Linear Programming

LEO LIBERTI

LIX, École Polytechnique, F-91128 Palaiseau, France

(liberti@lix.polytechnique.fr)

November 11, 2007

Abstract

This paper is a short didactical introduction to Linear Programming (LP). The main topics are: formulations, notes in convex analysis, geometry of LP, simplex method, duality, ellipsoid algorithm, interior point methods.

Contents

1 Introduction

Linear Programming (LP) is in some sense the fundamental tool of Operations Research. The first operations research programs have been modelled by using linear objective function and constraints, and, to date, the Simplex Method for solving LPs is one of the most practically efficient and powerful algorithms in Operations Research [Dan63].

The set of problems that can be modelled with LP techniques is large indeed, encompassing production management, human resources, blending, marketing, transportation, distribution, porfolio selection. Furthermore, LP is used to solve many sub-problems generated by complex algorithms.

1.1 Example

Diet Problem. *A diet of m nutrients is healthy if it has at least quantities $b = (b_1, \dots, b_m)$ of the nutrients. In order to compose such a diet, we need to buy quantities $x = (x_1, \dots, x_n)$ of n foods having unit costs $c = (c_1, \dots, c_n)$. Food j contains a_{ij} units of nutrient i. The solution of the following LP problem yields an x that satisfies the requirements of a healthy diet at minimum cost [BV04].*

$$\left.\begin{array}{rrcl} \min_x & \displaystyle\sum_{i=1}^{n} c^\top x & & \\ & Ax & \geq & b \\ & x & \geq & 0. \end{array}\right\} \quad (1)$$

1.2 Example

Transportation Problem. *Consider a transportation network modelled by a weighted bipartite directed graph $B = (U, V, A, d)$ with a set of departure vertices U, a set of destinations V, a set of arcs $A = \{(u, v) \mid u \in U, v \in V\}$ weighted by a nonnegative distance function $d : A \to \mathbb{R}_+$. A certain amount of material a_u is stored at each departure vertex $u \in U$, and we associate to each destination $v \in V$ a given demand b_v of the same material. The cost of routing a unit of material from $u \in U$ to $v \in V$ is directly proportional to the distance d_{uv}. We have to determine the transportation plan of least cost satisfying the demands at the destinations. The variables x_{uv} in the model below, associated to each arc $(u, v) \in A$, denote the amount of material routed on the arc.*

$$\left.\begin{array}{rccl}
\min_x & \sum\limits_{(u,v)\in A} d_{uv}x_{uv} & & \\
\forall u \in U & \sum\limits_{v\in V} x_{uv} & \leq & a_u \\
\forall v \in V & \sum\limits_{u\in U} x_{uv} & \geq & b_v \\
\forall (u, v) \in A & x_{uv} & \geq & 0.
\end{array}\right\} \qquad (2)$$

The above formulation correctly models the problem.

1.3 Example

Network Flow. *Given a network on a directed graph $G = (V, A)$ with a source node s, a destination node t, and capacities u_{ij} on each arc (i, j). We have to determine the maximum amount of material flow that can circulate on the network from s to t. The variables x_{ij}, defined for each arc (i, j) in the graph, denote the quantity of flow units.*

$$\left.\begin{array}{rccc}
\max_x & \sum\limits_{i\in\delta^+(s)} x_{si} & & \\
\forall\ i \leq V, i \neq s, i \neq t & \sum\limits_{j\in\delta^+(i)} x_{ij} & = & \sum\limits_{j\in\delta^-(i)} x_{ji} \\
\forall (i, j) \in A & 0 \leq x_{ij} \leq u_{ij}. & &
\end{array}\right\} \qquad (3)$$

The above formulation correctly models the problem.

The rest of this section is devoted to basic definitions, modelling tricks, and reformulations. We shall then give some brief notes about the geometry of linear programming. Finally, we shall give an overview of the most widely known methods for solving LPs.

1.1 Canonical and Standard forms

1.4 Definition

A linear programming problem (usually simply called LP) is a mathematical programming problem of the form:

$$\left.\begin{array}{rl}
\min_x & f(x) \\
s.t. & g(x) \leq 0,
\end{array}\right\} \qquad (4)$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, such that f, g are linear forms.

An LP is in *canonical form* if it is modelled as follows:

$$\left.\begin{array}{rl}
\min_x & c^\top x \\
\text{s.t.} & Ax \geq b \\
& x \geq 0
\end{array}\right\} \qquad (5)$$

where $x \in \mathbb{R}^n$ are the decision variables, $c \in \mathbb{R}^n$ is a cost vector, A is an $m \times n$ matrix (assume $\text{rank}(A) = m$), and b a vector in $\mathbb{R}^m$.

An LP is in *standard form* if it is modelled as follows:

$$\left.\begin{array}{rl}\min_x & c^\top x \\ \text{s.t.} & Ax = b \\ & x \geq 0\end{array}\right\} \tag{6}$$

again, $x \in \mathbb{R}^n$ are the decision variables, $c \in \mathbb{R}^n$ is a cost vector, A is an $m \times n$ matrix with (assume $\text{rank}(A) = m$), and b a vector in $\mathbb{R}^m$. An LP in canonical form can be put into standard form via the introduction of slack variables (see section 1.3).

Formally, we only deal with minimization, as maximization can be reduced to minimization by the following simple rule:

$$\max_{x \in X} f(x) = -\min_{x \in X} -f(x).$$

1.2 The notion of minimum

1.5 Definition
A point $x^* \in \mathbb{R}^n$ is feasible with respect to a problem (4) if $g(x^*) \leq 0$.

1.6 Definition
A feasible point $x^* \in \mathbb{R}^n$ is a local minimum *of problem (4) if there is a ball B centered at x^* and with radius $r \in \mathbb{R}_+$ such that for each feasible $x \in B$ we have $f(x^*) \leq f(x)$.*

1.7 Definition
A feasible point x^* is a global minimum *of problem (4) if there if for any feasible $x \in \mathbb{R}^n$ we have $f(x^*) \leq f(x)$.*

1.3 Common reformulations

In this section we report the most common reformulations that apply to linear problems.

1. Transforming an inequality to an equation. $Ax \leq b$ is equivalent to $Ax + s = b$ with $s \geq 0$ (introduction of *slack variables*).
2. Transforming unconstrained variables to constrained ones. An unconstrained variable x can be replaced by the difference $x_1 - x_2$ with $x_1, x_2 \geq 0$.
3. Piecewise linear minimization:
$$\min_{x \in X} \max_{i \leq m} \{{c_i}^\top x + d_i\} = \min\{t \in \mathbb{R} \mid \forall i \leq m \ ({c_i}^\top x + d_i \leq t), x \in X\}.$$
4. Absolute values (let $|x| = (|x_1|, \ldots, |x_n|)$):
$$\begin{array}{lll}\text{A:} & \min_{x \in X} c^\top |x| = & \min\{c^\top y \mid -y \leq x \leq y \wedge y \in \mathbb{R}^n_+, x \in X\} \\ \text{B:} & \min_{x \in X} c^\top |x| = & \min\{c^\top (x^+ + x^-) \mid x^+ - x^- = x \wedge x^+, x^- \in \mathbb{R}^n_+, x \in X\}.\end{array}$$
5. Linear fractional programming:
$$\min \left\{ \frac{c^\top x + d}{f^\top x + g} \mid Ax = b, f^\top x + g > 0, x \geq 0 \right\}.$$
We introduce variables $y = \frac{x}{f^\top x + g}$ and $z = \frac{1}{f^\top x + g}$, and the constraint $f^\top y + gz = 1$. Now the objective function becomes $c^\top y + dz$, and the constraint $Ax = b$ can be expressed in function of y, z as $Ay - bz = 0$. It can be shown that these two formulations yield the same solutions.

