]. The following quote is from his note on the method written in 1985 [Dan85].

> Origins of the Simplex Method, Summer 1947
>
> The first idea that would occur to anyone as a technique for solving a linear program, aside from the obvious one of moving through the interior of the convex set, is that of moving from one vertex to the next along edges of the polyhedral set. I discarded this idea immediately as impractical in higher dimensional spaces. It seemed intuitively obvious that there would be far too many vertices and edges to wander over in the general case for such a method to be efficient.
>
> **When Hurwicz came to visit me at the Pentagon in the summer of 1947, I told him how I had discarded this vertex-edge approach as intuitively inefficient for solving LP. I suggested Instead that we study the problem in the geometry of columns rather than the usual one of the rows – column geometry incidently was the one I had used in my Ph.D. thesis on the Neyman-Pearson Lemma. We dubbed the new method "climbing the bean pole." It looked to me efficient.**
>
> I felt sufficiently confident in this special case of what later became known as the simplex method that I proceeded to modify it so that it would work for linear programs without a convexifying row. I also developed a variant for getting a starting feasible solution called Phase I. It was then that I discovered that the method was really the previously discarded vertex-edge procedure in disguise (except for an added criterion for selecting the edge on which to move). **Apparently, in one geometry the simplex method looks efficient while in another it appeared to be very inefficient! Thus the simplex method was born in August 1947.**

The simplex method works really well in practice! However, Klee and Minty found a pathological instance for the simplex method, requiring exponential time to solve if the method is used [KM72]. Later, Khachiyan announced in 1979 that he proved that the Ellipsoid method, proposed by Naum Z. Shor [Sho77] (also Yudin and Nemirovski [YN76]), solves linear programming in (weakly) polynomial time! [Kha79], and the full proof was published in 1980 [Kha80]. Although this result is indeed

[1] I was not able to find a specific document announcing the result in 1947.

a breakthrough in theory, it is a general perception that the method is not as practical. Later, Karmarkar proposed an interior-point algorithm, which is much faster than the ellipsoid method and is proved to run in polynomial time [Kar84]. There have been far greater improvements in linear programming, both in practice and theory, and it is still one of the central research topics in optimization.

The following is a list of recent progress on fast algorithms for linear programming.

- (Khachiyan, 1980 [Kha80]) $O(n^6)$ time ellipsoid method.
- (Vaidya, FOCS1989 [Vai89]) $O(n^{2.5})$ time implementation of Karmarkar's method.
- (Cohen, Lee and Song, STOC2019 [CLS19]) $O^*(n^\omega + n^{2.5-\alpha/2} + n^{2+1/6})$ time.
- (Jiang, Song, Weinstein, and Zhang, STOC2021 [JSWZ21]) $O^*(n^\omega + n^{2.5-\alpha/2} + n^{2+1/18})$ time.

Here, n is the number of variables, O^* hides $n^{o(1)}$ factors, ω is the exponent of matrix multiplication, and α is the dual exponent of matrix multiplication. Currently, there exist algorithms that achieve $\omega \sim 2.38$ and $\alpha \sim 0.31$.

5 Quadratic programming

A quadratic program (QP) is an optimization problem of the following form.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^\top Q x + p^\top x \\ \text{subject to} \quad & Ax \geq b \end{aligned} \tag{QP}$$

The quadratic program is convex only if Q is positive semidefinite.

5.1 Example: portfolio optimization

We studied the following formulation of portfolio optimization.

$$\begin{aligned} \text{maximize} \quad & \mu^\top x - \gamma x^\top \Sigma x \\ \text{subject to} \quad & 1^\top x = 1, \\ & x \in \mathbb{R}^n_+ \end{aligned}$$

where $\gamma > 0$ and Σ is a covariance matrix that is positive semidefinite. Note that

$$\max\{f(x) : \ x \in C\} = -\min\{-f(x) : \ x \in C\}.$$

Therefore, the formulation is equivalent to

$$\begin{aligned} \text{minimize} \quad & \gamma x^\top \Sigma x - \mu^\top x \\ \text{subject to} \quad & 1^\top x = 1 \\ & x \geq 0 \end{aligned}$$

which is clearly a quadratic program.

