Supplement to “Linear and Nonlinear Programming”

David G. Luenberger and Yinyu Ye

Revised October 2005

Contents

Preface

List of Figures

Chapter 5

Interior-Point Algorithms

5.1 Introduction

Linear programming (LP), plays a distinguished role in optimization theory. In one sense it is a continuous optimization problem since the goal is to minimize a linear objective function over a convex polyhedron. But it is also a combinatorial problem involving selecting an extreme point among a finite set of possible vertices, as seen in Chapters 3 and 4.

An optimal solution of a linear program always lies at a vertex of the feasible region, which itself is a polyhedron. Unfortunately, the number of vertices associated with a set of n inequalities in m variables can be exponential in the dimension—up to $n!/m!(n-m)!$. Except for small values of m and n, this number is sufficiently large to prevent examining all possible vertices when searching for an optimal vertex.

The simplex method examines optimal candidate vertices in an intelligent fashion. As we know, it constructs a sequence of adjacent vertices with improving values of the objective function. Thus, the method travels along edges of the polyhedron until it hits an optimal vertex. Improved in various way in the intervening four decades, the simplex method continues to be the workhorse algorithm for solving linear programming problems.

Although it performs well in practice, the simplex method will examine every vertex when applied to certain linear programs. Klee and Minty in 1972 gave such an example. These examples confirm that, in the worst case, the simplex method uses a number of iterations that is exponential in the size of the problem to find the optimal solution. As interest in complexity theory grew, many researchers believed that a good algorithm should be polynomial —that is, broadly speaking, the running time required to compute the solution should be bounded above by a polynomial in the size, or the total data length, of the problem. The simplex method is not a polynomial algorithm.[1]

In 1979, a new approach to linear programming, Khachiyan's ellipsoid method, received dramatic and widespread coverage in the international press. Khachiyan proved that the ellipsoid method, developed during the 1970s by other mathematicians, is a polynomial algorithm for linear programming under a certain computational model. It constructs a sequence of shrinking ellipsoids with two properties: the current ellipsoid always contains the optimal solution set, and each member of the sequence undergoes a guaranteed reduction in volume, so that the solution set is squeezed more tightly at each iteration.

Experience with the method, however, led to disappointment. It was found that even with the best implementations the method was not even close to being competitive with the simplex method. Thus, after the dust eventually settled, the prevalent view among linear programming researchers was that Khachiyan had answered a major open question on the polynomiality of solving linear programs, but the simplex method remained the clear winner in practice.

This contradiction, the fact that an algorithm with the desirable theoretical property of polynomiality might nonetheless compare unfavorably with the (worst-case exponential) simplex method, set the stage for exciting new developments. It was no wonder, then, that the announcement by Karmarkar in 1984 of a new polynomial time algorithm, an interior-point method, with the potential to improve the practical effectiveness of the simplex method made front-page news in major newspapers and magazines throughout the world.

Interior-point algorithms are continuous iterative algorithms. Computational experience with sophisticated procedures suggests that the number of necessary iterations grows very slowly with problem size. This provides the potential for improvements in computation effectiveness for solving large-scale linear programs. The goal of this chapter is to provide some understanding on the complexity theories of linear programming and polynomial-time interior-point algorithms.

A few words on complexity theory

Complexity theory is arguably the foundation stone for analysis of computer algorithms. The goal of the theory is twofold: to develop criteria for

[1] We will be more precise about complexity notions such as "polynomial algorithm" in **S**5.1 below.

measuring the effectiveness of various algorithms (and thus, be able to compare algorithms using these criteria), and to assess the inherent difficulty of various problems.

The term *complexity* refers to the amount of resources required by a computation. In this chapter we focus on a particular resource, namely, computing time. In complexity theory, however, one is not interested in the execution time of a program implemented in a particular programming language, running on a particular computer over a particular input. This involves too many contingent factors. Instead, one wishes to associate to an algorithm more intrinsic measures of its time requirements.

Roughly speaking, to do so one needs to define:

- a notion of *input size*,
- a set of *basic operations*, and
- a *cost* for each basic operation.

The last two allow one to associate a *cost of a computation.* If x is any input, the cost $C(x)$ of the computation with input x is the sum of the costs of all the basic operations performed during this computation.

Let $\mathcal{A}$ be an algorithm and $\mathcal{I}_n$ be the set of all its inputs having size n. The *worst-case cost function* of $\mathcal{A}$ is the function $T^w_{\mathcal{A}}$ defined by

$$T^w_{\mathcal{A}}(n) = \sup_{x \in \mathcal{I}_n} C(x).$$

If there is a probability structure on $\mathcal{I}_n$ it is possible to define the *average-case cost function* $T^a_{\mathcal{A}}$ given by

$$T^a_{\mathcal{A}}(n) = \mathrm{E}_n(C(x)).$$

where E_n is the expectation over $\mathcal{I}_n$. However, the average is usually more difficult to find, and there is of course the issue of what probabilities to assign.

We now discuss how the objects in the three items above are selected. The selection of a set of basic operations is generally easy. For the algorithms we consider in this chapter, the obvious choice is the set $\{+, -, \times, /, \leq\}$ of the four arithmetic operations and the comparison. Selecting a notion of input size and a cost for the basic operations depends on the kind of data dealt with by the algorithm. Some kinds can be represented within a fixed amount of computer memory; some others require a variable amount.

Examples of the first are fixed-precision floating-point numbers, stored in a fixed amount of memory (usually 32 or 64 bits). For this kind of data the size of an element is usually taken to be 1 and consequently to have *unit size* per number.

Examples of the second are integer numbers which require a number of bits approximately equal to the logarithm of their absolute value. This (base 2) logarithm is usually referred to as the *bit size* of the integer. Similar ideas apply for rational numbers.

Let A be some kind of data and $\mathbf{x} = (x_1, \ldots, x_n) \in A^n$. If A is of the first kind above then we define $\text{size}(\mathbf{x}) = n$. Otherwise, we define $\text{size}(\mathbf{x}) = \sum_{i=1}^{n} \text{bit-size}(x_i)$.

The cost of operating on two unit-size numbers is taken to be 1 and is called *unit cost*. In the bit-size case, the cost of operating on two numbers is the product of their bit-sizes (for multiplications and divisions) or its maximum (for additions, subtractions, and comparisons).

The consideration of integer or rational data with their associated bit size and bit cost for the arithmetic operations is usually referred to as the *Turing model of computation*. The consideration of idealized reals with unit size and unit cost is today referred as the *real number arithmetic model*. When comparing algorithms, one should make clear which model of computation is used to derive complexity bounds.

A basic concept related to both models of computation is that of *polynomial time*. An algorithm $\mathcal{A}$ is said to be a polynomial time algorithm if $T^w_{\mathcal{A}}$ is bounded above by a polynomial. A problem can be solved in polynomial time if there is a polynomial time algorithm solving the problem. The notion of *average polynomial time* is defined similarly, replacing $T^w_{\mathcal{A}}$ by $T^a_{\mathcal{A}}$.

The notion of polynomial time is usually taken as the formalization of efficiency and it is the ground upon which complexity theory is built.

5.2 The simplex method is not polynomial-time*

When the simplex method is used to solve a linear program in standard form with the coefficient matrix $\mathbf{A} \in R^{m \times n}$, $\mathbf{b} \in R^m$ and $\mathbf{c} \in R^n$, the number of iterations to solve the problem starting from a basic feasible solution is typically a small multiple of m: usually between $2m$ and $3m$. In fact, Dantzig observed that for problems with $m \leq 50$ and $n \leq 200$ the number of iterations is ordinarily less than $1.5m$.

At one time researchers believed—and attempted to prove—that the simplex algorithm (or some variant thereof) always requires a number of iterations that is bounded by a polynomial expression in the problem size. That was until Victor Klee and George Minty published their landmark

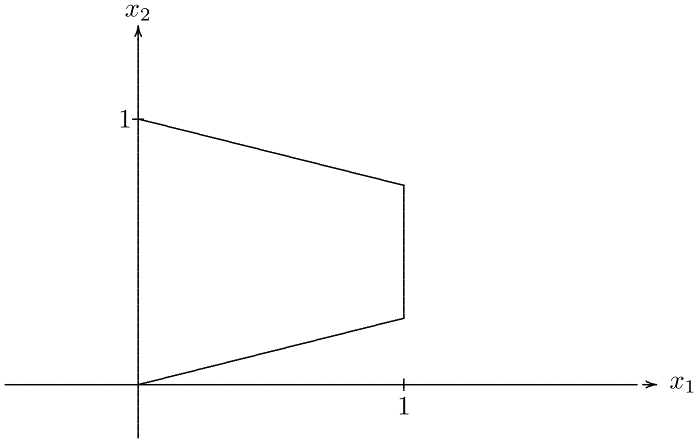

Figure 5.1: Feasible region for Klee-Minty problem;$n = 2$.

paper that exhibited a class of linear programs each of which requires an exponential number of iterations when solved by the conventional simplex method.

As originally stated, the Klee–Minty problems are not in standard form; they are expressed in terms of $2n$ linear inequalities in n variables. Their feasible regions are perturbations of the *unit cube* in n-space; that is,

$$[0,1]^n = \{x : 0 \le x_j \le 1, \quad j = 1, \dots, n\}.$$

One way to express a problem in this class is

$$\begin{array}{ll} \text{maximize} & x_n \\ \text{subject to} & x_1 \ge 0 \\ & x_1 \le 1 \\ & x_j \ge \varepsilon x_{j-1} \qquad j = 2, \dots, n \\ & x_j \le 1 - \varepsilon x_{j-1} \quad j = 2, \dots, n \end{array}$$

where $0 < \varepsilon < 1/2$. This presentation of the problem emphasizes the idea that the feasible region of the problem is a perturbation of the n-cube.

In the case of $n = 2$ and $\varepsilon = 1/4$, the feasible region of the linear program above looks like that of Figure 5.1

For the case where $n = 3$, the feasible region of the problem above looks like that of Figure 5.2

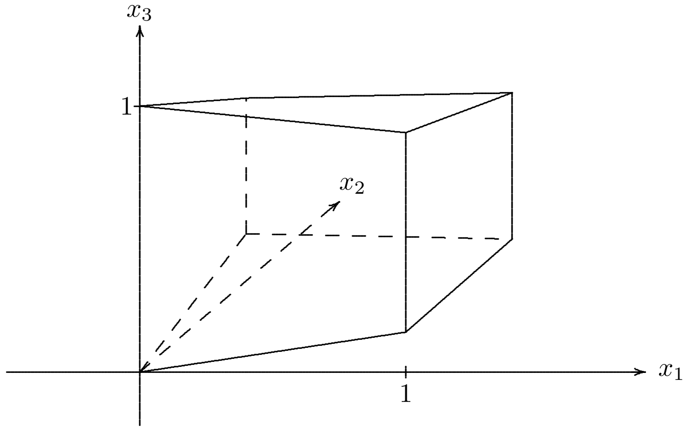

Figure 5.2: The case $n = 3$.

Consider

$$\begin{array}{lrll} \text{maximize} & \displaystyle\sum_{j=1}^{n} 10^{n-j} x_j & & \\ \text{subject to} & \displaystyle 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i & \leq 100^{i-1} & i = 1, \ldots, n \\ & x_j & \geq 0 & j = 1, \ldots, n \end{array} \tag{5.1}$$

The problem above is easily cast as a linear program in standard form.

Example 5.1 Suppose we want to solve the linear program

$$\begin{array}{lrcrcrcr} \text{maximize} & 100x_1 & + & 10x_2 & + & x_3 & & \\ \text{subject to} & x_1 & & & & & \leq & 1 \\ & 20x_1 & + & x_2 & & & \leq & 100 \\ & 200x_1 & + & 20x_2 & + & x_3 & \leq & 10,000 \end{array}$$

In this case, we have three constraints and three variables (along with their nonnegativity constraints). After adding slack variables, we get a problem in standard form. The system has $m = 3$ equations and $n = 6$ nonnegative variables. In tableau form, the problem is

	Variable	x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{b}$
T^0	4	1	0	0	1	0	0	1
	5	20	1	0	0	1	0	100
	6	200	20	1	0	0	1	10,000
	$\mathbf{c}^T$	100	10	1	0	0	0	0
					●	●	●	

The bullets below the tableau indicate the columns that are basic.

Note that we are maximizing, so the goal is to find a feasible basis that prices out *nonpositive*. In the objective function, the nonbasic variables x_1, x_2, and x_3 have coefficients 100, 10, and 1, respectively. Using the greedy rule[2] for selecting the incoming variable (see Section 3.8, Step 2 of the revised simplex method), we start making x_1 positive and find (by the minimum ratio test) that x_4 becomes nonbasic. After pivoting on the element in row 1 and column 1, we obtain a sequence of tables:

	Variable	x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{b}$
T^1	1	1	0	0	1	0	0	1
	5	0	1	0	–20	1	0	80
	6	0	20	1	–200	0	1	9,800
	$\mathbf{r}^T$	0	10	1	–100	0	0	–100
		●				●	●	

	Variable	x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{b}$
T^2	1	1	0	0	1	0	0	1
	2	0	1	0	–20	1	0	80
	6	0	0	1	200	–20	1	8,200
	$\mathbf{r}^T$	0	0	1	100	–10	0	–900
		●	●				●	

	Variable	x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{b}$
T^3	4	1	0	0	1	0	0	1
	2	20	1	0	0	1	0	100
	6	–200	0	1	0	–20	1	8,000
	$\mathbf{r}^T$	–100	0	1	0	–10	0	–1,000
			●		●		●	

	Variable	x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{b}$
T^4	4	1	0	0	1	0	0	1
	2	20	1	0	0	1	0	100
	3	–200	0	1	0	–20	1	8,000
	$\mathbf{r}^T$	100	0	0	0	10	–1	–9,000
			●	●	●			

[2] That is, selecting the pivot column as that with the largest "reduced cost" coefficient.