2 Convexity

Convex problems have the fundamental property of global minimality, i.e. if a point is locally minimal w.r.t. to a convex function f defined over a convex set X, then it is also globally minimal. LP problems are a subclass of convex problems. In this section we shall introduce the basic notions of convex analysis.

2.1 Convex sets

2.1 Definition
A set $S \subseteq \mathbb{R}^n$ is convex if for any two points $x, y \in S$ the segment between them is wholly contained in S, that is, $\forall \lambda \in [0,1]$ $(\lambda x + (1-\lambda)y \in S)$.

A linear equation $a^\top x = b$ where $a \in \mathbb{R}^n$ defines a hyperplane in $\mathbb{R}^n$. The corresponding linear inequality $a^\top x \leq b$ defines a closed half-space. Both hyperplanes and closed half-spaces are convex sets. Since any intersection of convex sets is a convex set, the subset of $\mathbb{R}^n$ defined by the system of closed half-spaces $Ax \geq b, x \geq 0$ is convex.

2.2 Definition
A polyhedron *is the intersection of a finite number of closed halfspaces. A bounded, non-empty polyhedron is a* polytope.

2.3 Definition
Let $S = \{x_i \mid i \leq n\}$ be a finite set of points in $\mathbb{R}^n$.

1. *The set span(S) of all linear combinations $\sum_{i=1}^n \lambda_i x_i$ of vectors in S is called the* linear hull *(or* linear closure*, or* span*) of S.*

2. *The set aff(S) of all affine combinations $\sum_{i=1}^n \lambda_i x_i$ such that $\sum_{i=1}^n \lambda_i = 1$ of vectors in S is called the* affine hull *(or* affine closure*) of S.*

3. *The set cone(S) of all conic combinations $\sum_{i=1}^n \lambda_i x_i$ such that $\forall i \leq n$ $(\lambda_i \geq 0)$ of vectors in S is called the* conic hull *(or* conic closure*, or* cone*) of S. A conic combination is* strict *if for all $i \leq n$ we have $\lambda_i > 0$.*

4. *The set conv(S) of all convex combinations $\sum_{i=1}^n \lambda_i x_i$ such that $\sum_{i=1}^n \lambda_i = 1$ and $\forall i \leq n$ $(\lambda_i \geq 0)$ of vectors in S is called the* convex hull *(or* convex closure*) of S. A convex combination is* strict *if for all $i \leq n$ we have $\lambda_i > 0$.*

2.4 Definition
Consider a polyhedron P and a closed half-space H in $\mathbb{R}^n$. Let $\overline{H \cap P}$ be the affine closure of $H \cap P$ and $d = \dim(\overline{H \cap P})$. If $d < n$ then $H \cap P$ is a face *of P. If $d = 0$, $H \cap P$ is called a* vertex *of P, and if $d = 1$, it is called an* edge*. If $d = n-1$ then $H \cap P$ is a* facet *of P.*

2.5 Lemma
Let $x^* \in P = \{x \geq 0 \mid Ax \geq b\}$. Then x^* is a vertex of P if and only if for any pair of distinct points $x', x'' \in P$, x^* cannot be a strict convex combination of x', x''.

Proof. ($\Rightarrow$) Suppose x^* is a vertex of P and there are distinct points $x', x'' \in P$ such that there is a $\lambda \in (0,1)$ with $x^* = \lambda x' + (1-\lambda)x''$. Since x^* is a vertex, there is a halfspace $H = \{x \in \mathbb{R}^n \mid h^\top x \leq d\}$ such that $H \cap P = \{x^*\}$. Thus $x', x'' \notin H$ and $h^\top x' > d$, $h^\top x'' > d$. Hence $h^\top x^* = h^\top(\lambda x' + (1-\lambda)x'') > d$, whence $x^* \notin H$, a contradiction. ($\Leftarrow$) Assume that x^* cannot be a strict convex combination of any pair of distinct points x', x'' of P, and suppose that x^* is not a vertex of P. Since x^* is not a vertex, there is a ball S of radius $\varepsilon > 0$ and center x^* wholly contained in P. Let $x' \in S$ such that $||x' - x^*|| = \frac{\varepsilon}{2}$. Let $x'' = (x^* - x') + x^*$. By construction, x'' is the point symmetric to x' on the line passing through x' and

x^*. Thus, $||x'' - x^*|| = \frac{\varepsilon}{2}$, $x'' \neq x'$, $x'' \in P$, and $x^* = \frac{1}{2}x' + \frac{1}{2}x''$, which is a strict convex combination of x', x'', a contradiction. □

2.2 Convex functions

2.6 Definition
A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is convex if for all $x, y \in X$ and for all $\lambda \in [0,1]$ we have:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y).$$

Note that for this definition to be consistent, $\lambda x + (1-\lambda)y$ must be in the domain of f, i.e. X must be convex. This requirement can be formally relaxed if we extend f to be defined outside X.

The main theorem in convex analysis, and the fundamental reason why convex functions are useful, is that a *local minimum* of a convex function over a convex set is also a *global minimum*, which we prove in Thm. 2.7. The proof is described graphically in Fig. 1.

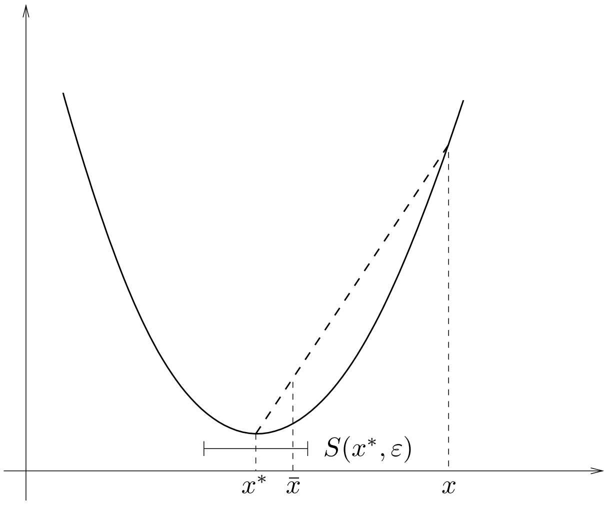

Figure 1: Graphical sketch of the proof of Thm. 2.7. The fact that $f(x^*)$ is the minimum in $S(x^*, \varepsilon)$ is enough to show that for any point $x \in X$, $f(x^*) \leq f(x)$.

2.7 Theorem
Let $f : \mathbb{R}^n \leftarrow \mathbb{R}$ be a convex function and $X \subseteq \mathbb{R}^n$ be a convex set. Given a point $x^* \in X$, suppose that there is a ball $S(x^*, \varepsilon) \subset X$ such that for all $x \in S(x^*, \varepsilon)$ we have $f(x) \geq f(x^*)$. Then $f(x^*) \leq f(x)$ for all $x \in X$.