5.2 Example: support vector machine

The next example is the formulation of support vector machine.

$$\min_{w,b} \quad \lambda\|w\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} \max\{0,\ 1 - y_i(w^\top x_i - b)\}.$$

Here, $\|w\|_2^2 = w^\top w = w^\top I w$ where I is the identity matrix, and therefore, $\|w\|_2^2$ is a quadratic term. Moreover, the max terms in the objective can be replaced by adding some auxiliary variables. Note that the formulation is equivalent to

$$\begin{aligned} \text{minimize} \quad & \lambda w^\top w + \frac{1}{n}\sum_{i=1}^{n} t_i \\ \text{subject to} \quad & t_i \geq \max\{0,\ 1 - y_i(w^\top x_i - b)\} \text{ for } i = 1, \ldots, n. \end{aligned}$$

Next, we can rewrite the constraints as linear constraints as the following.

$$\begin{aligned} \text{minimize} \quad & \lambda w^\top w + \frac{1}{n}\sum_{i=1}^{n} t_i \\ \text{subject to} \quad & t_i \geq 1 - y_i(w^\top x_i - b) \quad \text{for } i = 1, \ldots, n, \\ & t_i \geq 0 \qquad\qquad\qquad\quad \text{for } i = 1, \ldots, n. \end{aligned}$$

Therefore, it is a quadratic program as well.

5.3 Example: LASSO

Recall that LASSO can be formulated as

$$\min_{\beta} \quad \frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1.$$

Note that

$$\|y - X\beta\|_2^2 = (y - X\beta)^\top (y - X\beta) = \beta^\top X^\top X\beta - 2y^\top X\beta + y^\top y$$

Here, $X^\top X$ is positive semidefinite, and $y^\top y$ is a constant term which can be ignored from the objective. Moreover, we can replace the $\|\beta\|_1$ term by an auxiliary variable and a set of linear constraints. To be specific, the problem is equivalent to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n}\beta^\top X^\top X\beta - \frac{2}{n}y^\top X\beta + \lambda t \\ \text{subject to} \quad & t \geq \sum_{i=1}^{d} s_i, \\ & s_i \geq \beta \geq -s_i \text{ for } i = 1, \ldots, d. \end{aligned}$$

Hence, LASSO can be reformulated as a quadratic program.

6 Semidefinite programming

6.1 Motivation: max-cut

Semidefinite programming provides useful tools for solving difficult combinatorial optimization problems. For example, we consider the "max-cut problem" defined as follows. Given a graph

$G = (V, E)$, find a partition the vertex set V so that the number of edges crossing the partition is maximized. Here, a partition (V_1, V_2) of V consists of two sets V_1, V_2 satisfying $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$, and the set of edges crossing the partition is basically $\{uv \in E : u \in V_1, v \in V_2\}$. The problem can be formulated by the following (discrete) optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{ij \in E} \frac{1 - x_i x_j}{2} \\ \text{subject to} \quad & x_i \in \{-1, 1\} \text{ for } i \in V. \end{aligned}$$

As long as $x_i \in \mathbb{R}$, $x_i \in \{-1, 1\}$ is equivalent to $x_i^2 = 1$. Hence, the formulation is equivalent to

$$\begin{aligned} \text{maximize} \quad & \sum_{ij \in E} \frac{1 - x_i x_j}{2} \\ \text{subject to} \quad & x_i^2 = 1 \text{ for } i \in V. \end{aligned}$$

Let $d = |V|$. Then we consider a $d \times d$ matrix X whose entry at ith row and jth column, X_{ij}, is $x_i x_j$. Then we have that $X = xx^\top$, which is the outer product of vector x and itself. In fact, X is of the form $X = xx^\top$ if and only if X is positive semidefinite and the rank of X is precisey 1. What this implies is that, the max-cut formulation can be rewritten as

$$\begin{aligned} \text{maximize} \quad & \sum_{ij \in E} \frac{1 - X_{ij}}{2} \\ \text{subject to} \quad & X_{ii} = 1 \text{ for } i \in V, \\ & X \succeq 0, \\ & \operatorname{rank}(X) = 1. \end{aligned}$$