T^5

Variable	x_1	x_2	x_3	x_4	x_5	x_6	**b**
1	1	0	0	1	0	0	1
2	0	1	0	–20	1	0	80
3	0	0	1	200	–20	1	8,200
$\mathbf{r}^T$	0	0	0	–100	10	–1	–9,100
	●	●	●				

T^6

Variable	x_1	x_2	x_3	x_4	x_5	x_6	**b**
1	1	0	0	1	0	0	1
5	0	1	0	–20	1	0	80
3	0	20	1	–200	0	1	9,800
$\mathbf{r}^T$	0	–10	0	100	0	–1	–9,900
	●		●		●		

T^7

Variable	x_1	x_2	x_3	x_4	x_5	x_6	**b**
4	1	0	0	1	0	0	1
5	20	1	0	0	1	0	100
3	200	20	1	0	0	1	10,000
$\mathbf{r}^T$	–100	–10	0	0	0	–1	–10,000
			●	●	●		

From T^7 we see that the corresponding basic feasible solution

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (0, 0, 10^4, 1, 10^2, 0)$$

is optimal and that the objective function value is $10,000$. Along the way, we made $2^3 - 1 = 7$ pivot steps. The objective function strictly increased with each change of basis.

We see that the instance of the linear program (5.1) with $n = 3$ leads to $2^3 - 1$ pivot steps when the greedy rule is used to select the pivot column. The general problem of the class (5.1) takes $2^n - 1$ pivot steps. To get an idea of how bad this can be, consider the case where $n = 50$. Now $2^{50} - 1 \approx 10^{15}$. In a year with 365 days, there are approximately 3×10^7 seconds. If a computer ran continuously performing a hundred thousand iterations of the simplex algorithm per second, it would take approximately

$$\frac{10^{15}}{3 \times 10^5 \times 10^8} \approx 33 \quad \text{years}$$

to solve the problem using the greedy pivot selection rule[3].

[3] A path that visits each vertex of a unit n-cube once and only once is said to be *Hamiltonian path.* In this example, the simplex method generates a sequence of points which yields a Hamiltonian path on the cube. There is an amusing recreational literature that connects the Hamiltonian path with certain puzzles. See Martin Gardner in the references.

5.3 The Ellipsoid Method

The basic ideas of the ellipsoid method stem from research done in the nineteen sixties and seventies mainly in the Soviet Union (as it was then called) by others who preceded Khachiyan. The idea in a nutshell is to enclose the region of interest in each member of a sequence of ever smaller ellipsoids.

The significant contribution of Khachiyan was to demonstrate in two papers—published in 1979 and 1980—that under certain assumptions, the ellipsoid method constitutes a polynomially bounded algorithm for linear programming.

The method discussed here is really aimed at finding a point of a polyhedral set Ω given by a system of linear inequalities.

$$\Omega = \{\mathbf{y} \in R^m : \mathbf{y}^T \mathbf{a}_j \leq c_j, \quad j = 1, \dots n\}$$

Finding a point of Ω can be thought of as being equivalent to solving a linear programming problem.

Two important assumptions are made regarding this problem:

(A1) There is a vector $\mathbf{y}_0 \in R^m$ and a scalar $R > 0$ such that the closed ball $S(\mathbf{y}_0, R)$ with center $\mathbf{y}_0$ and radius R, that is

$$\{\mathbf{y} \in R^m : |\mathbf{y} - \mathbf{y}_0| \leq R\},$$

contains Ω.

(A2) There is a known scalar $r > 0$ such that if Ω is nonempty, then it contains a ball of the form $S(\mathbf{y}^*, r)$ with center at $\mathbf{y}^*$ and radius r. (This assumption implies that if Ω is nonempty, then it has a nonempty interior and its volume is at least $\mathrm{vol}(S(\mathbf{0}, r)))$[4].

Definition 5.1 An *Ellipsoid* in R^m is a set of the form

$$E = \{\mathbf{y} \in R^m : (\mathbf{y} - \mathbf{z})^T \mathbf{Q} (\mathbf{y} - \mathbf{z}) \leq 1\}$$

where $\mathbf{z} \in R^m$ is a given point (called the *center*) and $\mathbf{Q}$ is a positive definite matrix (see Section A.4 of Appendix A) of dimension m. This ellipsoid is denoted $\mathrm{ell}(\mathbf{z}, \mathbf{Q})$.

The unit sphere $S(\mathbf{0}, 1)$ centered at the origin $\mathbf{0}$ is a special ellipsoid with $\mathbf{Q} = \mathbf{I}$, the identity matrix.

[4] The (topological) interior of any set Ω is the set of points in Ω which are the centers of some balls contained in Ω.

The axes of a general ellipsoid are the eigenvectors of $\mathbf{Q}$ and the lengths of the axes are $\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_m^{-1/2}$, where the λ_i's are the corresponding eigenvalues. It is easily seen that the volume of an ellipsoid is

$$\mathrm{vol}(E) = \mathrm{vol}(S(\mathbf{0},1))\Pi_{i=1}^{m}\lambda_i^{-1/2} = \mathrm{vol}(S(\mathbf{0},1))\det(\mathbf{Q}^{-1/2}).$$

Cutting plane and new containing ellipsoid

In the ellipsoid method, a series of ellipsoids E_k are defined, with centers $\mathbf{y}_k$ and with the defining $\mathbf{Q} = \mathbf{B}_k^{-1}$, where $\mathbf{B}_k$ is symmetric and positive definite.

At each iteration of the algorithm, we will have $\Omega \subset E_k$. It is then possible to check whether $\mathbf{y}_k \in \Omega$. If so, we have found an element of Ω as required. If not, there is at least one constraint that is violated. Suppose $\mathbf{a}_j^T\mathbf{y}_k > c_j$. Then

$$\Omega \subset \frac{1}{2}E_k := \{\mathbf{y} \in E_k : \mathbf{a}_j^T\mathbf{y} \le \mathbf{a}_j^T\mathbf{y}_k\}$$

This set is half of the ellipsoid, obtained by cutting the ellipsoid in half through its center.

The successor ellipsoid E_{k+1} will be the minimal-volume ellipsoid containing $\frac{1}{2}E_k$. It is constructed as follows. Define

$$\tau = \frac{1}{m+1}, \quad \delta = \frac{m^2}{m^2-1}, \quad \sigma = 2\tau.$$

Then put

$$\begin{aligned}\mathbf{y}_{k+1} &= \mathbf{y}_k - \frac{\tau}{(\mathbf{a}_j^T\mathbf{B}_k\mathbf{a}_j)^{1/2}}\mathbf{B}_k\mathbf{a}_j \\ \mathbf{B}_{k+1} &= \delta\left(\mathbf{B}_k - \sigma\frac{\mathbf{B}_k\mathbf{a}_j\mathbf{a}_j^T\mathbf{B}_k}{\mathbf{a}_j^T\mathbf{B}_k\mathbf{a}_j}\right) \qquad (5.2)\end{aligned}$$

.

Theorem 5.1 *The ellipsoid $E_{k+1} = \mathrm{ell}(\mathbf{y}_{k+1}, \mathbf{B}_{k+1}^{-1})$ defined as above is the ellipsoid of least volume containing $\frac{1}{2}E_k$. Moreover,*

$$\frac{\mathrm{vol}(E_{k+1})}{\mathrm{vol}(E_k)} = \left(\frac{m^2}{m^2-1}\right)^{(m-1)/2}\frac{m}{m+1} < \exp\left(-\frac{1}{2(m+1)}\right) < 1.$$

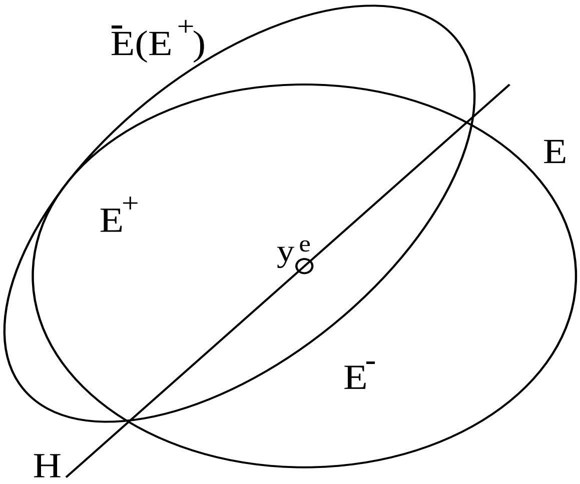

Figure 5.3: Illustration of the minimal-volume ellipsoid containing a half-ellipsoid.

Proof. We shall not prove the statement about the new ellipsoid being of least volume, since that is not necessary for the results that follow. To prove the remainder of the statement, we have

$$\frac{\mathrm{vol}(E_{k+1})}{\mathrm{vol}(E_k)} = \frac{\det(\mathbf{B}_{k+1}^{1/2})}{\det(\mathbf{B}_k^{1/2})}$$

For simplicity, by a change of coordinates, we may take $\mathbf{B}_k = \mathbf{I}$. Then $\mathbf{B}_{k+1}$ has $m-1$ eigenvalues equal to $\delta = \frac{m^2}{m^2-1}$ and one eigenvalue equal to $\delta - 2\delta\tau = \frac{m^2}{m^2-1}(1-\frac{2}{m+1}) = (\frac{m}{m+1})^2$. The reduction in volume is the product of the square roots of these, giving the equality in the theorem.

Then using $(1+x)^p \le e^{xp}$, we have

$$\begin{aligned}\left(\frac{m^2}{m^2-1}\right)^{(m-1)/2}\frac{m}{m+1} &= \left(1+\frac{1}{m^2-1}\right)^{(m-1)/2}\left(1-\frac{1}{m+1}\right)\\ &< \exp\left(\frac{1}{2(m+1)}-\frac{1}{(m+1)}\right) = \exp\left(-\frac{1}{2(m+1)}\right).\end{aligned}$$

□

Convergence

The ellipsoid method is initiated by selecting $\mathbf{y}_0$ and R such that condition (A1) is satisfied. Then $\mathbf{B}_0 = R^2\mathbf{I}$, and the corresponding E_0 contains Ω. The updating of the E_k's is continued until a solution is found.

Under the assumptions stated above, a single repetition of the ellipsoid method reduces the volume of an ellipsoid to one-half of its initial value in $O(m)$ iterations. Hence it can reduce the volume to less than that of a sphere of radius r in $O(m^2\log(R/r))$ iterations, since it volume is bounded from below by $\mathrm{vol}(S(\mathbf{0},1))r^m$ and the initial volume is $\mathrm{vol}(S(\mathbf{0},1))R^m$. Generally a single iteration requires $O(m^2)$ arithmetic operations. Hence the entire process requires $O(m^4\log(R/r))$ arithmetic operations.[5]

Ellipsoid method for usual form of LP

Now consider the linear program (where $\mathbf{A}$ is $m \times n$)

$$\text{(P)} \qquad \begin{array}{cc} \text{maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \ge \mathbf{0} \end{array}$$

and its dual

$$\text{(D)} \qquad \begin{array}{cc} \text{minimize} & \mathbf{y}^T\mathbf{b} \\ \text{subject to} & \mathbf{y}^T\mathbf{A} \ge \mathbf{c}^T \\ & \mathbf{y} \ge \mathbf{0}. \end{array}$$

Both problems can be solved by finding a feasible point to inequalities

$$\begin{array}{rcl} -\mathbf{c}^T\mathbf{x} + \mathbf{b}^T\mathbf{y} & \le & 0 \\ \mathbf{A}\mathbf{x} & \le & \mathbf{b} \\ -\mathbf{A}^T\mathbf{y} & \le & -\mathbf{c} \\ \mathbf{x}, \mathbf{y} & \ge & \mathbf{0}, \end{array} \tag{5.3}$$

where both $\mathbf{x}$ and $\mathbf{y}$ are variables. Thus, the total number of arithmetic operations of solving a linear program is bounded by $O((m+n)^4\log(R/r))$.

5.4 The Analytic Center

The new form of interior-point algorithm introduced by Karmarkar moves by successive steps inside the feasible region. It is the interior of the feasible

[5] Assumption (A2) is sometimes too strong. It has been shown, however, that when the data consists of integers, it is possible to perturb the problem so that (A2) is satisfied and if the perturbed problem has a feasible solution, so does the original Ω.

set rather than the vertices and edges that plays a dominant role in this type of algorithm. In fact, these algorithms purposely avoid the edges of the set, only eventually converging to one as a solution.

The study of these algorithms begins in the next section, but it is useful at this point to introduce a concept that definitely focuses on the interior of a set, termed the set's analytic center. As the name implies, the center is away from the edge.

In addition, the study of the analytic center introduces a special structure, termed a *barrier* that is fundamental to interior-point methods.

Consider a set $\mathcal{S}$ in a subset of $\mathcal{X}$ of R^n defined by a group of inequalities as

$$\mathcal{S} = \{\mathbf{x} \in \mathcal{X} : g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \ldots, m\},$$

and assume that the functions g_j are continuous. $\mathcal{S}$ has a nonempty interior $\overset{\circ}{\mathcal{S}}= \{\mathbf{x} \in \mathcal{X} : g_j(\mathbf{x}) > 0, \text{ all } j\}$. Associated with this definition of the set is the *potential function*

$$\psi(\mathbf{x}) = -\sum_{j=1}^{m} \log\, g_j(\mathbf{x})$$

defined on $\overset{\circ}{\mathcal{S}}$.

The *analytic center* of $\mathcal{S}$ is the vector (or set of vectors) that minimizes the potential; that is, the vector (or vectors) that solve

$$\min \psi(\mathbf{x}) = \min \left\{ -\sum_{j=1}^{m} \log g_j(\mathbf{x}) : \mathbf{x} \in \mathcal{X},\ g_j(\mathbf{x}) > 0 \text{ for each } j \right\}.$$

Example 5.2 A cube Consider the set S defined by $x_i \geq 0,\ (1 - x_i) \geq 0$, for $i = 1, 2, \ldots, n$. This is $\mathcal{S} = [0, 1]^n$, the unit cube in R^n. The analytic center can be found by differentiation to be $x_i = 1/2$, for all i. Hence, the analytic center is identical to what one would normally call the center of the unit cube.

In general, the analytic center depends on how the set is defined—on the particular inequalities used in the definition. For instance, the unit cube is also defined by the inequalities $x_i \geq 0,\ (1 - x_i)^d \geq 0$ with $d > 1$. In this case the solution is $x_i = 1/(d + 1)$ for all i. For large d this point is near the inner corner of the unit cube.