Proof. Let $x \in X$. Since f is convex over X, for all $\lambda \in [0,1]$ we have $f(\lambda x^* + (1-\lambda)x) \leq \lambda f(x^*) + (1-\lambda)f(x)$. Notice that there exists $\bar{\lambda} \in (0,1)$ such that $\bar{\lambda}x^* + (1-\bar{\lambda})x = \bar{x} \in S(x^*, \varepsilon)$. Consider $\bar{x}$: by convexity of f we have $f(\bar{x}) \leq \bar{\lambda}f(x^*) + (1-\bar{\lambda})f(x)$. After rearrangement, we have

$$f(x) \geq \frac{f(\bar{x}) - \bar{\lambda}f(x^*)}{1-\bar{\lambda}}.$$

Since $\bar{x} \in S(x^*, \varepsilon)$, we have $f(\bar{x}) \geq f(x^*)$, thus

$$f(x) \geq \frac{f(x^*) - \bar{\lambda}f(x^*)}{1-\bar{\lambda}} = f(x^*),$$

as required. □

3 The Simplex method

The simplex method is the fundamental algorithm used in linear programming. Although all implementations of the simplex algorithm have exponential time complexity, in practice it has been shown that the simplex algorithm performs as $O(m+n)$ in most cases, so it is a very efficient algorithm. It was invented by Dantzig in the late forties [Dan63]; rumour has it that the young Dantzig approached Von Neumann to present his results, and met with the response that his algorithm was correct but not very innovative. Dantzig himself, for the very same reason, chose not to immediately publish his algorithm in a scientific journal paper, although, to date, the simplex algorithm is the most famous algorithm in Operations Research and one of the most famous in the whole field of applied mathematics.

The simplex algorithms rests on four main observations.

- The LP (5) is equivalent to minimizing a linear form over a polyhedron.
- The minimum of a linear form over a polyhedron is attained at a vertex of the polyhedron (cf. Thm. 3.5): since there are finitely many vertices in a polyhedron in $\mathbb{R}^n$, the (continuous) LP problem can be transformed into a finite search.
- Verifying whether a vertex of a polyhedron is a local minimum w.r.t. a linear form is easy, it suffices to check that all adjacent vertices have higher associated objective function value.
- Polyhedra are convex sets and linear forms are convex functions, so any local minimum is also a global minimum.

The simplex algorithm starts from a vertex of the feasible polyhedron and moves to an adjacent vertex with lower associated objective function value. When no such vertex exists, the vertex is the minimum and the algorithm stops.

3.1 Geometry of Linear Programming

The material in this section is taken from [PS98, Fis99]. Consider the LP problem (in standard form)

$$\min_{x \in P} c^\top x \tag{7}$$

where P is the polyhedron $\{x \in \mathbb{R}^n_+ \mid Ax = b\}$ and $c \in \mathbb{R}^n$.

3.1 Definition
A set $\{A_i \mid i \in \beta\}$ of m linearly independent columns of A is a basis *of A. The variables $\{x_i \mid i \in \beta\}$ corresponding to the indices β of the basis are called* basic variables. *All other variables are called* nonbasic variables.

This suggests that we can partition the columns of A in $(B|N)$ where B is the nonsingular, square matrix of the basic columns and N are the nonbasic columns. Correspondingly, we partition the variables x into (x_B, x_N).

3.2 Definition
Given a polyhedron $P = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$, the feasible vectors x having $x_B = B^{-1}b \geq 0$ and $x_N = 0$ are called basic feasible solutions *(bfs) of P.*

3.3 Lemma
Given a polyhedron $P = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$ and a bfs x^* for P, there exists a cost vector $c \in \mathbb{R}^n$ such that x^* is the unique optimal solution of the problem $\min\{c^\top x \mid x \in P\}$.

Proof. Let $c_j = 0$ for all j such that x_j is a basic variable, and $c_j = 1$ otherwise. □

The most important result in this section states that bfs's correspond to vertices of P.

3.4 Theorem
Given a polyhedron $P = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$, any bfs for P is vertex of P, and vice versa.

Proof. ($\Rightarrow$) Let $x^* = (x^*_B, 0)$, with $x^*_B \geq 0$, be a bfs for P. By Lemma 3.3 there is a cost vector c such that for all $x \in P$, x^* is the unique vector such that $c^\top x^* \leq c^\top x$ for all $x \in P$. Thus, the hyperplane $c^\top(x - x^*) = 0$ intersects P in exactly one point, namely x^*. Hence x^* is a vertex of P. ($\Leftarrow$) Assume now that x^* is a vertex of P and suppose, to get a contradiction, that it is not a bfs. Consider the columns A_j of A such that $j \in \beta = \{j \leq n \mid x^*_j > 0\}$. If A_j are linearly independent, we have immediately that x^* a bfs for P, which contradicts the hypothesis. Thus, suppose A_j are linearly dependent. This means that there are scalars d_j, not all zero, such that $\sum_{j\in\beta} d_j A_j = 0$. On the other hand, since x^* satisfies $Ax = b$, we have $\sum_{j\in\beta} x_j A_j = b$. Thus, for all $\varepsilon > 0$, we obtain $\sum_{j\in\beta}(x_j - \varepsilon d_j)A_j = b$ and $\sum_{j\in\beta}(x_j + \varepsilon d_j)A_j = b$. Let x' have components $x_j - \varepsilon d_j$ for all $j \in \beta$ and 0 otherwise, and x'' have components $x_j + \varepsilon d_j$ for all $j \in \beta$ and 0 otherwise. By choosing a small enough ε we can ensure that $x', x'' \geq 0$. Since $Ax' = Ax'' = b$ by construction, both x' and x'' are in P. Thus, $x^* = \frac{1}{2}x' + \frac{1}{2}x''$ is a strict convex combination of two points of P, hence by Lemma 2.5 it cannot be a vertex of P, contradicting the hypothesis. □

By the following theorem, in order to solve an LP all we have to do is compute the objective function at each vertex of the feasible polyhedron.

3.5 Theorem
Consider problem (7). If P is closed and bounded, there is at least one bfs that solves the problem. Furthermore, if x', x'' are two distinct solutions, any convex combination of x', x'' also solves the problem.

Proof. Since the polyhedron P is closed and bounded, the function $f(x) = c^\top x$ attains a minimum on P, say at x' (and by convexity of P, x' is the global minimum). Since $x' \in P$, x' is a convex combination of the vertices $v_1, \ldots, v_p$ of P, say $x' = \sum_{i=1}^{p} \lambda_i v_i$ with $\lambda_i \geq 0$ for all $i \leq p$ and $\sum_{i=1}^{p} \lambda_i = 1$. Thus,

$$c^\top x' = \sum_{i=1}^{p} \lambda_i c^\top v_i.$$

Let $j \leq p$ be such that $c^\top v_j \leq c^\top v_i$ for all $i \leq p$. Then

$$\sum_{i=1}^{p} \lambda_i c^\top v_i \geq c^\top v_j \sum_{i=1}^{p} \lambda_i = c^\top v_j,$$

whence $c^\top x' \geq c^\top v_j$. But since $c^\top x'$ is minimal, we have $c^\top x' = c^\top v_j$, which implies $x' = v_j$, a vertex of P. For the second part of the theorem, consider a convex combination $x = \lambda x' + (1 - \lambda)x''$ with $\lambda \geq 0$. We have $c^\top x = \lambda c^\top x' + (1 - \lambda)c^\top x''$. Since x', x'' are solutions, we have $c^\top x' = c^\top x''$, and hence

$$c^\top x = c^\top x'(\lambda + (1 - \lambda)) = c^\top x',$$

which shows that x is also a solution of the problem. □

Theorem 3.5 states that P should be closed and bounded, so it requires that P should in fact be a polytope (polyhedra may be unbounded). In fact, this theorem can be modified[1] to apply to unbounded polyhedra by keeping track of the unboundedness directions (also called extreme rays).

3.2 Moving from vertex to vertex

Unfortunately, polyhedra may be defined by a number of vertices exponential in the size of the instance, so the above approach is not practical. However, it is possible to look for the optimal vertex by moving from vertex to vertex along the edges of the polyhedron following the direction of decreasing cost, and checking at each vertex if an optimality condition is satisfied. This is a summary description of the *simplex method*. Since vertices of a polyhedron correspond to bfs's by Theorem 3.4, and a bfs has at most m nonzero components out of n, there are at worst (nm) vertices in a given polyhedron. Thus, this algorithm has worst-case factorial complexity. However, its average performance is very good, as has been confirmed by more than 50 years of practice. In order to fully describe this method, we need an efficient way of moving along a path of vertices.