Here, the constrsint $\operatorname{rank}(X) = 1$ is nonconvex, after taking which we obtain a convex optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{ij \in E} \frac{1 - X_{ij}}{2} \\ \text{subject to} \quad & X_{ii} = 1 \text{ for } i \in V, \\ & X \succeq 0. \end{aligned}$$

This is often called the *semidefinite programming (SDP) relaxation* of max-cut.

6.2 General form

More generally, a *semidefinite program* is an optimization problem of the following form. Let C and $A_1, \ldots, A_n$ be $d \times d$ matrices, and we have

$$\begin{aligned} \text{minimize} \quad & \operatorname{tr}(C^\top X) \\ \text{subject to} \quad & \operatorname{tr}(A_\ell^\top X) = b_\ell \text{ for } \ell = 1, \ldots, n \\ & X \succeq 0 \end{aligned} \tag{SDP}$$

where

$$\operatorname{tr}(C^\top X) = \sum_{i=1}^d \sum_{j=1}^d C_{ij} X_{ij} \quad \text{and} \quad \operatorname{tr}(A_\ell^\top X) = \sum_{i=1}^d \sum_{j=1}^d (A_\ell)_{ij} X_{ij}.$$

The *dual* semidefinite program of (SDP) is

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell=1}^{m} b_\ell y_i \\ \text{subject to} \quad & \sum_{\ell=1}^{m} y_\ell A_\ell \preceq C \end{aligned} \tag{dual-SDP}$$

where $\sum_{\ell=1}^{m} y_\ell A_\ell \preceq C$ means $C - \sum_{\ell=1}^{m} y_\ell A_\ell$ is positive semidefinite. If an optimization is in either form, we say that it is a semidefinite program.

We will study more about duality later in this course. We have dicussed LP duality, and in particular, we know how to derive the dual of a linear program. The notion of duality extends to more general classes of convex optimization problems, such as semidefinite programming. We will learn how to derive the dual of a given optimization problem, and we will discuss the notion of weak and strong duality.

6.3 Example: quadratic programming

(QP) can be rewritten as

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & Ax \geq b, \\ & x^\top Q x + 2p^\top x \leq 2t. \end{aligned}$$

In fact, this can be expressed as an instance of (dual-SDP) by rewriting $Ax \geq b$ and $x^\top Q x + 2p^\top x \leq 2t$ using some positive semidefinite matrices.

Note that $Ax - b$ is a vector and $Ax \geq b$ means that the entries of $Ax - b$ are nonnegative. $\mathrm{Diag}(Ax - b)$ is the diagonal matrix whose diagonal entries are the components of $Ax - b$. In fact, $Ax - b \geq 0$ holds if and only if

$$\mathrm{Diag}(Ax - b) \succeq 0$$

which means that $\mathrm{Diag}(Ax - b)$ is positive semidefinite.

Next we consider $x^\top Q x + 2p^\top x \leq 2t$ where Q is positive semidefinite.

Lemma 5.1. *For any positive semidefinite matrix Q, there exists a matrix P such that $Q = P^\top P$.*

Proof. By the eigen decomposition theorem from Lecture 2, Q can be written as $Q = U \Lambda U^\top$ where U is an orthnormal matrix and Λ is a diagonal matrix whose diagonal entries consist of the eigenvalues of Q. Since Q is positive semidefinite, all its eigenvalues are nonnegative, and therefore, all diagonal entries of Λ are nonnegative. Then $\Lambda^{1/2}$ can be properly defined by taking the square root of each diagonal entry of Λ. Then $\Lambda = (\Lambda^{1/2})^\top \Lambda^{1/2}$ as $\Lambda^{1/2}$ is symmetric as well. Then

$$Q = U \Lambda U^\top = U (\Lambda^{1/2})^\top \Lambda^{1/2} U^\top = (\Lambda^{1/2} U^\top)^\top (\Lambda^{1/2} U^\top).$$

Taking $P = \Lambda^{1/2} U^\top$, we have $Q = P^\top P$. □

By Lemma 5.1, $x^\top Q x + 2p^\top x \leq 2t$ is equivalent to

$$x^\top P^\top P x + 2p^\top x \leq 2t$$

for some matrix P. We also need the following result.