Also, the additional of redundant inequalities can also change the location of the analytic center. For example, repeating a given inequality will change the center's location.

There are several sets associated with linear programs for which the analytic center is of particular interest. One such set is the feasible region itself. Another is the set of optimal solutions. There are also sets associated with the dual and primal-dual formulations. All of these are related in important ways.

Let us illustrate, by considering the analytic center associated with a bounded polytope Ω in $\mathcal{R}^m$ represented by n $(> m)$ linear inequalities; that is,

$$\Omega = \{\mathbf{y} \in R^m : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} \geq \mathbf{0}\},$$

where $\mathbf{A} \in R^{m\times n}$ and $\mathbf{c} \in R^n$ are given and $\mathbf{A}$ has rank m. Denote the interior of Ω by

$$\overset{\circ}{\Omega}= \{\mathbf{y} \in R^m : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} > \mathbf{0}\}.$$

The potential function for this set is

$$\psi_\Omega(\mathbf{y}) \equiv -\sum_{j=1}^{n} \log(c_j - \mathbf{y}^T\mathbf{a}_j) = -\sum_{j=1}^{n} \log s_j, \tag{5.4}$$

where $\mathbf{s} \equiv \mathbf{c} - \mathbf{A}^T\mathbf{y}$ is a *slack vector.* Hence the potential function is the negative sum of the logarithms of the slack variables.

The analytic center of Ω is the interior point of Ω that minimizes the potential function. This point is denoted by $\mathbf{y}^a$ and has the associated $\mathbf{s}^a = \mathbf{c} - \mathbf{A}^T\mathbf{y}^a$. $(\mathbf{y}^a, \mathbf{s}^a)$ is uniquely defined, since the potential function is strictly convex in a bounded convex Ω.

Setting to zero the derivatives of $\psi(\mathbf{y})$ with respect to each y_i gives

$$\sum_{j=1}^{m} \frac{a_{ij}}{c_j - \mathbf{y}^T\mathbf{a}_j} = 0, \text{ for all } i.$$

which can be written

$$\sum_{j=1}^{m} \frac{a_{ij}}{s_j} = 0, \text{ for all } i.$$

Now define $x_i = 1/s_j$ for each i. We introduce the notion

$$\mathbf{x} \circ \mathbf{s} \equiv (x_1s_1, x_2s_2, \ldots, x_ns_n)^T,$$

which is *component multiplication.* Then the analytic center is defined by the conditions

$$\begin{aligned} \mathbf{x} \circ \mathbf{s} &= \mathbf{1} \\ \mathbf{A}\mathbf{x} &= \mathbf{0} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} &= \mathbf{c}. \end{aligned}$$

The analytic center can be defined when the interior is empty or equalities are present, such as

$$\Omega = \{\mathbf{y} \in R^m : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} \geq \mathbf{0},\ \mathbf{B}\mathbf{y} = \mathbf{b}\}.$$

In this case the analytic center is chosen on the linear surface $\{\mathbf{y} : \mathbf{B}\mathbf{y} = \mathbf{b}\}$ to maximize the product of the slack variables $\mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y}$. Thus, in this context the interior of Ω refers to the interior of the positive orthant of slack variables: $R^n_+ \equiv \{\mathbf{s} :\ \mathbf{s} \geq \mathbf{0}\}$. This definition of interior depends only on the region of the slack variables. Even if there is only a single point in Ω with $\mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y}$ for some $\mathbf{y}$ where $\mathbf{B}\mathbf{y} = \mathbf{b}$ with $\mathbf{s} > \mathbf{0}$, we still say that $\overset{\circ}{\Omega}$ is not empty.

5.5 The Central Path

The concept underlying interior-point methods for linear programming is to use nonlinear programming techniques of analysis and methodology. The analysis is often based on differentiation of the functions defining the problem. Traditional linear programming does not require these techniques since the defining functions are linear. Duality in general nonlinear programs is typically manifested through Lagrange multipliers (which are called dual variables in linear programming). The analysis and algorithms of the remaining sections of the chapter use these nonlinear techniques. These techniques are discussed systematically in later chapters, so rather than treat them in detail at this point, these current sections provide only minimal detail in their application to linear programming. It is expected that most readers are already familiar with the basic method for minimizing a function by setting its derivative to zero, and for incorporating constraints by introducing Lagrange multipliers. These methods are discussed in detail in Chapters 11-15.

The computational algorithms of nonlinear programming are typically iterative in nature, often characterized as search algorithms. When at any point, a direction for search is established and then a move in that direction is made to define the next point. There are many varieties of such search algorithms and they are systematically presented in chapters blank through blank. In this chapter, we use versions of Newton's method as the search algorithm, but we postpone a detailed study of the method until later chapters.

The transfer of ideas has gone in two directions. The ideas of interior-point methods for linear programming have been extended to provide new approaches to nonlinear programming as well. This chapter is intended to show how this merger of linear and nonlinear programming produces elegant and effective methods. And these ideas take an especially pleasing form when applied to linear programming. Study of them here, even without all the detailed analysis, should provide good intuitive background for the more general manifestations.

Consider a primal linear program in standard form

$$\begin{array}{lll} \text{(LP)} & \text{minimize} & \mathbf{c}^T\mathbf{x} \\ & \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & & \mathbf{x} \geq \mathbf{0}. \end{array} \tag{5.5}$$

We denote the feasible region of this program by $\mathcal{F}_p$. We assume that $\overset{\circ}{\mathcal{F}} = \{\mathbf{x} : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} > \mathbf{0}\}$ is nonempty and the optimal solution set of the problem is bounded.

Associated with this problem, we define for $\mu \geq 0$ the *barrier problem*

$$\begin{array}{lll} \text{(BP)} & \text{minimize} & \mathbf{c}^T\mathbf{x} - \mu \sum_{j=1}^{n} \log x_j \\ & \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & & \mathbf{x} > \mathbf{0}. \end{array} \tag{5.6}$$

It is clear that $\mu = 0$ corresponds to the original problem (5.5). As $\mu \to \infty$, the solution approaches the analytic center of the feasible region (when it is bounded), since the barrier term swamps out $\mathbf{c}^T\mathbf{x}$ in the objective. As μ is varied continuously toward 0, there is a path $\mathbf{x}(\mu)$ defined by the solution to (BP). This path $\mathbf{x}(\mu)$ is termed the *primal central path.* As $\mu \to 0$ this path converges to the analytic center of the optimal face $\{\mathbf{x} : \mathbf{c}^T\mathbf{x} = z^*, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where z^* is the optimal value of (LP).

A strategy for solving (LP) is to solve (BP) for smaller and smaller values of μ and thereby approach a solution to (LP). This is indeed the basic idea of interior-point methods.

At any $\mu > 0$, under the assumptions that we have made for problem (5.5), the necessary and sufficient conditions for a unique and bounded solution are obtained by introducing a *Lagrange multiplier* vector $\mathbf{y}$ for the linear equality constraints to form the *Lagrangian* (see Chapter ?)

$$\mathbf{c}^T\mathbf{x} - \mu \sum_{j=1}^{n} \log x_j - \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}).$$

The derivatives with respect to the x_j's are set to zero, leading to the conditions

$$c_j - \mu/x_j - \mathbf{y}^T\mathbf{a}_j = 0, \text{ for each } j$$

or

$$\mu \mathbf{X}^{-1}\mathbf{1} + \mathbf{A}^T\mathbf{y} \;=\; c \tag{5.7}$$

where as before $\mathbf{a}_j$ is the j-th column of $\mathbf{A}$ and $\mathbf{X}$ is the diagonal matrix whose diagonal entries are $\mathbf{x} > \mathbf{0}$. Setting $s_j = \mu/x_j$ the complete set of conditions can be rewritten

$$\begin{aligned} \mathbf{x} \circ \mathbf{s} &= \mu\mathbf{1} \\ \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} &= \mathbf{c}. \end{aligned} \tag{5.8}$$

Note that $\mathbf{y}$ is a dual feasible solution and $\mathbf{c} - \mathbf{A}^T\mathbf{y} > \mathbf{0}$ (see Exercise 5.3).

Example 5.3 A square primal Consider the problem of maximizing x_1 within the unit square $S = [0,1]^2$. The problem is formulated as

$$\begin{array}{lc} \max & x_1 \\ \text{subject to} & x_1 + x_3 = 1 \\ & x_2 + x_4 = 1 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0. \end{array}$$

Here x_3 and x_4 are slack variables for the original problem to put it in standard form. The optimality conditions for $\mathbf{x}(\mu)$ consist of the original 2 linear constraint equations and the four equations

$$\begin{aligned} y_1 + s_1 &= 1 \\ y_2 + s_2 &= 0 \\ y_1 + s_3 &= 0 \\ y_2 + s_4 &= 0 \end{aligned}$$

together with the relations $s_i = \mu/x_i$ for $i = 1, 2 \ldots, 4$. These equations are readily solved with a series of elementary variable eliminations to find

$$\begin{aligned} x_1(\mu) &= \frac{1 - 2\mu \pm \sqrt{1 + 4\mu^2}}{2} \\ x_2(\mu) &= 1/2. \end{aligned}$$

Using the "+" solution, it is seen that as $\mu \to 0$ the solution goes to $x \to (1, 1/2)$. Note that this solution is not a corner of the cube. Instead it is at the analytic center of the optimal face $\{x : x_1 = 1,\ 0 \leq x_2 \leq 1\}$. The limit of $\mathbf{x}(\mu)$ as $\mu \to \infty$ can be seen to be the point $(1/2, 1/2)$. Hence, the central path in this case is a straight line progressing from the analytic center of the square (at $\mu \to \infty$) to the analytic center of the optimal face (at $\mu \to 0$).

Dual central path

Now consider the dual problem

$$\begin{array}{lll} \text{(LD)} & \text{maximize} & \mathbf{y}^T\mathbf{b} \\ & \text{subject to} & \mathbf{y}^T\mathbf{A} + \mathbf{s}^T = \mathbf{c}^T \\ & & \mathbf{s} \geq \mathbf{0}. \end{array}$$

We may apply the barrier approach to this problem by formulating the problem

$$\begin{array}{lll} \text{(BD)} & \text{maximize} & \mathbf{y}^T\mathbf{b} + \mu \sum_{j=1}^{n} \log s_j \\ & \text{subject to} & \mathbf{y}^T\mathbf{A} + \mathbf{s}^T = \mathbf{c}^T \\ & & \mathbf{s} > \mathbf{0}. \end{array}$$

We assume that the dual feasible set $\mathcal{F}_d$ has an interior $\overset{\circ}{\mathcal{F}}_d = \{(\mathbf{y}, \mathbf{s}) : \mathbf{y}^T\mathbf{A} + \mathbf{s}^T = \mathbf{c}^T,\ \mathbf{s} > \mathbf{0}\}$ is nonempty and the optimal solution set of (LD) is bounded. Then, as μ is varied continuously toward 0, there is a path $(\mathbf{y}(\mu), \mathbf{s}(\mu))$ defined by the solution to (BD). This path $\mathbf{y}(\mu)$ is termed the *dual central path.*

To work out the necessary and sufficient conditions we introduce $\mathbf{x}$ as a Lagrange multiplier and form the Lagrangian

$$\mathbf{y}^T\mathbf{b} + \mu \sum_{j=1}^{n} \log\, s_j - (\mathbf{y}^T\mathbf{A} + \mathbf{s}^T - \mathbf{c}^T)\mathbf{x}.$$

Setting to zero the derivative with respect to y_i leads to

$$b_i - \mathbf{a}_{i.}\mathbf{x} = 0, \text{ for all } i$$

where $\mathbf{a}_{.i}$ is the i-th row of A. Setting to zero the derivative with respect to s_j leads to

$$\mu/s_j - x_j = 0, \text{ for all } j.$$

Combining these equations and including the original constraint yields the complete set of conditions

$$\begin{array}{rcl} \mathbf{x} \circ \mathbf{s} & = & \mu\mathbf{1} \\ \mathbf{A}\mathbf{x} & = & \mathbf{b} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} & = & \mathbf{c}. \end{array}$$

These are identical to the optimality conditions for the primal central path (5.8). Note that $\mathbf{x}$ is a primal feasible solution and $\mathbf{x} > \mathbf{0}$.

To see the geometric representation of the dual central path, consider the dual level set

$$\Omega(z) = \{\mathbf{y} : \ \mathbf{c}^T - \mathbf{y}^T\mathbf{A} \geq \mathbf{0}, \ \mathbf{y}^T\mathbf{b} \geq z\}$$

for any $z < z^*$ where z^* is the optimal value of (LD). Then, the analytic center $(\mathbf{y}(z), \mathbf{s}(z))$ of $\Omega(z)$ coincides the dual central path as z tends to the optimal value z^* from below (Figure 5.4).

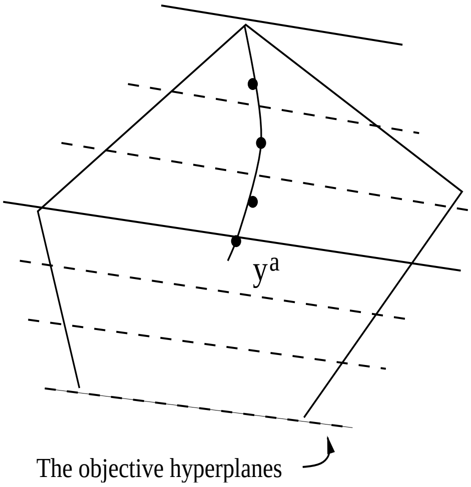

Figure 5.4: The central path as analytic centers in the dual feasible region.

Example 5.4 The square dual Consider the dual of the LP of example 5.3. The feasible region is the set of vectors $\mathbf{y}$ with $y_1 \leq -1, \ y_2 \leq 0$. (The values of s_1 and s_2 are the slack variables of these inequalities.) The solution to the dual barrier problem is easily found from the solution of the primal barrier problem to be

$$y_1(\mu) = -1 - \mu/x_1(\mu), \quad y_2 = -2\mu.$$

As $\mu \to 0$, we have $y_1 \to -1, \ y_2 \to 0$, which is the unique solution to the dual LP. However, as $\mu \to \infty$, the vector y is unbounded, for in this case the dual feasible set is itself unbounded.