Consider a bfs x^* for $P = \{x \geq 0 \mid Ax = b\}$ and let β be the set of indices of the basic variables. Let A_i be the i-th column of A. We have:

$$\sum_{i \in \beta} x_i^* A_i = b. \tag{8}$$

Now, fix a $j \notin \beta$; A_j is a linear combination of the A_i in the basis. Thus, there exist multipliers x_{ij} such that for all $j \notin \beta$,

$$\sum_{i \in \beta} x_{ij} A_i = A_j. \tag{9}$$

Multiply Eq. (9) by a scalar θ and subtract it from Eq. (8) to get:

$$\sum_{i \in \beta} (x_i^* - \theta x_{ij}) A_i + \theta A_j = b. \tag{10}$$

Now suppose we want to move (by increasing θ from its initial value 0) from the current bfs to another point inside the feasible region. In order to move to a feasible point, we need $x_i^* - \theta x_{ij} \geq 0$ for all $i \in \beta$. If $x_{ij} \leq 0$ for all i, then θ can grow indefinitely (this means the polyhedron P is unbounded). Assuming a bounded polyhedron, we have a bounded $\theta > 0$. This means

$$\theta = \min_{\substack{i \in \beta \\ x_{ij} > 0}} \frac{x_i^*}{x_{ij}}. \tag{11}$$

If $\theta = 0$ then there is $i \in \beta$ such that $x_i^* = 0$. This means that the bfs x^* is *degenerate*. We assume a nondegenerate x^*. Let $k \in \beta$ be the index minimizing θ in the expression above. The coefficient of A_k in Eq. (10) becomes 0, whereas the coefficient of A_j is nonzero.

3.6 Proposition
Let x^* be a bfs, $j \notin \beta$, x_{ij} be as in Eq. (10) for all $i \in \beta$ and θ be as in Eq. (11). The point $x' = (x'_1, \ldots, x'_n)$ defined by

$$x'_i = \begin{cases} x_i^* - \theta x_{ij} & \forall i \in \beta \backslash \{k\} \\ \theta & i = k \\ 0 & \textit{otherwise} \end{cases}$$

is a bfs.

[1] See D. Bertsekas' course notes on convexity at http://www.dim.uchile.cl/~ceimat/sea/apuntes/Chapter1.pdf, Prop. 1.6.6, p. 109.

Proof. First notice that x' is a feasible solution by construction. Secondly, for all $i \notin \beta, i \neq$ we have $x'_i = x^*_i = 0$ and for all $i \in \beta, i \neq k$ we have $x'_i \geq 0$. By the definition of x', we have $x'_j \geq 0$ and $x'_k = 0$. Thus, if we define $\beta' = \beta \backslash \{k\} \cup \{j\}$, it only remains to be shown that $\{x_i \mid i \in \beta'\}$ is a set of basic variables for A. In other words, if we partition the columns of A according to the index set β', obtaining $A = (B'|N')$, we have to show that B' is nonsingular. Notice $(B|N) = A = (B'|N')$ implies $B^{-1}A = (I|B^{-1}N) = (B^{-1}B'|B^{-1}N')$ (here '=' is taken informally to mean: both matrices at each side of the equality sign, taken as column sets, are equal). Eq. (9) can be stated in matrix form as $BX^\top = N$ where X is the $(n-m) \times m$ matrix whose (p,q)-th entry is x_{pq}, thus $B^{-1}N = X^\top$. By construction of B' we have $B' = B \backslash \{A_k\} \cup \{A_j\}$. Thus, $B^{-1}B' = (e_1, \ldots, e_{k-1}, B^{-1}A_j, e_{k+1}, \ldots, e_n)$, where e_i is the vector with i-th component set to 1 and the rest set to zero, and $B^{-1}A_j$ is the j-th column of X, i.e. $(x_{1j}, \ldots, x_{nj})^\top$. Hence, $|\det(B^{-1}B')| = |x_{kj}| > 0$, since $\theta > 0$ implies $x_{kj} \neq 0$. Thus, $\det B' \neq 0$ and B' is nonsingular. □

Intuitively, Theorem 3.6 says that any column A_j outside the basis can replace a column A_k in the basis if we choose k as the index that minimizes θ in Eq. (11). Notice that this implies that the column A_k exiting the basis is always among those columns i in Eq. (9) which have multiplier $x_{ij} > 0$. In other words, a column A_j can replace column A_k in the basis only if the linear dependence of A_j on A_k is nonzero (A_k is a "nonzero component" of A_j). Informally, we say that x_j enters the basis and x_k leaves the basis. In order to formalize this process algorithmically, we need to find the value of the multipliers x_{ij}. By the proof of Prop. 3.6, x_{ij} is the (i,j)-th component of a matrix X satisfying $X^\top = B^{-1}N$. Knowing x_{ij} makes it possible to calculate β' and the corresponding partition B', N' of the columns of A. The new bfs x' can be obtained as $(B')^{-1}b$, since $Ax' = b$ implies $(B'|N')x' = b$ and hence $Ix' = (B')^{-1}b$ (recall $N'x' = 0$ since x' is a bfs).

3.3 Decrease direction

All we need now is a way to identify "convenient" variables to enter into the basis. In other words, from a starting bfs we want to move to other bfs's (with the method described above) so that the objective function value decreases. To this end, we iteratively select to enter the basis the variable x_j having the most negative *reduced cost* (the reduced costs are the coefficients of the objective function expressed in terms of the current nonbasic variables). Writing c as (c_B, c_N) according to the current basic/nonbasic partition, the reduced costs $\bar{c}^\top$ are obtained as $c^\top - c_B B^{-1}A$. The algorithm terminates when there is no negative reduced cost (i.e. no vertex adjacent to the current solution has a smaller objective function value). This criterion defines a local optimum. Since LP problems are convex, any local minimum is also a global one by Theorem 2.7.

3.4 Bland's rule

When a bfs is degenerate, an application of the simplex algorithm as stated above may cause the algorithm to fail to converge. This happens because a reduced cost in the objective function identifies a variable x_j to enter the basis, replacing x_k, with degenerate value $x'_j = \theta = 0$. This results in a new basis having exactly the same objective function value as before; at the next iteration, it may happen that the selected variable to enter the basis is again x_k. Thus, the current basis alternatively includes x_k and x_j without any change to the objective function value. It can be shown that the following simple rule entirely avoids these situations: whenever possible, always choose the variable having the lowest index for entering/leaving the basis.

3.5 Simplex method in matrix form

Here we give a summary of the algebraic operations involved in a generic simplex algorithm step applied to a linear programming problem in standard form $\min\{c^\top x \mid Ax = b \wedge x \geq 0\}$. Suppose we are given a current bfs x ordered so that variables $x_B = (x_1, \ldots, x_m, 0, \ldots, 0)$ are basic and $x_N = (0, \ldots, 0, x_{m+1}, \ldots, x_n)$ are nonbasic. We write $A = B + N$ where B consists of the basic columns $1, \ldots m$ of A and 0 in the columns $m+1, \ldots n$, and N consists of 0 in the first m columns and then the nonbasic columns $m+1, \ldots n$ of A.