Lemma 5.2. *Let $y \in \mathbb{R}^d$. Then $y^\top y \le s$ is equivalent to*

$$\begin{pmatrix} s & y^\top \\ y & I \end{pmatrix} \succeq 0$$

where I is the $d \times d$ identity matrix.

Proof. ($\Leftarrow$) Note that

$$(1, -y^\top) \begin{pmatrix} s & y^\top \\ y & I \end{pmatrix} \begin{pmatrix} 1 \\ -y \end{pmatrix} = s - y^\top y \ge 0.$$

($\Rightarrow$) Let $u \in \mathbb{R}$ and $v \in \mathbb{R}^d$. Then

$$\begin{aligned} (u, v^\top) \begin{pmatrix} s & y^\top \\ y & I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} &= u^2 s + 2uy^\top v + v^\top v \\ &\ge u^2 y^\top y + 2uy^\top v + v^\top v \\ &= (uy + v)^\top (uy + v) \\ &\ge 0. \end{aligned}$$

Therefore, the matrix is positive semidefinitem as required. □

By Lemma 5.2, $x^\top P^\top P x + 2p^\top x \le 2t$ is equivalent to

$$\begin{pmatrix} 2t - 2p^\top x & (Px)^\top \\ Px & I \end{pmatrix} \succeq 0.$$

Finally, we have shown that (QP) is equivalent to the following optimization problem.

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & \mathrm{Diag}(Ax - b) \succeq 0, \\ & \begin{pmatrix} 2t - 2p^\top x & (Px)^\top \\ Px & I \end{pmatrix} \succeq 0. \end{aligned}$$

References

[CET16] Richard W. Cottle, B. Curtis Eaves, and Mukund N. Thapa. *Dantzig, George B. (1914–2005)*, pages 1–14. Palgrave Macmillan UK, London, 2016. 4.3

[CLS19] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 938–942, New York, NY, USA, 2019. Association for Computing Machinery. 4.3

[Dan85] George B. Dantzig. Impact of limeiar programing on computfer dvelopment. Technical Report 85-7, Department of Operations Research, Stanford University, June 1985. 4.3

[JSWZ21] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. A faster algorithm for solving general lps. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 823–832, New York, NY, USA, 2021. Association for Computing Machinery. 4.3

[Kan39] Leonid V. Kantorovich. The mathematical method of production planning and organization. *Management Science*, 6:363–422, 1939. 4.3

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, page 302–311, New York, NY, USA, 1984. Association for Computing Machinery. 4.3

[Kha79] Leonid.G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979. 4.3

[Kha80] Leonid.G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980. 4.3

[KM72] Victor Klee and George J. Minty. How good is the simplex algorithm? In Oved Shisha, editor, *Inequalities III (Proceedings of the Third Symposium on Inequalities held at the University of California, Los Angeles, Calif., September 1–9, 1969, dedicated to the memory of Theodore S. Motzkin)*, pages 159–175, 1972. 4.3

[Sho77] Naum Z. Shor. Cut-off method with space extension in convex programming problems. *Cybernetics and systems analysis*, 13(1):94–96, 1977. 4.3

[Vai89] P.M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*, pages 332–337, 1989. 4.3

[YN76] David B. Yudin and Arkadi S. Nemirovski. Evaluation of the information com- plexity of mathematical programming problems. *Ekonomika i Matematicheskie Metody*, 12:128–142, 1976. 4.3