Primal–dual central path

Suppose the feasible region of the primal (LP) has interior points and its optimal solution set is bounded. Then, the dual also has interior points (again see Exercise 5.3). The primal–dual path is defined to be the set of vectors $(\mathbf{x}(\mu), \mathbf{y}(\mu), \mathbf{s}(\mu))$ that satisfy the conditions

$$\begin{array}{rcl} \mathbf{x} \circ \mathbf{s} & = & \mu \mathbf{1} \\ \mathbf{A}\mathbf{x} & = & \mathbf{b} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} & = & \mathbf{c} \\ \mathbf{x},\ \mathbf{s} & \geq & \mathbf{0} \end{array} \tag{5.9}$$

for $0 \leq \mu \leq \infty$. Hence the central path is defined without explicit reference to any optimization problem. It is simply defined in terms of the set of equality and inequality conditions.

Since conditions (5.8) and (5.9) are identical, the primal–dual central path can be split into two components by projecting onto the relevant space, as described in the following proposition.

Proposition 5.2 Suppose the feasible sets of the primal and dual programs contain interior points. Then the primal–dual central path $\mathbf{x}(\mu), \mathbf{y}(\mu), \mathbf{s}(\mu))$ exists for all μ, $0 \leq \mu < \infty$. Furthermore, $\mathbf{x}(\mu)$ is the primal central path, and $(\mathbf{y}(\mu), \mathbf{s}(\mu))$ is the dual central path.

Moreover, the central path converges to the primal–dual optimal solution set, as stated below.

Proposition 5.3 Suppose the feasible sets of the primal and dual programs contain interior points. Then $\mathbf{x}(\mu)$ and $(\mathbf{y}(\mu), \mathbf{s}(\mu))$ converge to the analytic centers of the optimal primal solution and dual solution faces, respectively, as $\mu \to 0$.

Duality gap

Let $(\mathbf{x}(\mu), \mathbf{y}(\mu), \mathbf{s}(\mu))$ be on the primal-dual central path. Then from (5.9) it follows that

$$\mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{y}^T\mathbf{A}\mathbf{x} + \mathbf{s}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{s}^T\mathbf{x} = n\mu.$$

The value $\mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{s}^T\mathbf{x}$ is the difference between the primal objective value and the dual objective value. This value is always nonnegative (see the weak duality lemma in Section 4.2) and is termed the *duality gap*.

The duality gap provides a measure of closeness to optimality. For any primal feasible $\mathbf{x}$, the value $\mathbf{c}^T\mathbf{x}$ gives an upper bound as $\mathbf{c}^T\mathbf{x} \geq z^*$ where

	P-F	D-F	0-Duality
Primal Simplex	$\surd$		$\surd$
Dual Simplex		$\surd$	$\surd$
Primal Barrier	$\surd$		
Primal-Dual Path-Following	$\surd$	$\surd$	
Primal-Dual Potential-Reduction	$\surd$	$\surd$	

z^* is the optimal value of the primal. Likewise, for any dual feasible pair $(\mathbf{y}, \mathbf{s})$, the value $\mathbf{y}^T\mathbf{b}$ gives a lower bound as $\mathbf{y}^T\mathbf{b} \leq z^*$. The difference, the duality gap $g = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b}$, provides a bound on z^* as $z^* \geq \mathbf{c}^T\mathbf{x} - g$. Hence if at a feasible point $\mathbf{x}$, a dual feasible $(\mathbf{y}, \mathbf{s})$ is available, the quality of $\mathbf{x}$ can be measured as $\mathbf{c}^T\mathbf{x} - z^* \leq g$.

At any point on the primal–dual central path, the duality gap is equal to $n\mu$. Hence, this value gives a measure of optimality. It is clear that as $\mu \to 0$ the duality gap goes to zero, and hence both $\mathbf{x}(\mu)$ and $(\mathbf{y}(\mu), \mathbf{s}(\mu))$ approach optimality for the primal and dual, respectively.

5.6 Solution Strategies

The various definitions of the central path directly suggest corresponding strategies for solution of an LP. We outline three general approaches here: the primal barrier or path-following method, the primal-dual path-following method and the primal-dual potential-reduction method, although the details of their implementation and analysis must be deferred to later chapters after study of general nonlinear methods. The following table depicts these solution strategies and the simplex methods described in Chapters 3 and 4 with respect to how they meet the three optimality conditions: Primal Feasibility (P-F), Dual Feasibility (D-F), and Zero-Duality during the iterative process: For example, the primal simplex method keeps improving a primal feasible solution, maintains the zero-duality gap (complementarity condition) and moves toward dual feasibility; while the dual simplex method keeps improving a dual feasible solution, maintains the zero-duality gap (complementarity condition) and moves toward primal feasibility (see Section 4.3). The primal barrier method will keep improving a primal feasible solution and move toward dual feasibility and complementarity; and the primal-dual interior-point methods will keep improving a primal and dual feasible solution pair and move toward complementarity.

Primal barrier method

A direct approach is to use the barrier construction and solve the the problem

$$\begin{array}{rc} \text{minimize} & \mathbf{c}^T\mathbf{x} - \mu \sum_{j=1}^n \log x_j \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{array} \tag{5.10}$$

for a very small value of μ. In fact, if we desire to reduce the duality gap to ε it is only necessary to solve the problem for $\mu = \varepsilon/n$. Unfortunately, when μ is small, the problem (5.10) could be highly ill-conditioned in the sense that the necessary conditions are nearly singular. This makes it difficult to directly solve the problem for small μ.

An overall strategy, therefore, is to start with a moderately large μ (say $\mu = 100$) and solve that problem approximately. The corresponding solution is a point approximately on the primal central path, but it is likely to be quite distant from the point corresponding to the limit of $\mu \to 0$. However this solution point at $\mu = 100$ can be used as the starting point for the problem with a slightly smaller μ, for this point is likely to be close to the solution of the new problem. The value of μ might be reduced at each stage by a specific factor, giving $\mu_{k+1} = \gamma\mu_k$, where γ is a fixed positive parameter less than one and k is the stage count.

If the strategy is begun with a value μ_0, then at the k-th stage we have $\mu_k = \gamma^k \mu_0$. Hence to reduce μ_k/μ_0 to below ε, requires

$$k = \frac{\log \varepsilon}{\log \gamma}$$

stages.

Often a version of Newton's method for minimization is used to solve each of the problems. For the current strategy, Newton's method works on a problem (5.10) with fixed μ by moving from a given point $\mathbf{x} \in \overset{\circ}{\mathcal{F}}_p$ to a closer point $\mathbf{x}^+ \in \overset{\circ}{\mathcal{F}}_p$ by solving for $\mathbf{d}_x$, $\mathbf{d}_y$ and $\mathbf{d}_s$ from the linearized version of the central path equations of (5.7), as

$$\begin{array}{rcl} \mu\mathbf{X}^{-2}\mathbf{d}_x + \mathbf{d}_s & = & \mu\mathbf{X}^{-1}\mathbf{1} - \mathbf{c}, \\ \mathbf{A}\mathbf{d}_x & = & \mathbf{0}, \\ -\mathbf{A}^T\mathbf{d}_y - \mathbf{d}_s & = & \mathbf{0}. \end{array} \tag{5.11}$$

(Recall that $\mathbf{X}$ is the diagonal matrix whose diagonal entries are $\mathbf{x} > \mathbf{0}$.) The new point is then updated by taking a step in the direction of $\mathbf{d}_x$, as $\mathbf{x}^+ = \mathbf{x} + \mathbf{d}_x$.

Notice that if $\mathbf{x}\circ\mathbf{s} = \mu\mathbf{1}$ for some $\mathbf{s} = \mathbf{c}-\mathbf{A}^T\mathbf{y}$, then $\mathbf{d} \equiv (\mathbf{d}_x, \mathbf{d}_y, \mathbf{d}_s) = \mathbf{0}$ because the current point is already the central path solution for μ. If some component of $\mathbf{x}\circ\mathbf{s}$ is less than μ, then $\mathbf{d}$ will tend to increment the solution so as to increase that component. The converse will occur if $\mathbf{x}\circ\mathbf{s}$ is greater than μ.

This process may be repeated several times until a point close enough to the proper solution to the barrier problem for the given value of μ is obtained. That is, until the necessary and sufficient conditions (5.7) are (approximately) satisfied.

There are several details involved in a complete implementation and analysis of Newton's method. These items are discussed in later chapters of the text. However, the methods works well if either μ is moderately large, or if the algorithm is initiated at a point very close to the solution, exactly as needed for the barrier strategy discussed in this subsection.

To solve (5.11), premultiplying both sides by $\mathbf{X}^2$ we have

$$\mu\mathbf{d}_x + \mathbf{X}^2\mathbf{d}_s = \mu\mathbf{X}\mathbf{1} - \mathbf{X}^2\mathbf{c}.$$

Then, premultiplying by $\mathbf{A}$ and using $\mathbf{A}\mathbf{d}_x = \mathbf{0}$, we have

$$\mathbf{A}\mathbf{X}^2\mathbf{d}_s = \mu\mathbf{A}\mathbf{X}\mathbf{1} - \mathbf{A}\mathbf{X}^2\mathbf{c}.$$

Using $\mathbf{d}_s = -\mathbf{A}^T\mathbf{d}_y$ we have

$$(\mathbf{A}\mathbf{X}^2\mathbf{A}^T)\mathbf{d}_y = -\mu\mathbf{A}\mathbf{X}\mathbf{1} + \mathbf{A}\mathbf{X}^2\mathbf{c}.$$

Thus, $\mathbf{d}_y$ can be computed by solving the above linear system of equations, which, together with computing $\mathbf{d}_s$ and $\mathbf{d}_x$, amount to $O(nm^2+m^3)$ arithmetic operations for each Newton step.

5.6.1 Primal-dual path-following

Another strategy for solving LP is to follow the central path from a given initial primal-dual solution pair. Consider a linear program in the standard form (LP) and (LD). Assume that $\overset{\circ}{\mathcal{F}}\neq \emptyset$; that is, both

$$\overset{\circ}{\mathcal{F}}_p= \{\mathbf{x}: \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} > \mathbf{0}\} \neq \emptyset$$

and

$$\overset{\circ}{\mathcal{F}}_d= \{(\mathbf{y}, \mathbf{s}): \ \mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y} > \mathbf{0}\} \neq \emptyset,$$

and denote by z^* the optimal objective value.

The central path can be expressed as

$$\mathcal{C} = \left\{ (\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \overset{\circ}{\mathcal{F}}\colon\ \mathbf{x} \circ \mathbf{s} = \frac{\mathbf{x}^T\mathbf{s}}{n}\mathbf{1} \right\}$$

in the primal-dual form. On the path we have $\mathbf{x} \circ \mathbf{s} = \mu\mathbf{1}$ and hence $\mathbf{s}^T\mathbf{x} = n\mu$. A *neighborhood* of the central path $\mathcal{C}$ is of the form

$$\mathcal{N}(\eta) = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \overset{\circ}{\mathcal{F}}\colon |\mathbf{s} \circ \mathbf{x} - \mu\mathbf{1}| < \eta\mu, \text{where } \mu = \mathbf{s}^T\mathbf{x}/n\} \tag{5.12}$$

for some $\eta \in (0, 1)$, say $\eta = 1/4$. This can be thought of as a tube whose center is the central path.

The idea of the path-following method is to move within a tubular neighborhood of the central path toward the solution point. A suitable initial point $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{N}(\eta)$ can be found by solving the barrier problem for some fixed μ_0 or from an initialization phase proposed later. After that, step by step moves are made, alternating between a predictor step and a corrector step. After each pair of steps, the point achieved is again in the fixed given neighborhood of the central path, but closer to the LP solution set.

The predictor step is designed to move essentially parallel to the true central path. The step $\mathbf{d} \equiv (\mathbf{d}_x, \mathbf{d}_y, \mathbf{d}_s)$ is determined from the linearized version of the primal-dual central path equations of (5.9), as

$$\begin{array}{rcl} \mathbf{s} \circ \mathbf{d}_x + \mathbf{x} \circ \mathbf{d}_s & = & \gamma\mu\mathbf{1} - \mathbf{x} \circ \mathbf{s}, \\ \mathbf{A}\mathbf{d}_x & = & \mathbf{0}, \\ -\mathbf{A}^T\mathbf{d}_y - \mathbf{d}_s & = & \mathbf{0}, \end{array} \tag{5.13}$$

where one selects $\gamma = 0$. (To show the dependence of $\mathbf{d}$ on the current pair $(\mathbf{x}, \mathbf{s})$ and the parameter γ, we write $\mathbf{d} = \mathbf{d}(\mathbf{x}, \mathbf{s}, \gamma)$.)

The new point is then found by taking a step in the direction of $\mathbf{d}$, as $(\mathbf{x}^+, \mathbf{y}^+, \mathbf{s}^+) = (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\mathbf{d}_x, \mathbf{d}_y, \mathbf{d}_s)$, where α is called the step-size. Note that $\mathbf{d}_x^T\mathbf{d}_s = -\mathbf{d}_x^T\mathbf{A}^T\mathbf{d}_y = 0$ here. Then

$$(\mathbf{x}^+)^T\mathbf{s}^+ = (\mathbf{x} + \alpha\mathbf{d}_x)^T(\mathbf{s} + \alpha\mathbf{d}_s) = \mathbf{x}^T\mathbf{s} + \alpha(\mathbf{d}_x^T\mathbf{s} + \mathbf{x}^T\mathbf{d}_s) = (1 - \alpha)\mathbf{x}^T\mathbf{s}.$$

Thus, the predictor step will then reduce the duality gap by a factor $1 - \alpha$, because the new direction $\mathbf{d}$ will tend to reduce it. The maximum possible step-size α in that direction is made in that parallel direction without going outside of the neighborhood $\mathcal{N}(2\eta)$.