1. *Express the basic variables in terms of the nonbasic variables.* From $Ax = b$ we get $Bx_B + Nx_N = b$. Let B^{-1} be the inverse of the square submatrix of B consisting of the first m columns (i.e. the basic columns). We pre-multiply by B^{-1} to get:
$$x_B = B^{-1}b - B^{-1}Nx_N. \tag{12}$$
2. *Select an improving direction.* Express the objective function in terms of the nonbasic variables: $c^\top x = c_B^\top x_B + c_N^\top x_N = c_B^\top (B^{-1}b - B^{-1}Nx_N) + c_N^\top x_N$, whence:
$$c^\top x = c_B^\top B^{-1}b + \bar{c}_N^\top x_N, \tag{13}$$
where $\bar{c}_N^\top = c_N^\top - c_B^\top B^{-1}N$ are the reduced costs. If all the reduced costs are nonnegative there is no nonbasic variable which yields an objective function decrease if inserted in the basis, since the variables must also be nonegative. Geometrically speaking it means that there is no adjacent vertex with a lower objective function value, which in turn, by Thm. 2.7, means that we are at an optimum, and the algorithm terminates. Otherwise, select an index $h \in \{m+1, \ldots n\}$ of a nonbasic variable x_h with negative reduced cost (or, as in Section 3.4, select the least such h). We now wish to insert x_h in the basis by increasing its value from its current value 0. Geometrically, this corresponds to moving along an edge towards an adjacent vertex.
3. *Determine the steplength.* Inserting x_h in the basis implies that we should determine an index $l \leq m$ of a basic variable which exits the basis (thereby taking value 0). Let $\bar{b} = B^{-1}b$, and let $\bar{a}_{ij}$ be the (i, j)-th component of $B^{-1}N$ (notice $\bar{a}_{ij} = 0$ for $j \leq m$). By (12) we can write $x_i = \bar{b}_i - \sum\limits_{j=m+1}^{n} \bar{a}_{ij}x_j$ for all $i \leq m$. Since we only wish to increase the value of x_h and all the other nonbasic variables will keep value 0, we can write $x_i = \bar{b}_i - \bar{a}_{ih}x_h$. At this point, increasing the value of x_h may impact on the feasibility of x_i only if it becomes negative, which can only happen if $\bar{a}_{ih}$ is positive. Thus, we get $x_h \leq \frac{\bar{b}_i}{\bar{a}_{ih}}$ for each $i \leq m$ and $\bar{a}_{ih} > 0$, and hence:
$$\begin{aligned} l &= \operatorname{argmin}\{\frac{\bar{b}_i}{\bar{a}_{ih}} \mid i \leq m \wedge \bar{a}_{ih} > 0\} & (14) \\ x_h &= \frac{\bar{b}_l}{\bar{a}_{lh}}. & (15) \end{aligned}$$
The above procedure fails if $\bar{a}_{ih} \leq 0$ for all $i \leq m$. Geometrically, it means that x_h can be increased in value without limit: this implies that the value of the objective function becomes unbounded. In other words, the problem is unbounded.

4 Duality

Every mathematical programming problem has a *dual form*. The original problem is sometimes called the *primal problem*. It can be shown that the optimal solution of the dual of any convex problem is exactly the same as the solution of the primal problem (this is true for LP problems also, since they are convex problems). For nonconvex problems, the solution of the dual yields a lower bound to the solution of the primal problem. In this section, we shall derive the dual of an LP problem as a special case of a much more general setting.

4.1 Equality constraints and Lagrange Multipliers method

Some material for this section has been taken from [Pug02]. For a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ we define the function vector ∇f as $(\nabla f)(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)^\top$, where $x = (x_1, \ldots, x_n)^\top$. We denote by $\nabla f(x')$ the evaluation of ∇f at x'. If g is a vector-valued function $g(x) = (g_1(x), \ldots, g_m(x))$, then ∇g is the set of function vectors $\{\nabla g_1, \ldots, \nabla g_m\}$, and $\nabla g(x')$ is the evaluation of ∇g at x'.

Consider the following nonlinear, possibly nonconvex programming problem with continuous variables defined over $\mathbb{R}^n$:

$$\left.\begin{array}{rlcl} \min\limits_{x\in\mathbb{R}^n} & f(x) & & \\ \text{s.t.} & g(x) & = & 0, \end{array}\right\} \tag{16}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are C^1 (i.e., continuously differentiable) functions.

A *constrained critical point* of problem (16) is a point $x^* \in \mathbb{R}^n$ such that $\nabla f(x^*) = 0$ and $g(x^*) = 0$. It is easy to show that such an x^* is a maximum, or a minimum, or a saddle point of $f(x)$.

4.1 Theorem (Lagrange Multiplier Method)
If x^* is a constrained critical point of problem (16), $m \le n$, and $\nabla g(x^*)$ is a linearly independent set of vectors, then $\nabla f(x^*)$ is a linear combination of the set of vectors $\nabla g(x^*)$.

Proof. Suppose, to get a contradiction, that $\nabla g(x^*)$ and $\nabla f(x^*)$ are linearly independent. In the case $\nabla f(x^*) = 0$, the theorem is trivially true, so assume $\nabla f(x^*) \neq 0$. Choose vectors $w_{m+2}, \ldots, w_n$ such that the set $J = \{\nabla g_1(x^*), \ldots, \nabla g_m(x^*), \nabla f(x^*), w_{m+2}, \ldots, w_n\}$ is a basis of $\mathbb{R}^n$. For $m + 2 \le i \le n$ define $h_i(x) = \langle w_i, x\rangle$. Consider the map

$$F(x) = (F_1(x), \ldots, F_n(x)) = (g_1(x), \ldots, g_m(x), f(x), h_{m+2}(x), \ldots, h_n(x)).$$

Since the jacobian of F evaluated at x^* is J, and J is nonsingular, by the inverse function theorem F is a local diffeomorphism of $\mathbb{R}^n$ to itself. Thus, the equations $y_i = F_i(x)$ $(i \le n)$ are a local change of coordinates in a neighbourhood of x^*. With respect to coordinates y_i, the surface S defined by $g(x) = 0$ is the coordinate hyperplane $0 \times \mathbb{R}^{n-m}$. Notice that the $(k+1)$-st coordinate, $y_{k+1} = f(x)$, cannot have a critical point on the coordinate plane $0 \times \mathbb{R}^{n-m}$, so x^* is not a constrained critical point, which is a contradiction. □

In the proof of the above theorem, we have implicitly used the fact that the criticality of points is invariant with respect to diffeomorphism (this can be easily established by showing that the derivatives of the transformed functions are zero at the point expressed in the new coordinates).

By Theorem (4.1) there exist scalars $\lambda_1, \ldots, \lambda_m$, such that

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) = 0.$$

The above condition is equivalent to saying that if x^* is a constrained critical point of f s.t. $g(x^*) = 0$, then x^* is a critical point of the following (unconstrained) function:

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x). \tag{17}$$

Function (17) is called the *Lagrangian* of f w.r.t. g, and $\lambda_1, \ldots, \lambda_m$ are called the *Lagrange multipliers*. Intuitively, when we are optimizing over a subspace defined by $Ax = b$, the optimization direction must be a linear combination of the vectors which are normal to the hyperplane system $Ax = b$. The situation is depicted in Fig. 2.

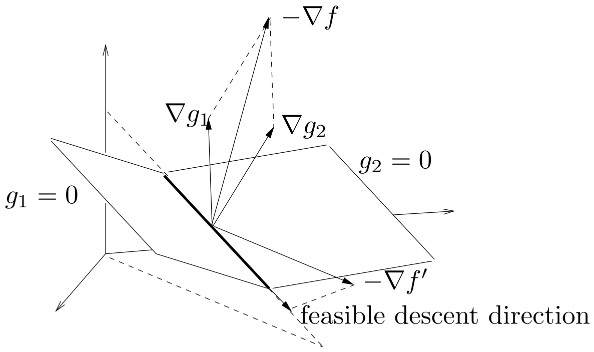

Figure 2: If ∇f is linearly dependent on the constraint gradients, there is no feasible descent direction. $\nabla f'$, which does not have this property, identifies a feasible descent direction when projected on the feasible space.