The corrector step essentially moves perpendicular to the central path in order to get closer to it. This step moves the solution back to within the neighborhood $\mathcal{N}(\eta)$, and the step is determined by selecting $\gamma = 1$ as in

(5.13) with $\mu = \mathbf{x}^T\mathbf{s}/n$. Notice that if $\mathbf{x} \circ \mathbf{s} = \mu\mathbf{1}$, then $\mathbf{d} = \mathbf{0}$ because the current point is already the central path solution.

This corrector step is identical to one step of the barrier method. Note, however, that the predictor–corrector method requires only one sequence of steps each consisting of a single predictor and corrector. This contrasts with the barrier method which requires a complete sequence for each μ to get back to the central path, and then an outer sequence to reduce the μ's.

One can prove that for any $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\eta)$ with $\mu = \mathbf{x}^T\mathbf{s}/n$, the step-size satisfied

$$\alpha \geq \frac{1}{2\sqrt{n}}.$$

Thus, the operation complexity of the method is identical to that of the barrier method with the overall operation complexity $O(n^{.5}(nm^2 + m^3)\log(1/\varepsilon))$ to achive $\mu/\mu_0 \leq \varepsilon$ where $n\mu_0$ is the initial duality gap. Moreover, one can prove that the step-size $\alpha \to 1$ as $\mathbf{x}^T\mathbf{s} \to 0$, that is, the duality reduction speed is accelerated as the gap becomes smaller.

5.6.2 Primal-dual potential function

In this method a *primal-dual potential function* is used to measure the solution's progress. The potential is reduced at each iteration. There is no restriction on either neighborhood or step-size during the iterative process as long as the potential is reduced. The greater the reduction of the potential function, the faster the convergence of the algorithm. Thus, from a practical point of view, potential-reduction algorithms may have an advantage over *path-following algorithms* where iterates are confined to lie in certain neighborhoods of the central path.

For $\mathbf{x} \in \overset{\circ}{\mathcal{F}}_p$ and $(\mathbf{y}, \mathbf{s}) \in \overset{\circ}{\mathcal{F}}_d$ the primal–dual potential function is defined by

$$\psi_{n+\rho}(\mathbf{x}, \mathbf{s}) \equiv (n + \rho)\log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n} \log(x_j s_j), \tag{5.14}$$

where $\rho \geq 0$.

From the arithmetic and geometric mean inequality (also see Exercise 5.9) we can derive that

$$n\log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n} \log(x_j s_j) \geq n \log n.$$

Then

$$\psi_{n+\rho}(\mathbf{x},\mathbf{s}) = \rho\log(\mathbf{x}^T\mathbf{s}) + n\log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n}\log(x_j s_j) \geq \rho\log(\mathbf{x}^T\mathbf{s}) + n\log n. \tag{5.15}$$

Thus, for $\rho > 0$, $\psi_{n+\rho}(\mathbf{x},\mathbf{s}) \to -\infty$ implies that $\mathbf{x}^T\mathbf{s} \to 0$. More precisely, we have from (5.15)

$$\mathbf{x}^T\mathbf{s} \leq \exp\left(\frac{\psi_{n+\rho}(\mathbf{x},\mathbf{s}) - n\log n}{\rho}\right).$$

Hence the primal–dual potential function gives an explicit bound on the magnitude of the duality gap.

The objective of this method is to drive the potential function down toward minus infinity. The method of reduction is a version of Newton's method (5.13). In this case we select $\gamma = n/(n+\rho)$. Notice that is a combination of a predictor and corrector choice. The predictor uses $\gamma = 0$ and the corrector uses $\gamma = 1$. The primal–dual potential method uses something in between. This seems logical, for the predictor moves parallel to the central path toward a lower duality gap, and the corrector moves perpendicular to get close to the central path. This new method does both at once. Of course, this intuitive notion must be made precise.

For $\rho \geq \sqrt{n}$, there is in fact a guaranteed decrease in the potential function by a fixed amount δ (see Exercises 5.11 and 5.12). Specifically,

$$\psi_{n+\rho}(\mathbf{x}^+,\mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x},\mathbf{s}) \leq -\delta \tag{5.16}$$

for a constant $\delta \geq 0.2$. This result provides a theoretical bound on the number of required iterations and the bound is competitive with other methods. However, a faster algorithm may be achieved by conducting a line search along direction $\mathbf{d}$ to achieve the greatest reduction in the primal-dual potential function at each iteration.

We outline the algorithm here:

Algorithm 5.1 *Given* $(\mathbf{x}^0,\mathbf{y}^0,\mathbf{s}^0) \in \overset{\circ}{\mathcal{F}}$ *with* $\psi_{n+\rho}(\mathbf{x}^0,\mathbf{s}^0) \leq \rho\log((\mathbf{s}^0)^T\mathbf{x}^0) + n\log n + O(\sqrt{n}\log n)$. *Set* $\rho \geq \sqrt{n}$ *and* $k = 0$.

While $\frac{(\mathbf{s}^k)^T\mathbf{x}^k}{(\mathbf{s}^0)^T\mathbf{x}^0} \geq \varepsilon$ **do**

1. *Set* $(\mathbf{x},\mathbf{s}) = (\mathbf{x}^k,\mathbf{s}^k)$ *and* $\gamma = n/(n+\rho)$ *and compute* $(\mathbf{d}_x,\mathbf{d}_y,\mathbf{d}_s)$ *from (5.13).*

2. *Let* $\mathbf{x}^{k+1} = \mathbf{x}^k + \bar{\alpha}\mathbf{d}_x$, $\mathbf{y}^{k+1} = \mathbf{y}^k + \bar{\alpha}\mathbf{d}_y$, *and* $\mathbf{s}^{k+1} = \mathbf{s}^k + \bar{\alpha}\mathbf{d}_s$ *where*

$$\bar{\alpha} = \arg\min_{\alpha\geq 0}\psi_{n+\rho}(\mathbf{x}^k + \alpha\mathbf{d}_x, \mathbf{s}^k + \alpha\mathbf{d}_s).$$

3. Let $k = k + 1$ and return to Step 1.

Theorem 5.4 *Algorithm 5.1 terminates in at most $O(\rho \log(n/\varepsilon))$ iterations with*

$$\frac{(\mathbf{s}^k)^T\mathbf{x}^k}{(\mathbf{s}^0)^T\mathbf{x}^0} \le \varepsilon.$$

Proof. Note that after k iterations, we have from (5.16)

$$\psi_{n+\rho}(\mathbf{x}^k, \mathbf{s}^k) \le \psi_{n+\rho}(\mathbf{x}^0, \mathbf{s}^0) - k \cdot \delta \le \rho \log((\mathbf{s}^0)^T\mathbf{x}^0) + n \log n + O(\sqrt{n} \log n) - k \cdot \delta.$$

Thus, from the inequality (5.15),

$$\rho \log((\mathbf{s}^k)^T\mathbf{x}^k) + n \log n \le \rho \log((\mathbf{s}^0)^T\mathbf{x}^0) + n \log n + O(\sqrt{n} \log n) - k \cdot \delta,$$

or

$$\rho(\log((\mathbf{s}^k)^T\mathbf{x}^k) - \log((\mathbf{s}^0)^T\mathbf{x}^0)) \le -k \cdot \delta + O(\sqrt{n} \log n).$$

Therefore, as soon as $k \ge O(\rho \log(n/\varepsilon))$, we must have

$$\rho(\log((\mathbf{s}^k)^T\mathbf{x}^k) - \log((\mathbf{s}^0)^T\mathbf{x}^0)) \le -\rho \log(1/\varepsilon),$$

or

$$\frac{(\mathbf{s}^k)^T\mathbf{x}^k}{(\mathbf{s}^0)^T\mathbf{x}^0} \le \varepsilon.$$

□

Theorem 5.4 holds for any $\rho \ge \sqrt{n}$. Thus, by choosing $\rho = \sqrt{n}$, the iteration complexity bound becomes $O(\sqrt{n} \log(n/\varepsilon))$.

Computational complexity cost of each iteration

The computation of each iteration basically invokes the solving (5.13) for $\mathbf{d}$. Note that the first equation of (5.13) can be written as

$$\mathbf{S}\mathbf{d}_x + \mathbf{X}\mathbf{d}_s = \gamma\mu\mathbf{1} - \mathbf{X}\mathbf{S}\mathbf{1}$$

where $\mathbf{X}$ and $\mathbf{S}$ are two diagonal matrices whose diagonal entries are $\mathbf{x} > \mathbf{0}$ and $\mathbf{s} > \mathbf{0}$, respectively. Premultiplying both sides by $\mathbf{S}^{-1}$ we have

$$\mathbf{d}_x + \mathbf{S}^{-1}\mathbf{X}\mathbf{d}_s = \gamma\mu\mathbf{S}^{-1}\mathbf{1} - \mathbf{x}.$$

Then, premultiplying by $\mathbf{A}$ and using $\mathbf{A}\mathbf{d}_x = \mathbf{0}$, we have

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{d}_s = \gamma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1} - \mathbf{A}\mathbf{x} = \gamma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1} - \mathbf{b}.$$

Using $\mathbf{d}_s = -\mathbf{A}^T\mathbf{d}_y$ we have

$$(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^T)\mathbf{d}_y = \mathbf{b} - \gamma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}.$$

Thus, the primary computational cost of each iteration of the interior-point algorithm discussed in this section is to form and invert the normal matrix $\mathbf{A}\mathbf{X}\mathbf{S}^{-1}\mathbf{A}^T$, which typically requires $O(nm^2+m^3)$ arithmetic operations. However, an approximation of this matrix can be updated and inverted using far fewer arithmetic operations. In fact, using a rank-one technique (see Chapter 10) to update the approximation inverse of the normal matrix during the iterative progress, one can reduce the average number of arithmetic operations per iteration to $O(\sqrt{n}m^2)$. Thus, if relative tolerance ε is viewed as a variable, we have the following total arithmetic operation complexity bound to solve an linear program:

Corollary 5.5 *Let $\rho = \sqrt{n}$. Then, Algorithm 5.1 terminates in at most $O(nm^2\log(n/\varepsilon))$ arithmetic operations with*

$$\frac{(\mathbf{x}^k)^T\mathbf{s}^k}{(\mathbf{x}^0)^T\mathbf{s}^0} \le \varepsilon.$$

5.7 Termination and Initialization*

There are several remaining important issues concerning interior-point algorithms for LP. The first issue involves termination. Unlike the simplex method for linear programming which terminates with an exact solution, interior-point algorithms are continuous optimization algorithms that generate an infinite solution sequence converging to the optimal solution set. If the data of an LP instance are integral or rational, an argument is made that, after the worst-case time bound, an exact solution can be rounded from the latest approximate solution. Several questions arise. First, under the real number computation model (that is, the LP data consists of real numbers), how can we terminate at an exact solution? Second, regardless of the data's status, is there a practical test, which can be computed cost-effectively during the iterative process, to identify an exact solution so that the algorithm can be terminated before the worse-case time bound? Here, by exact solution we mean one that could be found using exact arithmetic, such as the solution of a system of linear equations, which can be computed in a number of arithmetic operations bounded by a polynomial in n.

The second issue involves initialization. Almost all interior-point algorithms solve the LP problem under the regularity assumption that $\overset{\circ}{\mathcal{F}} \neq \emptyset$. What is to be done if this is not true?

A related issue is that interior-point algorithms have to start at a strictly feasible point near the central path. One way is to explicitly bound the feasible region of the LP problem by a big M number. If the LP problem has integral data, this number can be set to 2^L in the worst case, where L is the length of the LP data in binary form. This is not workable for large problems. Moreover, if the LP problem has real data, no computable bound is known.

Termination

In the previously studied complexity bounds for interior-point algorithms, we have an ε which needs to be zero in order to obtain an exact optimal solution. Sometimes it is advantageous to employ an early termination or rounding method while ε is still moderately large. There are five basic approaches.

- A "purification" procedure finds a feasible corner whose objective value is at least as good as the current interior point. This can be accomplished in strongly polynomial time (that is, the complexity bound is a polynomial only in the dimensions m and n). One difficulty is that then may be many non-optimal vertices may be close to the optimal face, and the procedure might require many pivot steps for difficult problems.

- A second method seeks to identify an optimal basis. It has been shown that if the LP problem is nondegenerate, the unique optimal basis may be identified early. The procedure seems to work well for some LP problems but it has difficulty if the problem is degenerate. Unfortunately, most real LP problems are degenerate.

- The third approach is to slightly perturb the data such that the new LP problem is nondegenerate and its optimal basis remains one of the optimal bases of the original LP problem. There are questions about how and when to perturb the data during the iterative process, decisions which can significantly affect the success of the effort.

- The fourth approach is to guess the optimal face and find a feasible solution on that face. It consists of two phases: the first phase uses interior point algorithms to identify the complementarity partition (P^*, Z^*) (see Exercise 5.5), and the second phase adapts the simplex method to find an optimal primal (or dual) basic solution and one can use (P^*, Z^*) as a starting base for the second phase. This method is often called the cross-over method. It is guaranteed to work in finite time and is implemented in several popular LP software packages.

- The fifth approach is to guess the optimal face and project the current interior point onto the interior of the optimal face. This again uses interior point algorithms to identify the complementarity partition (P^*, Z^*), and then solves a least-squares problem for the projection; see Figure 5.5. The termination criterion is guaranteed to work in finite time.

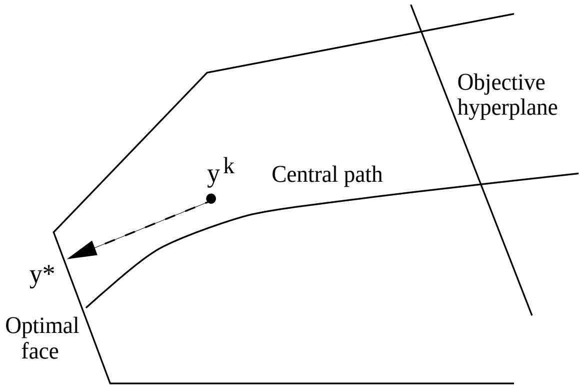

Figure 5.5: Illustration of the projection of an interior point onto the optimal face.

The fourth and fifth methods above are based on the fact that (as observed in practice and subsequently proved) many interior-point algorithms for linear programming generate solution sequences that converge to a strictly complementary solution or an interior solution on the optimal face; see Exercise (5.7).

Initialization

Most interior-point algorithms must be initiated at a strictly feasible point. The complexity of obtaining such an initial point is the same as that of solving the LP problem itself. More importantly, a complete LP algorithm should accomplish two tasks: 1) detect the infeasibility or unboundedness status of the LP problem, then 2) generate an optimal solution if the problem is neither infeasible nor unbounded.

Several approaches have been proposed to accomplish these goals:

- The primal and dual can be combined into a single linear feasibility problem, and then an LP algorithm can find a feasible point of the problem. Theoretically, this approach achieves the currently best iteration complexity bound, i.e., $O(\sqrt{n}\log(1/\varepsilon))$. Practically, a significant disadvantage of this approach is the doubled dimension of the system of equations that must be solved at each iteration.

- The big M method can be used by adding one or more artificial column(s) and/or row(s) and a huge penalty parameter M to force solutions to become feasible during the algorithm. Theoretically, this approach also achieves the best iteration bound. A major disadvantage of this approach is the numerical problems caused by the addition of coefficients of large magnitude.

- Phase I-then-Phase II methods are effective. They first try to find a feasible point (and possibly one for the dual problem), and then begin to look for an optimal solution if the problem is feasible and bounded. Theoretically, this approach also achieves the best iteration complexity bound. A major disadvantage of this approach is that the two (or three) related LP problems must be solved sequentially.

- A variation is the combined Phase I-Phase II method. This approaches feasibility and optimality simultaneously. To our knowledge, the currently best iteration complexity bound of this approach is $O(n\log(1/\varepsilon))$, as compared to $O(\sqrt{n}\log(1/\varepsilon))$ of the three above. Other disadvantages of the method include the assumption of nonempty interior and the need of an objective lower bound.

There is a homogeneous and self-dual (HSD) LP algorithm to overcome the difficulties mentioned above. The algorithm, while also achieves theoretically best $O(\sqrt{n}\log(1/\varepsilon))$ complexity bound, is often used in LP software packages and possesses the following features:

- It solves a linear programming problem without any regularity assumption concerning the existence of optimal, feasible, or interior feasible solutions.

- It can start at $\mathbf{x} = \mathbf{e}$, $\mathbf{y} = \mathbf{0}$ and $\mathbf{s} = \mathbf{e}$, feasible or infeasible, on the central ray of the positive orthant (cone), and it does not require a big M penalty parameter or lower bound.

- Each iteration solves a system of linear equations whose dimension is almost the same as that solved in the standard (primal-dual) interior-point algorithms.

- If the LP problem has a solution, the algorithm generates a sequence that approaches feasibility and optimality simultaneously; if the problem is infeasible or unbounded, the algorithm will produce an infeasibility certificate for at least one of the primal and dual problems; see Exercise 5.4.

5.8 Notes

Computation and complexity models were developed by a number of scientists; see, e.g., Cook [14], Hartmanis and Stearns [30] and Papadimitriou and Steiglitz [49] for the bit complaxity models and Blum et al. [12] for the real number arithmetic model.

Most materials of subsection 5.1 are based on a teaching note of Cottle on *Linear Programming* tatught at Stanford. Practical performances of the simplex method can be seen in Bixby [10].

The ellipsoid method was developed by Khachiyan [34]; more developments of the ellipsoid method can be found in Bland and Todd [11] and Goldfarb and Todd [25].

The “analytic center” for a convex polyhedron given by linear inequalities was introduced by Huard [32], and later by Sonnevend [53]. The *barrier function*, was introduced by Frisch [23].

McLinden [40] earlier, then Bayer and Lagarias [6, 7], Megiddo [41], and Sonnevend [53], analyzed the central path for linear programming and convex optimization; also see Fiacco and McCormick [21].

The path-following algorithms were first developed by Renegar [50]. A primal barrier or path-following algorithm is independently analyzed by Gonzaga [28]. Both Gonzaga [28] and Vaidya [59] developed a rank-one updating technique in solving the Newton equation of each iteration, and proved that each iteration uses $O(n^{2.5})$ arithmetic operations on average. Kojima, Mizuno and Yoshise [36] and Monteiro and Adler [44] developed a symmetric primal-dual path-following algorithm with the same iteration and arithmetic operation bounds.

The predictor-corrector algorithms were developed by Mizuno et al. [43]. A more practical predictor-corrector algorithm was proposed by Mehrotra [42] (also see Lustig et al. [39] and Zhang and Zhang [66]). His technique has been used in almost all of the LP interior-point implementations.

A primal potential reduction algorithm was initially proposed by Karmarkar [33]. The primal-dual potential function was proposed by Tanabe [54] and Todd and Ye [56]. The primal-dual potential reduction algorithm was developed by Ye [64], Freund [22], Kojima, Mizuno and Yoshise [37], Goldfarb and Xiao [26], Gonzaga and Todd [29], Todd [55], Tuncel4 [57], Tunccel [57], etc.

The homogeneous and self-dual embedding method can be found Ye et al. [65], Luo et al. [38], Andersen2 [3], and many others.

In addition to those mentioned earlier, there are several comprehensive books which cover interior-point linear programming algorithms. They are Bazaraa, Jarvis and Sherali [5], Bertsekas [8], Bertsimas and Tsitsiklis [9], Cottle [15], Cottle, Pang and Stone [16], Dantzig and Thapa [18, 19], Fang and Puthenpura [20], den Hertog [31], Murty [45], Nash and Sofer [46], Roos et al. [51], Saigal [52], Vanderbei [61], Vavasis [62], Wright [63], etc.

5.9 Exercises

5.1 *Prove the volume reduction rate in Theorem 5.1 for the ellipsoid method.*

5.2 *Develop a cutting plane method, based on the ellipsoid method, to find a point satisfying convex inequalities*

$$f_i(\mathbf{x}) \le 0,\ i = 1, ..., m,\ |\mathbf{x}|^2 \le R^2,$$

where f_i's are convex functions of $\mathbf{x}$ in C^1.

5.3 *Consider linear program (5.5) and assume that $\overset{\circ}{\mathcal{F}} = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0}\}$ is nonempty and its optimal solution set is bounded. Then, the dual of the problem has a nonempty interior.*

5.4 *(Farkas' lemma) Exactly one of the feasible set $\{\mathbf{x} :\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \ge \mathbf{0}\}$ and the feasible set $\{\mathbf{y} :\ \mathbf{y}^T\mathbf{A} \le \mathbf{0},\ \mathbf{y}^T\mathbf{b} = 1\}$ is nonempty. A vector $\mathbf{y}$ in the latter set is called an infeasibility certificate for the former.*

5.5 *(Strict complementarity) Given any linear program in the standard form, the primal optimal face is*

$$\Omega_p = \{\mathbf{x}_{P^*} : \mathbf{A}_{P^*}\mathbf{x}_{P^*} = \mathbf{b},\ \mathbf{x}_{P^*} \ge \mathbf{0}\},$$

and the dual optimal face is

$$\Omega_d = \{(\mathbf{y}, \mathbf{s}_{Z^*}) : \mathbf{A}_{P^*}^T\mathbf{y} = \mathbf{c}_{P^*},\quad \mathbf{s}_{Z^*} = \mathbf{c}_{Z^*} - \mathbf{A}_{Z^*}^T\mathbf{y} \ge \mathbf{0}\},$$

where (P^*, Z^*) is a partition of $\{1, 2, ..., n\}$. Prove that the partition is unique, and each face has an interior, that is,

$$\overset{\circ}{\Omega}_p = \{\mathbf{x}_{P^*} : \mathbf{A}_{P^*}\mathbf{x}_{P^*} = \mathbf{b},\ \mathbf{x}_{P^*} > \mathbf{0}\},$$

and

$$\overset{\circ}{\Omega}_d = \{(\mathbf{y}, \mathbf{s}_{Z^*}) : \mathbf{A}_{P^*}^T\mathbf{y} = \mathbf{c}_{P^*},\quad \mathbf{s}_{Z^*} = \mathbf{c}_{Z^*} - \mathbf{A}_{Z^*}^T\mathbf{y} > \mathbf{0}\},$$

are both nonempty.

5.6 *(Central path theorem) Let* $(\mathbf{x}(\mu), \mathbf{y}(\mu), \mathbf{s}(\mu))$ *be the central path of (5.9). Then prove*

i) *The central path point* $(\mathbf{x}(\mu), \mathbf{s}(\mu))$ *is bounded for* $0 < \mu \le \mu^0$ *and any given* $0 < \mu^0 < \infty$.

ii) *For* $0 < \mu' < \mu$,

$$\mathbf{c}^T\mathbf{x}(\mu') \le \mathbf{c}^T\mathbf{x}(\mu) \quad \textit{and} \quad \mathbf{b}^T\mathbf{y}(\mu') \ge \mathbf{b}^T\mathbf{y}(\mu).$$

Furthermore, if $\mathbf{x}(\mu') \ne \mathbf{x}(\mu)$ *and* $\mathbf{y}(\mu') \ne \mathbf{y}(\mu)$,

$$\mathbf{c}^T\mathbf{x}(\mu') < \mathbf{c}^T\mathbf{x}(\mu) \quad \textit{and} \quad \mathbf{b}^T\mathbf{y}(\mu') > \mathbf{b}^T\mathbf{y}(\mu).$$

iii) $(\mathbf{x}(\mu), \mathbf{s}(\mu))$ *converges to an optimal solution pair for (LP) and (LD). Moreover, the limit point* $\mathbf{x}(0)_{P^*}$ *is the analytic center on the primal optimal face, and the limit point* $\mathbf{s}(0)_{Z^*}$ *is the analytic center on the dual optimal face, where* (P^*, Z^*) *is the strict complementarity partition of the index set* $\{1, 2, ..., n\}$.

5.7 *Consider a primal-dual interior point* $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\eta)$ *where* $\eta < 1$. *Prove that there is a fixed quantity* $\delta > 0$ *such that*

$$x_j \ge \delta, \ \forall j \in P^*$$

and

$$s_j \ge \delta, \ \forall j \in Z^*,$$

where (P^*, Z^*) *is defined in Exercise 5.5.*

5.8 *(Potential level theorem) Define the potential level set*

$$\Psi(\delta) := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \overset{\circ}{\mathcal{F}}\colon \ \psi_{n+\rho}(\mathbf{x}, \mathbf{s}) \le \delta\}.$$

Prove

i)

$$\Psi(\delta^1) \subset \Psi(\delta^2) \quad \textit{if} \quad \delta^1 \le \delta^2.$$

ii) *For every* δ, $\Psi(\delta)$ *is bounded and its closure* $\hat{\Psi}(\delta)$ *has non-empty intersection with the LP solution set.*

5.9 *Given* $\mathbf{0} < \mathbf{x}, \mathbf{s} \in R^n$*, show that*

$$n \log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n} \log(x_j s_j) \geq n \log n$$

and

$$\mathbf{x}^T\mathbf{s} \leq \exp\left(\frac{\psi_{n+\rho}(\mathbf{x},\mathbf{s}) - n\log n}{\rho}\right).$$

5.10 *(Logarithmic approximation) If* $\mathbf{d} \in R^n$ *such that* $|\mathbf{d}|_\infty < 1$ *then*

$$\mathbf{e}^T\mathbf{d} \geq \sum_{i=1}^{n} \log(1+d_i) \geq \mathbf{e}^T\mathbf{d} - \frac{|\mathbf{d}|^2}{2(1-|\mathbf{d}|_\infty)} \ .$$

5.11 *Let the direction* $(\mathbf{d}_x, \mathbf{d}_y, \mathbf{d}_s)$ *be generated by system (5.13) with* $\gamma = n/(n+\rho)$ *and* $\mu = \mathbf{x}^T\mathbf{s}/n$*, and let step size*

$$\alpha = \frac{\theta\sqrt{\min(\mathbf{Xs})}}{\|(\mathbf{XS})^{-1/2}(\frac{\mathbf{x}^T\mathbf{s}}{(n+\rho)}\mathbf{1} - \mathbf{Xs})\|} \ , \tag{5.17}$$

where θ *is a positive constant less than* 1. *Let*

$$\mathbf{x}^+ = \mathbf{x} + \alpha\mathbf{d}_x, \quad \mathbf{y}^+ = \mathbf{y} + \alpha\mathbf{d}_y, \quad \text{and} \quad \mathbf{s}^+ = \mathbf{s} + \alpha\mathbf{d}_s.$$

Then, using exercise 5.10 and the concavity of logarithmic function show $(\mathbf{x}^+, \mathbf{y}^+, \mathbf{s}^+) \in \overset{\circ}{\mathcal{F}}$ *and*

$$\psi_{n+\rho}(\mathbf{x}^+, \mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x}, \mathbf{s})$$

$$\leq -\theta\sqrt{\min(\mathbf{Xs})}\|(\mathbf{XS})^{-1/2}(\mathbf{1} - \frac{(n+\rho)}{\mathbf{x}^T\mathbf{s}}\mathbf{Xs})\| + \frac{\theta^2}{2(1-\theta)} \ .$$

5.12 *Let* $\mathbf{v} = \mathbf{Xs}$ *in exercise 5.11. Then, prove*

$$\sqrt{\min(\mathbf{v})}\|\mathbf{V}^{-1/2}(\mathbf{1} - \frac{(n+\rho)}{\mathbf{1}^T\mathbf{v}}\mathbf{v})\| \geq \sqrt{3/4} \ ,$$

where $\mathbf{V}$ *is the diagonal matrix of* $\mathbf{v}$. *Thus, the two exercises imply*

$$\psi_{n+\rho}(\mathbf{x}^+, \mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x}, \mathbf{s}) \leq -\theta\sqrt{3/4} + \frac{\theta^2}{2(1-\theta)} = -\delta$$

for a constant δ. *One can verify that* $\delta > 0.2$ *when* $\theta = 0.4$.

Chapter 12

Penalty and Barrier Methods

The interior-point algorithms discussed for linear programming in Chapter 5 are, as we mentioned then, closely related to the barrier methods presented in this chapter. They can be naturally extended to solving nonlinear programming problems while maintain both theoretical and practical efficiency. Below we present two important such problems.