Hence, in order to find the solution of problem (16), by Theorem 4.1, one should find all critical points of $L(x, \lambda)$ and then select the one with minimum objective function value. In view of Theorem 3.5 and the fact that polyhedra have an exponential number of vertices, this approach is of limited use on all but the simplest problems. It does, however, provide at least necessary conditions for the characterization of a constrained minimum.

Notice, however, that problem (16) only caters for equality constraints; in particular, the range of each variable is $\mathbb{R}$ and there is no general way to reformulate an inequality constraint to an equation constraint unless one takes into account *nonnegative* slack variables (see Section 1.3), i.e. other inequality constraints. Furthermore, the proof of Theorem 4.1 is heavily based on the assumption that the only constraints of the problem are equality constraints, so an extension of the proof to encompass inequality constraints seems unlikely.

4.2 Inequality Constraints and Farkas' Lemma

Material from this section has been taken from [BSS93, Lee04] In order to take into account inequality constraints, we introduce the following simple example.

4.2 Example
Consider the problem $\min\{-x_1 - x_2 \mid x_1 - 1 \le 0, x_2 - 1 \le 0\}$. *The minimum is obviously at* $x^* = (1,1)$ *as depicted in Fig. 3 (the figure only shows the positive quadrant; the feasible region actually extends to the other quadrants). By inspection, it is easy to see that at the point* $x^* = (1,1)$ *any further movement towards the direction of objective function decrease would take* x^* *outside the feasible region, i.e. no feasible direction decreases the objective function value. Notice that the descent direction at* x^* *is in the cone defined by the normal vectors to the constraints in* x^*. *In this particular case, the descent direction of the objective function is the vector* $-\nabla f(x^*) = (1,1)$. *The normal vector to* $g_1(x) = x_1 - 1$ *at* x^* *is* $\nabla g_1(x^*) = (1,0)$ *and the normal vector to* $g_2(x) = x_2 - 1$ *at* x^* *is* $\nabla g_2(x^*) = (0,1)$. *Notice that we can express* $-\nabla f(x^*) = (1,1)$ *as* $\lambda_1(1,0) + \lambda_2(0,1)$ *with* $\lambda_1 = 1 > 0$ *and* $\lambda_2 = 1 > 0$. *In other words,* $(1,1)$ *is a conic combination of* $(1,0)$ *and* $(0,1)$.

If we now consider the problem of minimizing $\bar{f}(x) = x_1 + x_2$ *subject to the same constraints as above, it appears clear that* $x^* = (1,1)$ *cannot be a local minimum, as there are many feasible descent directions. Take for example the direction vector* $y = (-1,-1)$. *This direction is feasible w.r.t. the constraints:* $\nabla g_1(x^*)y = (1,0)(-1,-1) = (-1,0) \le 0$ *and* $\nabla g_2(x^*)y = (0,1)(-1,-1) = (0,-1) \le 0$. *Moreover, it is a nonzero descent direction:* $-\nabla\bar{f}(x^*)y = -(1,1)(-1,-1) = (1,1) > 0$.

In summary, either the descent direction at x^* *is a conic combination of the gradients of the constraints*

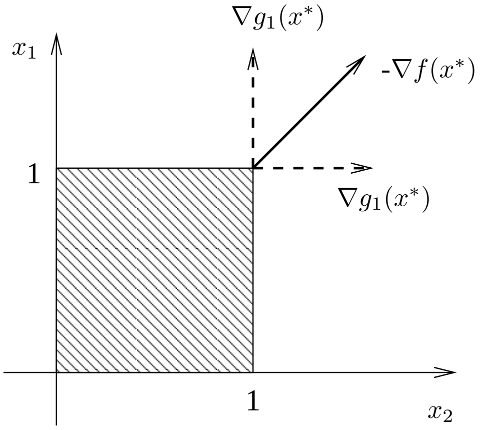

Figure 3: The problem of Example 4.2.

(and in this case x^* may be a local minimum), or there is a nonzero feasible descent direction (and in this case x^* cannot be a local minimum).

The example above is an application of a well known theorem of alternatives called Farkas' Lemma. This will be useful in Section 4.3 to give necessary conditions for a point to be a constrained local minimum of an inequality constrained problem.

4.3 Proposition
Given a non-empty, closed convex set $S \subseteq \mathbb{R}^n$ and a point $x^* \notin S$, there exists a unique point $x' \in S$ with minimum distance from x^*. Furthermore, x' is the minimizing point if and only if for all $x \in S$ we have $(x^* - x')^\top (x - x') \leq 0$.

Proof. Let $x' \in S$ be the point minimizing $f(x) = ||x^* - x||$ subject to $x \in S$ (which exists by Weierstraß Theorem). To show that it is unique, notice first that f is convex: for $\lambda \in [0,1]$ and $x_1, x_2 \in S$ we have $f(\lambda x_1 + (1-\lambda)x_2) \leq f(\lambda x_1) + f((1-\lambda)x_2) = \lambda f(x_1) + (1-\lambda) f(x_2)$ by triangular inequality and homogeneity of the norm. Suppose now $y \in \mathbb{R}^n$ such that $f(y) = f(x')$. Since $x', y \in S$ and S is convex, the point $z = \frac{x'+y}{2}$ is in S. We have $f(z) \leq \frac{f(x')}{2} + \frac{f(y)}{2} = f(x')$. Since $f(x')$ is minimal, $f(z) = f(x')$. Furthermore, $f(z) = ||x^* - z|| = ||x^* - \frac{x'+y}{2}|| = ||\frac{x^*-x'}{2} + \frac{x^*-y}{2}||$. By the triangle inequality, $f(z) \leq \frac{1}{2}||x^* - x'|| + \frac{1}{2}||x^* - y||$. Since equality must hold as $f(z) = f(x') = f(y)$, vectors $x^* - x'$ and $x^* - y$ must be collinear, i.e. there is $\theta \in \mathbb{R}$ such that $x^* - x' = \theta(x^* - y)$. Since $f(x') = f(y)$, $|\theta| = 1$. Supposing $\theta = -1$ we would have $x^* = \frac{x'+y}{2}$, which would imply $x^* \in S$, contradicting the hypothesis. Hence $\theta = 1$ and $x' = y$ as claimed. For the second part of the Proposition, assume x' is the minimizing point in S and suppose there is $x \in S$ such that $(x^* - x')^\top (x - x') > 0$. Take a point $y \neq x'$ on the segment $\overline{x, x'}$. Since S is convex, $y \in S$. Furthermore, since $y - x'$ is collinear to $x - x'$, $(x^* - x')^\top (y - x') > 0$. Thus, the angle between $y - x'$ and $x^* - x'$ is acute. Choose y close enough to x' so that the largest angle of the triangle T having x^*, x', y as vertices is the angle in y (such a choice is always possible). By elementary geometry, the longest side of such a triangle is the segment $\overline{x^*, x'}$ opposite to the angle in y. This implies $||x^* - y|| < ||x^* - x'||$, which is a contradiction, as x' was chosen to minimize the distance from x^*. Conversely, suppose for all $x \in S$ we have $(x^* - x')^\top (x - x') \leq 0$. This means that the angle in x' is obtuse (and hence it is the largest angle of T), and consequently the opposite side $\overline{x, x^*}$ is longer than the side $\overline{x', x^*}$. □

4.4 Proposition
Given a non-empty, closed convex set $S \subseteq \mathbb{R}^n$ and a point $x^* \notin S$, there exists a separating hyperplane $h^\top x = d$ (with $h \geq 0$) such that $h^\top x \leq d$ for all $x \in S$ and $h^\top x^* > d$.