12.9 Convex quadratic programming

Quadratic programming (QP), specially convex quadratic programming, plays an important role in optimization. In one sense it is a continuous optimization problem and a fundamental subroutine frequently employed by general nonlinear programming, but it is also considered one of the challenging combinatorial problems because its optimality conditions are linear equalities and inequalities.

The barrier and interior-point LP algorithms discussed in Chapter 5 can be extended to solve convex quadratic programming (QP). We present one extension, the primal-dual potential reduction algorithm, in this section.

Since solving convex QP problems in the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{array} \tag{12.1}$$

where given matrix $\mathbf{Q} \in R^{n\times n}$ is positive semidefinite, $\mathbf{A} \in R^{n\times m}$, $\mathbf{c} \in R^n$ and $\mathbf{b} \in R^m$, reduces to finding $\mathbf{x} \in R^n$, $\mathbf{y} \in R^m$ and $\mathbf{s} \in R^n$ to satisfy the following optimality conditions:

$$\begin{array}{rcl} \mathbf{x}^T\mathbf{s} & = & 0 \\ \mathbf{M}\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} - \begin{pmatrix} \mathbf{s} \\ \mathbf{0} \end{pmatrix} & = & \mathbf{q} \\ \mathbf{x}, \mathbf{s} & \geq & \mathbf{0}, \end{array} \tag{12.2}$$

where

$$\mathbf{M} = \begin{pmatrix} \mathbf{Q} & -\mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \in R^{(n+m)\times(n+m)} \quad \text{and} \quad \mathbf{q} = \begin{pmatrix} -\mathbf{c} \\ \mathbf{b} \end{pmatrix} \in R^{n+m}.$$

The potential function for solving this problem has the same form as (5.14) over the interior of the feasible region, $\overset{\circ}{\mathcal{F}}$, of (12.2). Once we have an interior feasible point $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ with $\mu = \mathbf{x}^T\mathbf{s}/n$, we can generate a new iterate $(\mathbf{x}^+, \mathbf{y}^+, \mathbf{s}^+)$ by solving for $(\mathbf{d}_x, \mathbf{d}_y, \mathbf{d}_s)$ from the system of linear equations:

$$\begin{array}{rcl} \mathbf{S}\mathbf{d}_x + \mathbf{X}\mathbf{d}_s & = & \gamma\mu\mathbf{1} - \mathbf{X}\mathbf{s}, \\ \mathbf{M}\begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_y \end{pmatrix} - \begin{pmatrix} \mathbf{d}_s \\ \mathbf{0} \end{pmatrix} & = & \mathbf{0}, \end{array} \tag{12.3}$$

where $\mathbf{X}$ and $\mathbf{S}$ are two diagonal matrices whose diagonal entries are $\mathbf{x} > \mathbf{0}$ and $\mathbf{s} > \mathbf{0}$, respectively.

We again choose $\gamma = n/(n+\rho) < 1$. Then, assign $\mathbf{x}^+ = \mathbf{x} + \bar{\alpha}\mathbf{d}_x$, $\mathbf{y}^+ = \mathbf{y} + \bar{\alpha}\mathbf{d}_y$, and $\mathbf{s}^+ = \mathbf{s} + \bar{\alpha}\mathbf{d}_s$ where

$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi_{n+\rho}(\mathbf{x} + \alpha\mathbf{d}_x, \mathbf{s} + \alpha\mathbf{d}_s).$$

Since $\mathbf{Q}$ is positive semidefinite, we have

$$\mathbf{d}_x^T\mathbf{d}_s = (\mathbf{d}_x; \mathbf{d}_y)^T(\mathbf{d}_s; \mathbf{0}) = (\mathbf{d}_x; \mathbf{d}_y)^T\mathbf{M}(\mathbf{d}_x; \mathbf{d}_y) = \mathbf{d}_x^T\mathbf{Q}\mathbf{d}_x \geq 0$$

(where we recall that $\mathbf{d}_x^T\mathbf{d}_s = 0$ in the LP case).

Similar to the case of LP, one can prove that for $\rho \geq \sqrt{n}$

$$\psi_{n+\rho}(\mathbf{x}^+, \mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x}, \mathbf{s}) \leq -\theta\sqrt{3/4} + \frac{\theta^2}{2(1-\theta)} = -\delta$$

for a constant $\delta > 0.2$. This result will provide an iteration complexity bound that is identical to linear programming; see Chapter 5.

12.10 Semidefinite programming

Semidefinite Programming, hereafter SDP, is a natural extension of Linear programming. In LP, the variables form a vector which is required to be component-wise nonnegative, where in SDP they are components of a symmetric matrix and it is constrained to be positive semidefinite. Both of them may have linear equality constraints as well. Although SDP is known to be a convex optimization model, no efficient algorithm is known for solving it. One of the major result in optimization during the past decade is the discovery that interior-point algorithms for LP, discussed in Chapter 5, can be effectively adapted to solving SDP with both theoretical and practical efficiency. During the same period, the applicable domain of SDP has dramatically widened to combinatory optimization, statistical computation, robust optimization, Euclidean distance geometry, quantum computing, etc. SDP established its full popularity.

Let $\mathbf{C}$ and $\mathbf{A}_i$, $i = 1, 2, ..., m$, be given n-dimensional symmetric matrices and $\mathbf{b} \in R^m$, and let $\mathbf{X}$ be an unknown n-dimensional symmetric matrix. Then, the SDP model can be written as

$$\begin{array}{lll} (SDP) & \text{minimize} & \mathbf{C} \bullet \mathbf{X} \\ & \text{subject to} & \mathbf{A}_i \bullet \mathbf{X} = b_i, i = 1, 2, ..., m, \ \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{12.4}$$

where the $\bullet$ operation is the matrix inner product

$$\mathbf{A} \bullet \mathbf{B} \equiv \text{trace } \mathbf{A}^T\mathbf{B} = \sum_{i,j} a_{ij} b_{ij}.$$

The notation $\mathbf{X} \succeq \mathbf{0}$ means that $\mathbf{X}$ is positive semidefinite , and $\mathbf{X} \succ \mathbf{0}$ means that $\mathbf{X}$ is positive definite. If a matrix $\mathbf{X} \succ \mathbf{0}$ and satisfies all equalities in (SDP), it is called a (*primal*) strictly or interior feasible solution. .

Note that, in SDP, we minimize a linear function of a symmetric matrix constrained in the positive semidefinite matrix cone and subjected to linear equality constraints. This is in contrast to the nonnegative orthant cone required for LP.

Example 12.1 Binary quadratic optimization Consider a binary quadratic optimization

$$\begin{array}{ll} \text{minimize} & \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \text{subject to} & x_j = \{1, -1\}, \ \forall j, \end{array}$$

which is a difficult nonconvex optimization problem. The problem can be written as

$$\begin{array}{ll} \text{minimize} & \mathbf{Q} \bullet \mathbf{x}\mathbf{x}^T \\ \text{subject to} & \mathbf{I}_j \bullet \mathbf{x}\mathbf{x}^T = 1, \ \forall j, \end{array}$$

where $\mathbf{I}_j$ is the all zero matrix except 1 at the jth diagonal.

An SDP relaxation of the problem is

$$\begin{array}{ll} \text{minimize} & \mathbf{Q} \bullet \mathbf{X} \\ \text{subject to} & \mathbf{I}_j \bullet \mathbf{X} = 1,\ \forall j, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array}$$

It has been shown that an optimal SDP solution constitutes a good approximation to the original problem.

Example 12.2 Sensor localization Suppose we have n unknown points (sensors) $\mathbf{x}_j \in R^d$, $j = 1, ..., n$. For a pair of two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in N_e, we have a Euclidean distance measure $\hat{d}_{ij}$ between them. Then, the localization problem is to find $\mathbf{x}_j$, $j = 1, ..., n$, such that

$$|\mathbf{x}_i - \mathbf{x}_j|^2 = (\hat{d}_{ij})^2, \quad \forall (i,j) \in N_e,$$

subject to possible rotation and translation.

Let $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_n]$ be the $2 \times n$ matrix that needs to be determined. Then

$$|\mathbf{x}_i - \mathbf{x}_j|^2 = (\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{X}^T \mathbf{X} (\mathbf{e}_i - \mathbf{e}_j),$$

where $\mathbf{e}_i \in R^n$ is the vector with 1 at the ith position and zero everywhere else. Let $\mathbf{Y} = \mathbf{X}^T \mathbf{X}$. Then SDP relaxation of the localization problem is to find $\mathbf{Y}$ such that

$$\begin{array}{c} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \bullet \mathbf{Y} = (\hat{d}_{ij})^2, \quad \forall (i,j) \in N_e, \\ \mathbf{Y} \succeq \mathbf{0}. \end{array}$$

Example 12.3 Fast Markov chain design This problem is to design a symmetric stochcastic probability transition matrix $\mathbf{P}$ such that $p_{ij} = p_{ji} = 0$ for all (i,j) in a given non-edge set N_o and the spectral radius of matrix $\mathbf{P} - \mathbf{e}\mathbf{e}^T/n$ is minimized.

Let $\mathbf{I}_{ij} \in R^{n \times n}$ be the matrix with 1 at the (ij)th and (ji)th positions and zero everywhere else. Then, the problem can be framed as an SDP problem

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \mathbf{I}_{ij} \bullet \mathbf{P} = 0,\ \forall (i,j) \in N_o, \\ & \mathbf{I}_{ij} \bullet \mathbf{P} \geq 0,\ \forall (i,j) \notin N_o, \\ & \mathbf{e}_i \mathbf{e}^T \bullet \mathbf{P} = 1,\ \forall i = 1, \ldots, n, \\ & \lambda \mathbf{I} \succeq \mathbf{P} - \frac{1}{n}\mathbf{e}\mathbf{e}^T \succeq -\lambda \mathbf{I}. \end{array}$$

Duality theorems for SDP

The dual of (SDP) is

$$\begin{array}{lll} (SDD) & \text{maximize} & \mathbf{y}^T\mathbf{b} \\ & \text{subject to} & \sum_i^m y_i\mathbf{A}_i + \mathbf{S} = \mathbf{C}, \\ & & \mathbf{S} \succeq \mathbf{0}. \end{array} \tag{12.5}$$

Example 12.4 Given symmetric matrices $\mathbf{A}_i$, $i = 0, \ldots, m$, and consider designing $\mathbf{y} \in R^m$ to maximize the lowest eigenvalue of $\mathbf{A}_0 + \sum_i^m y_i\mathbf{A}_i$. The problem can be post as an SDP problem:

$$\begin{array}{ll} \text{maximize} & \lambda \\ \text{subject to} & \lambda\mathbf{I} - \sum_i^m y_i\mathbf{A}_i + \mathbf{S} = \mathbf{A}_0, \\ & \mathbf{S} \succeq \mathbf{0}. \end{array}$$

The weak duality theorem for SDP is identical to that of (LP) and (LD).

Lemma 12.1 *(Weak Duality Lemma in SDP) Let the feasible regions, $\mathcal{F}_p$ of (SDP) and $\mathcal{F}_d$ of (SDD), be both non-empty. Then,*

$$\mathbf{C} \bullet \mathbf{X} \geq \mathbf{b}^T\mathbf{y} \quad \textit{where} \quad \mathbf{X} \in \mathcal{F}_p, \ (\mathbf{y}, \mathbf{S}) \in \mathcal{F}_d.$$

But we need more to make the strong duality theorem hold.

Theorem 12.2 *(Strong Duality Theorem in SDP) Suppose $\mathcal{F}_p$ and $\mathcal{F}_d$ are non-empty and at least one of them has an interior. Then, $\mathbf{X}$ is optimal for (SDP) if and only if the following conditions hold:*

i) $\mathbf{X} \in \mathcal{F}_p$*;*

ii) *there is* $(\mathbf{y}, \mathbf{S}) \in \mathcal{F}_d$*;*

iii) $\mathbf{C} \bullet \mathbf{X} = \mathbf{b}^T\mathbf{y}$ *or* $\mathbf{X} \bullet \mathbf{S} = 0$*.*

If the non-empty interior condition of Theorem 12.2 does not hold, then the duality gap may not be zero at optimality.

Example 12.5 *The following SDP has a duality gap:*

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ \mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} 0 \\ 10 \end{pmatrix}.$$

The primal minimal objective value is 0 *and the dual maximal objective value is* -10 *so that the duality gap is* 10*.*

Interior-point algorithms for SDP

Let (SDP) and (SDD) both have interior point feasible solution. Then, the central path can be expressed as

$$\mathcal{C} = \left\{ (\mathbf{X}, \mathbf{y}, \mathbf{S}) \in \overset{\circ}{\mathcal{F}}\colon\ \mathbf{XS} = \mu\mathbf{I},\ 0 < \mu < \infty \right\}.$$

The primal-dual potential function for SDP is

$$\psi_{n+\rho}(\mathbf{X}, \mathbf{S}) = (n + \rho)\log(X \bullet S) - \log(\det(X) \cdot \det(S))$$

where $\rho \geq 0$. Note that if $\mathbf{X}$ and $\mathbf{S}$ are diagonal matrices, these definitions reduce to those for LP.

Once we have an interior feasible point $(\mathbf{X}, \mathbf{y}, \mathbf{S})$, we can generate a new iterate $(\mathbf{X}^+, \mathbf{y}^+, \mathbf{S}^+)$ by solving for $(\mathbf{D}_x, \mathbf{d}_y, \mathbf{D}_s)$ from a system of linear equations

$$\begin{array}{rcl} \mathbf{D}^{-1}\mathbf{D}_x\mathbf{D}^{-1} + \mathbf{D}_s & = & \gamma\mu\mathbf{X}^{-1} - \mathbf{S}, \\ \mathbf{A}_i \bullet \mathbf{D}_x & = & 0,\ \forall i, \\ -\sum_i^m (\mathbf{d}_y)_i \mathbf{A}_i - \mathbf{D}_s & = & \mathbf{0}, \end{array} \tag{12.6}$$

where the (scaling) matrix

$$\mathbf{D} = \mathbf{X}^{.5}(\mathbf{X}^{.5}\mathbf{S}\mathbf{X}^{.5})^{-.5}\mathbf{X}^{.5}$$

and $\mu = \mathbf{X} \bullet \mathbf{S}/n$. Then assign $\mathbf{X}^+ = \mathbf{X} + \bar{\alpha}\mathbf{D}_x$, $\mathbf{y}^+ = \mathbf{y} + \bar{\alpha}\mathbf{d}_y$, and $\mathbf{S}^+ = \mathbf{s} + \bar{\alpha}\mathbf{D}_s$ where

$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi_{n+\rho}(\mathbf{X} + \alpha\mathbf{D}_x, \mathbf{S} + \alpha\mathbf{D}_s).$$

Furthermore, we can show

$$\psi_{n+\rho}(\mathbf{X}^+, \mathbf{S}^+) - \psi_{n+\rho}(\mathbf{X}, \mathbf{S}) \leq -\delta$$

for a constant $\delta > 0.2$.

This will provide an iteration complexity bound that is identical to linear programming; see Chapter 5.

12.11 Notes

Many researchers have applied interior-point algorithms to convex QP problems. These algorithms can be divided into three groups: the primal scaling algorithm, the dual scaling algorithm, and the primal-dual scaling algorithm. Relations among these algorithms can be seen in Anstreicher, den Hertog, Roos and Terlaky [31, 4].

There have been several remarkable applications of SDP; see, e.g., Goemans and Williamson [24] and Vandenberghe and Boyd [13, 60].

The SDP example with a duality gap was constructed by Freund.

The primal potential reduction algorithm for positive semi-definite programming is due to Alizadeh [2, 1] and to Nesterov and Nemirovskii [47]. The primal-dual SDP algorithm described here is due to Nesterov and Todd [48].

12.12 Exercises

12.1 *(Farkas' lemma in SDP) Let* $\mathbf{A}_i$, $i = 1, ..., m$, *have rank* m *(i.e.,* $\sum_i^m y_i \mathbf{A}_i = \mathbf{0}$ *implies* $\mathbf{y} = \mathbf{0}$*). Then, there exists a symmetric matrix* $\mathbf{X} \succ \mathbf{0}$ *with*

$$\mathbf{A}_i \bullet \mathbf{X} = b_i, \quad i = 1, ..., m,$$

if and only if $\sum_i^m y_i \mathbf{A}_i \preceq \mathbf{0}$ *and* $\sum_i^m y_i \mathbf{A}_i \neq \mathbf{0}$ *implies* $\mathbf{b}^T \mathbf{y} < 0$.

12.2 *Let* $\mathbf{X}$ *and* $\mathbf{S}$ *be both positive definite. Then prove*

$$n \log(\mathbf{X} \bullet \mathbf{S}) - \log(\det(\mathbf{X}) \cdot \det(\mathbf{S})) \geq n \log n.$$

12.3 *Consider SDP and the potential level set*

$$\Psi(\delta) := \{(\mathbf{X}, \mathbf{y}, \mathbf{S}) \in \overset{\circ}{\mathcal{F}}: \ \psi_{n+\rho}(\mathbf{X}, \mathbf{S}) \leq \delta\}.$$

Prove that

$$\Psi(\delta^1) \subset \Psi(\delta^2) \quad \textit{if} \quad \delta^1 \leq \delta^2,$$

and for every δ $\Psi(\delta)$ *is bounded and its closure* $\hat{\Psi}(\delta)$ *has non-empty intersection with the SDP solution set.*

12.4 *Let both (SDP) and (SDD) have interior feasible points. Then for any* $0 < \mu < \infty$, *the central path point* $(\mathbf{X}(\mu), \mathbf{y}(\mu), \mathbf{S}(\mu))$ *exists and is unique. Moreover,*

i) *the central path point* $(\mathbf{X}(\mu), \mathbf{S}(\mu))$ *is bounded where* $0 < \mu \leq \mu^0$ *for any given* $0 < \mu^0 < \infty$.

ii) *For* $0 < \mu' < \mu$,

$$\mathbf{C} \bullet \mathbf{X}(\mu') < \mathbf{C} \bullet \mathbf{X}(\mu) \quad \textit{and} \quad \mathbf{b}^T \mathbf{y}(\mu') > \mathbf{b}^T \mathbf{y}(\mu)$$

if $\mathbf{X}(\mu) \neq \mathbf{X}(\mu')$ *and* $\mathbf{y}(\mu) \neq \mathbf{y}(mu')$.

iii) $(\mathbf{X}(\mu), \mathbf{S}(\mu))$ *converges to an optimal solution pair for (SDP) and (SDD), and the rank of the limit of* $\mathbf{X}(\mu)$ *is maximal among all optimal solutions of (SDP) and the rank of the limit* $\mathbf{S}(\mu)$ *is maximal among all optimal solutions of (SDD).*

Bibliography

[1] F. Alizadeh. *Combinatorial optimization with interior point methods and semi–definite matrices.* Ph.D. Thesis, University of Minnesota, Minneapolis, MN, 1991.

[2] F. Alizadeh. Optimization over the positive semi-definite cone: Interior–point methods and combinatorial applications. In P. M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 1–25. North Holland, Amsterdam, The Netherlands, 1992.

[3] E. D. Andersen and Y. Ye. On a homogeneous algorithm for the monotone complementarity problem. *Math. Programming*, 84:375-400, 1999.

[4] K. M. Anstreicher, D. den Hertog, C. Roos, and T. Terlaky. A long step barrier method for convex quadratic programming. *Algorithmica*, 10:365–382, 1993.

[5] M. S. Bazaraa, J. J. Jarvis, and H. F. Sherali. *Linear Programming and Network Flows*, chapter 8.4 : Karmarkar's projective algorithm, pages 380–394, chapter 8.5: Analysis of Karmarkar's algorithm, pages 394–418. John Wiley & Sons, New York, second edition, 1990.

[6] D. A. Bayer and J. C. Lagarias. The nonlinear geometry of linear programming, Part I: Affine and projective scaling trajectories. *Transactions of the American Mathematical Society*, 314(2):499–526, 1989.

[7] D. A. Bayer and J. C. Lagarias. The nonlinear geometry of linear programming, Part II: Legendre transform coordinates. *Transactions of the American Mathematical Society*, 314(2):527–581, 1989.

[8] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1995.

[9] D.M. Bertsimas and J.N. Tsitsiklis. *Linear Optimization.* Athena Scientific, Belmont, Massachusetts, 1997.

[10] R. E. Bixby. Progress in linear programming. *ORSA J. on Comput.*, 6(1):15–22, 1994.

[11] R. G. Bland, D. Goldfarb and M. J. Todd. The ellipsoidal method: a survey. *Operations Research*, 29:1039–1091, 1981.

[12] L. Blum, F. Cucker, M. Shub, and S. Smale, Complexity and Real Computation (Springer-Verlag, 1996).

[13] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Science.* SIAM Publications. SIAM, Philadelphia, 1994.

[14] S. A. Cook. The complexity of theorem-proving procedures. *Proc. 3rd ACM Symposium on the Theory of Computing*, 151–158, 1971.

[15] R. W. Cottle. *Linear Programming.* Lecture Notes for MS&E 310, Stanford University, Stanford, 2002.

[16] R. Cottle, J. S. Pang, and R. E. Stone. *The Linear Complementarity Problem*, chapter 5.9 : Interior–point methods, pages 461–475. Academic Press, Boston, 1992.

[17] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey, 1963.

[18] G.B. Dantzig and M.N. Thapa. *Linear Programming 1: Introduction.* Springer-Verlag, New York, 1997.

[19] G.B. Dantzig and M.N. Thapa. *Linear Programming 2: Theory and Extensions.* Springer-Verlag, New York, 2003.

[20] S. C. Fang and S. Puthenpura. *Linear Optimization and Extensions.* Prentice–Hall, Englewood Cliffs, NJ, 1994.

[21] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques.* John Wiley & Sons, New York, 1968. Reprint : Volume 4 of *SIAM Classics in Applied Mathematics*, SIAM Publications, Philadelphia, PA, 1990.

[22] R. M. Freund. Polynomial–time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Math. Programming*, 51:203–222, 1991.

[23] K. R. Frisch. The logarithmic potential method for convex programming. Unpublished manuscript, Institute of Economics, University of Oslo, Oslo, Norway, 1955.

[24] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[25] D. Goldfarb and M. J. Todd. Linear Programming. In G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors, *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages 141–170. North Holland, Amsterdam, The Netherlands, 1989.

[26] D. Goldfarb and D. Xiao. A primal projective interior point method for linear programming. *Math. Programming*, 51:17–43, 1991.

[27] A. J. Goldman and A. W. Tucker. Polyhedral convex cones. In H. W. Kuhn and A. W. Tucker, Editors, *Linear Inequalities and Related Systems*, page 19–40, Princeton University Press, Princeton, NJ, 1956.

[28] C. C. Gonzaga. An algorithm for solving linear programming problems in $O(n^3L)$ operations. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, Springer Verlag, New York, 1989, pages 1–28.

[29] C. C. Gonzaga and M. J. Todd. An $O(\sqrt{n}L)$–iteration large–step primal–dual affine algorithm for linear programming. *SIAM J. Optimization*, 2:349–359, 1992.

[30] J. Hartmanis and R. E. Stearns, On the computational complexity of algorithms. *Trans. A.M.S*, 117:285–306, 1965.

[31] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming, Algorithms and Complexity.* Ph.D. Thesis, Faculty of Mathematics and Informatics, TU Delft, NL–2628 BL Delft, The Netherlands, 1992.

[32] P. Huard. Resolution of mathematical programming with nonlinear constraints by the method of centers. In J. Abadie, editor, *Nonlinear Programming*, pages 207–219. North Holland, Amsterdam, The Netherlands, 1967.

[33] N. K. Karmarkar. A new polynomial–time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[34] L. G. Khachiyan. A polynomial algorithm for linear programming. *Doklady Akad. Nauk USSR*, 244:1093–1096, 1979. Translated in *Soviet Math. Doklady*, 20:191–194, 1979.

[35] V. Klee and G. J. Minty. How good is the simplex method. In O. Shisha, editor, *Inequalities III*, Academic Press, New York, NY, 1972.

[36] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial–time algorithm for a class of linear complementarity problems. *Math. Programming*, 44:1–26, 1989.

[37] M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Math. Programming*, 50:331–342, 1991.

[38] Z. Q. Luo, J. Sturm and S. Zhang. Conic Convex Programming And Self-Dual Embedding. *Optimization Methods and Software*, 14:169–218, 2000.

[39] I. J. Lustig, R. E. Marsten, and D. F. Shanno. On implementing Mehrotra's predictor–corrector interior point method for linear programming. *SIAM J. Optimization*, 2:435–449, 1992.

[40] L. McLinden. The analogue of Moreau's proximation theorem, with applications to the nonlinear complementarity problem. *Pacific Journal of Mathematics*, 88:101–161, 1980.

[41] N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, pages 131–158. Springer Verlag, New York, 1989.

[42] S. Mehrotra. On the implementation of a primal–dual interior point method. *SIAM J. Optimization*, 2(4):575–601, 1992.

[43] S. Mizuno, M. J. Todd, and Y. Ye. On adaptive step primal–dual interior–point algorithms for linear programming. *Mathematics of Operations Research*, 18:964–981, 1993.

[44] R. D. C. Monteiro and I. Adler. Interior path following primal–dual algorithms : Part I : Linear programming. *Math. Programming*, 44:27–41, 1989.

[45] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*, volume 3 of *Sigma Series in Applied Mathematics*, chapter 11.4.1 : The Karmarkar's algorithm for linear programming, pages 469–494. Heldermann Verlag, Nassauische Str. 26, D–1000 Berlin 31, Germany, 1988.

[46] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming.* New York, McGraw-Hill Companies, Inc., 1996.

[47] Yu. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms.* SIAM Publications. SIAM, Philadelphia, 1993.

[48] Yu. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1):1–42, 1997.

[49] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice–Hall, Englewood Cliffs, NJ, 1982.

[50] J. Renegar. A polynomial–time algorithm, based on Newton's method, for linear programming. *Math. Programming*, 40:59–93, 1988.

[51] C. Roos, T. Terlaky and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, Chichester, 1997.

[52] R. Saigal. *Linear Programming: Modern Integrated Analysis.* Kluwer Academic Publisher, Boston, 1995.

[53] G. Sonnevend. An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prekopa, J. Szelezsan, and B. Strazicky, editors, *System Modelling and Optimization : Proceedings of the 12th IFIP–Conference held in Budapest, Hungary, September 1985*, volume 84 of *Lecture Notes in Control and Information Sciences*, pages 866–876. Springer Verlag, Berlin, Germany, 1986.

[54] K. Tanabe. Complementarity–enforced centered Newton method for mathematical programming. In K. Tone, editor, *New Methods for Linear Programming*, pages 118–144, The Institute of Statistical Mathematics, 4–6–7 Minami Azabu, Minatoku, Tokyo 106, Japan, 1987.

[55] M. J. Todd. A low complexity interior point algorithm for linear programming. *SIAM J. Optimization*, 2:198–209, 1992.

[56] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15:508–529, 1990.

[57] L. Tunccel. Constant potential primal–dual algorithms: A framework. *Math. Programming*, 66:145–159, 1994.

[58] R. Tutuncu. An infeasible-interior-point potential-reduction algorithm for linear programming. Ph.D. Thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1995.

[59] P. M. Vaidya. An algorithm for linear programming which requires $O((m+n)n^2 + (m+n)^{1.5}nL)$ arithmetic operations. *Math. Programming*, 47:175–201, 1990. Condensed version in : *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, 1987, pages 29–38.

[60] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[61] R. J. Vanderbei. *Linear Programming: Foundations and Extensions* Kluwer Academic Publishers, Boston, 1997.

[62] S. A. Vavasis. *Nonlinear Optimization: Complexity Issues.* Oxford Science, New York, NY, 1991.

[63] S. J. Wright. *Primal-Dual Interior-Point Methods* SIAM, Philadelphia, 1996.

[64] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Math. Programming*, 50:239–258, 1991.

[65] Y. Ye, M. J. Todd and S. Mizuno. An $O(\sqrt{n}L)$–iteration homogeneous and self–dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.

[66] Y. Zhang and D. Zhang. On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms. *Math. Programming*, 68:303–317, 1995.

Index