Proof. Let x' be the point of S of minimum (strictly positive, since $x^* \not\in S$) distance from x^*. Let $y = \frac{x'+x^*}{2}$ be the midpoint between x', x^*. The plane normal to the vector $x^* - y$ and passing through y separates x^* and S. □

4.5 Theorem (Farkas' Lemma)
Let A be an $m \times n$ matrix and c be a vector in $\mathbb{R}^n$. Then exactly one of the following systems has a solution: (a) $Ax \leq 0$ and $c^\top x > 0$ for some $x \in \mathbb{R}^n$; (b) $\mu^\top A = c^\top$ and $\mu \geq 0$ for some $\mu \in \mathbb{R}^m$.

Proof. Suppose system (b) has a solution. Since $\mu^\top Ax = c^\top x$ for each $x \in \mathbb{R}^n$, pick x such that $Ax \leq 0$. Since $\mu \geq 0$, we have $c^\top x \leq 0$. Conversely, suppose system (b) has no solution. Let $\text{Im}_+(A^\top) = \{z \in \mathbb{R}^n \mid z^\top = \mu^\top A, \mu \geq 0\}$; $\text{Im}_+(A^\top)$ is convex, and $c \not\in \text{Im}_+(A^\top)$. By Prop. (4.4), there is a separating hyperplane $h^\top x = d$ such that $h^\top z \leq d$ for all $z \in \text{Im}_+(A^\top)$ and $h^\top c > d$. Since $0 \in \text{Im}_+(A^\top)$, $d \geq 0$, hence $h^\top c > 0$. Furthermore, $d \geq z^\top h = \mu^\top Ah$ for all $\mu \geq 0$. Since μ can be arbitrarily large, $\mu^\top Ah \leq d$ implies $Ah \leq 0$. Take $x = h$: x solves system (a). □

Farkas' Lemma has the following geometric interpretation: given A and c as in the statement of Theorem 4.5, either there is a vector x that makes a non-acute angle with every row of A and an acute angle with c, or c is in the conical hull of the rows of A (see Fig. 4).

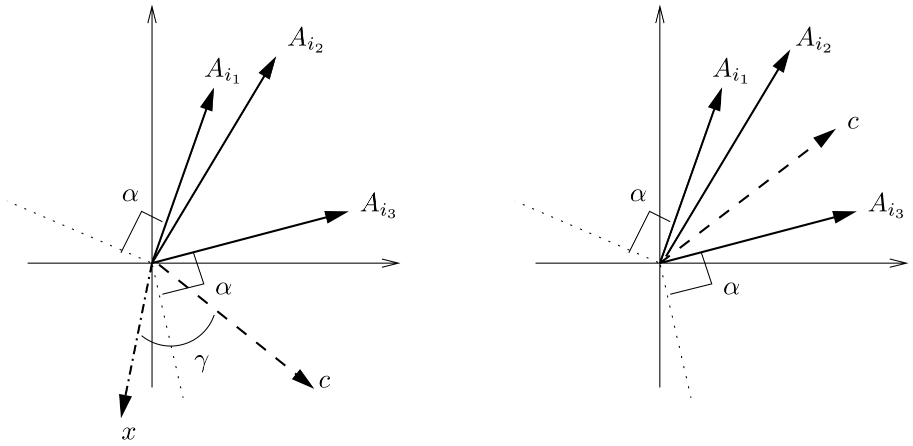

Figure 4: Geometric interpretation of Farkas' Lemma. Since $\alpha = \pi/2$ and $\gamma < \pi/2$, x makes non-acute angles with the rows of A, and an acute angle with c, which is not in the cone spanned by the rows of A (picture on the left). The alternative situation is that c is in the cone spanned by the rows of A, and an x with the mentioned properties does not exist (picture on the right).

4.3 Karush-Kuhn-Tucker necessary conditions

Consider the following problem:

$$\left.\begin{array}{rlcl} \min\limits_{x\in\mathbb{R}^n} & f(x) & & \\ \text{s.t.} & g(x) & \leq & 0, \end{array}\right\} \quad (18)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are C^1 functions. A *constrained minimum* of problem (18) is a minimum x^* of $f(x)$ such that $g(x^*) \leq 0$. It can be shown that if x^* is a constrained minimum then there is no nonzero feasible descent direction at x^* (see [BSS93], p. 141). We shall define a *feasible direction at* x^* as a direction vector y such that $(\nabla g(x^*))^\top y \leq 0$, and a *nonzero descent direction at* x^* as a direction vector y such that $(-\nabla f(x^*))^\top y > 0$.

4.6 Theorem (Karush-Kuhn-Tucker Conditions)
If x^* is a constrained minimum of problem (18), I is the maximal subset of $\{1,\ldots,m\}$ such that $g_i(x^*)=0$ for all $i \in I$, and $\nabla\bar{g}$ is a linearly independent set of vectors (where $\bar{g}=\{g_i(x^*) \mid i \in I\}$), then $\nabla f(x^*)$ is a conic combination of the vectors in $\nabla\bar{g}$, i.e. there exist scalars λ_i for all $i \in I$ such that the following conditions hold:

$$\begin{aligned} \nabla f(x^*) + \sum_{i\in I} \lambda_i \nabla g_i(x^*) &= 0 \qquad (19)\\ \forall i \in I \quad (\lambda_i &\geq 0). \qquad (20)\end{aligned}$$

Proof. Since x^* is a constrained minimum and $\nabla\bar{g}$ is linearly independent, there is no nonzero feasible descent direction at x^* such that $(\nabla\bar{g}(x^*))^\top y \leq 0$ and $-\nabla f(x^*)y > 0$. By a direct application of Farkas' Lemma (4.5), there is a vector $\lambda \in \mathbb{R}^{|I|}$ such that $\nabla(\bar{g}(x^*))\lambda = -\nabla f(x^*)$ and $\lambda \geq 0$. □

The Karush-Kuhn-Tucker (KKT) necessary conditions (19), (20) can be reformulated to the following:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) &= 0 \qquad (21)\\ \forall i \leq m \quad (\lambda_i g_i(x^*) &= 0) \qquad (22)\\ \forall i \leq m \quad (\lambda_i &\geq 0). \qquad (23)\end{aligned}$$

This is easily verified by defining $\lambda_i = 0$ for all $i \notin I$. Conditions (22) are called *complementary slackness conditions*, and they express the fact that if a constraint is not active at x^* then its corresponding Lagrange multiplier is 0.

4.4 Weak duality

Consider now a general NLP with both inequality and equality constraints:

$$\left.\begin{array}{rrcl} \min\limits_{x\in\mathbb{R}^n} & f(x) & & \\ \text{s.t.} & g(x) & \leq & 0 \\ & h(x) & = & 0, \end{array}\right\} \qquad (24)$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are C^1 functions. A *constrained minimum* of problem (24) is a minimum x^* of $f(x)$ such that $g(x^*) \leq 0$ and $h(x) = 0$.

By applying theorems 4.1 and 4.6, we can define the Lagrangian of problem (24) by the following:

$$L(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \mu_i h_i(x), \qquad (25)$$

and the corresponding KKT conditions as:

$$\left.\begin{array}{rcll} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) & = & 0 & \\ \lambda_i g_i(x^*) & = & 0 & \forall i \leq m \\ \lambda_i & \geq & 0 & \forall i \leq m. \end{array}\right\} \qquad (26)$$

In order to derive the general KKT conditions (26), it is sufficient to sum Eq. (19) and (21) and then divide the sum by the scalar 2.

4.7 Theorem (Weak duality)
Let S be the feasible region of problem (24) and x^* be its global solution. For each $\lambda \geq 0$ and μ,

$$\bar{L}(\lambda,\mu) = \min_{x\in S} L(x,\lambda,\mu) \leq f(x^*).$$

In other words, $\bar{L}(\lambda, \mu)$ is a lower bound for the solution of (24).

Proof. Since $x \in S$ and $\lambda \geq 0$, $\lambda^\top g(x) \leq 0$ and $\mu^\top h(x) = 0$. The result follows by Eq. (25). □

In view of Theorem 4.7, it makes sense to look for the greatest lower bound given by duality; this is defined as

$$\max_{\lambda \geq 0, \mu} \bar{L}(\lambda, \mu).$$

4.5 The dual of an LP problem

Consider now an LP in standard form (6). The Lagrangian is:

$$L(x, \lambda, \mu) = c^\top x + \lambda^\top(-x) + \mu^\top(Ax - b).$$

Its duality bound is given by:

$$\max_{\lambda \geq 0, \mu} \min_{x \in S} -b^\top \mu + x^\top(c - \lambda + A^\top \mu).$$

The KKT conditions (26) are:

$$\begin{array}{rcll} c - \lambda + A^\top \mu & = & 0 & \\ \lambda_i x_i & = & 0 & \forall i \leq n \\ \lambda_i & \geq & 0 & \forall i \leq n. \end{array}$$

By considering λ as slack variables (see Section 1.3), the first and third conditions above can simply be reformulated to:

$$A^\top \mu + c \geq 0.$$

The KKT conditions ensure $(c - \lambda + \mu A) = 0$, hence the duality bound can be obtained by solving the following *dual problem* in the μ variables:

$$\left.\begin{array}{lrcl} \max\limits_{\mu} & -b^\top \mu & & \\ \text{s.t.} & A^\top \mu + c & \geq & 0. \end{array}\right\} \tag{27}$$

4.8 Theorem (Strong duality for LP)
The objective function value at an optimum of the LP dual (27) is equal to the objective function value at an optimum of the original LP (6).

Proof. Consider the following "inverse" reformulation of the dual (27):

$$\left.\begin{array}{lrcl} \min\limits_{\mu} & b^\top \mu & & \\ \text{s.t.} & -A^\top \mu - c & \leq & 0. \end{array}\right\} \tag{28}$$

It is evident that if μ^* is a solution of (28), then it also solves (27) with objective function value $-b^\top \mu^*$. The Lagrangian for problem (28) is $L(\mu, x) = b^\top \mu + x^\top(-A^\top \mu - c)$, where $x \geq 0$ is the vector of Lagrange multipliers. By rearranging terms, we get $L(\mu, x) = -(Ax - b)^\top \mu - x^\top c$. By the KKT conditions, we get $\nabla L = 0$ (derivatives taken w.r.t. μ, not x), hence $Ax - b = 0$. The complementary slackness conditions (22) read $x^\top(-A^\top \mu^* - c) = 0$, i.e. $-b^\top \mu^* = c^\top x$; thus, there exist a primal feasible point x such that the objective function value of the primal problem evaluated at x is the same as the optimal objective function value of the dual problem. Since by weak duality $-b^\top \mu^* \leq c^\top x'$ for any primal feasible x', this implies that x is an optimal solution of the primal problem, and the theorem is proved. □

4.6 Economic interpretation of the dual

Consider the diet problem of Example 1.1. Its dual, $\max\{yb \mid yA \le c^\top \wedge y \ge 0\}$, can be interpreted as follows. A megalomaniac pharmaceutical firm wishes to replace human food with nutrient pills: it wishes to set the prices of the pills as high as possible, whilst being competitive with the cost of the foods. In this setting, y are the prices of the m nutrient pills, b is the quantity of nutrients required, and c are the costs of the food.

4.7 Sensitivity analysis

Material from this section has been taken from [Fis99]. We write the optimality conditions as follows:

$$\begin{aligned} \bar{b} &= B^{-1}b \ge 0 \qquad (29) \\ \bar{c}^\top &= c^\top - c_B^\top B^{-1}A \ge 0. \qquad (30) \end{aligned}$$

Eq. (29) expresses primal feasibility and Eq. (30) expresses dual feasibility.

Suppose now we have a variation in b, say $b \to b + \Delta b$: Eq. (29) becomes $B^{-1}b \ge -B^{-1}\Delta b$. This system defines a polyhedron in Δb where the optimal basis does not change. The variable values and objective function obviously change. The variation of the objective function is $(c_B^\top B^{-1})\Delta b = y^*\Delta b$, where y^* are the optimal dual variable values: these, therefore, can be seen to measure the sensitivity of the objective function value to a small change in the constraint coefficients.

5 Simplex variants

In this section we briefly describe some of the most popular variants of the simplex algorithm. Material in this section has been taken from [PS98, Fis99].

5.1 Revised Simplex method

The revised simplex mehod is basically a smart storage and update scheme for the data of the current iteration of the simplex algorithm. In practice, we only need to store B^{-1}, as the rest of the data can be obtained via premultiplication by B^{-1}. The disadvantages of this method reside in the high numerical instability of updating B^{-1} directly. The instability problem is usually addressed by storing B^{-1} in various factorized forms.

5.2 Two-phase Simplex method

If no starting bfs is available for (6), we can artificially look for one by solving the auxiliary problem:

$$\left.\begin{array}{rl} \min_{x,y} & \mathbf{1}y \\ \text{s.t.} & Ax + yI = b \\ & x \in \mathbb{R}^n_+ \\ & y \in \mathbb{R}^m_+ \end{array}\right\}$$

where $\mathbf{1}$ is the row vector consisting of all 1's. Here, the bfs where $B = I$ is immediately evident (all x are nonbasics, all y are basics), and if (6) is feasible, the optimal objective function value of the auxiliary problem will be 0 with $y = 0$, yielding a bfs in the x variables only. This solution can then be used as a starting bfs for the original problem.

5.3 Dual Simplex method

Another way to deal with the absence of a starting bfs is to apply the dual simplex method. What happens in the (primal) simplex algorithm is that we maintain primal feasibility by moving from vertex to vertex, whilst trying to decrease the objective function until this is not possible anymore. In the dual simplex algorithm, we maintain the optimality of the objective function (in the sense that the current dual simplex objective function value is always a lower bound with respect to the primal optimal value) whilst trying to achieve feasibility. We use the same notation as in Section 3.5.

We start with a (possibly infeasible) basic primal solution where all the reduced costs are nonnegative. If $\bar{b} = B^{-1}b = x_B \geq 0$ the algorithm terminates: if x_B is primal feasible, then we have an optimal basis. Otherwise, the primal problem is infeasible. If $b \not\geq 0$, we select $l \leq m$ such that $\bar{b}_l < 0$. We then find $h \in \{m+1, \ldots, n\}$ such that $\bar{a}_{lh} < 0$ and a_{lh} is minimum among $\{\frac{\bar{c}_j}{|\bar{a}_{lj}|} \mid j \leq n \wedge \bar{a}_{lj} < 0\}$ (this ensures that the reduced costs will stay nonnegative), and we swap x_l with x_h in the current basis. If $\bar{a}_{lj} \geq 0$ for all $j \leq n$, then the primal problem is infeasible.

The advantage of the dual simplex method is that we can add cutting planes (valid inequalities) to the simplex tableau during the execution of the algorithm. A valid cut should make the current optimal solution infeasible, but since the dual simplex bases are not assumed to be primal feasible, this is not a problem.

5.4 Column generation

It often happens that the number of variables in a problem is much larger than the number of constraints. Consequently, whilst the basis is has acceptably small cardinality, the computational costs of running the simplex algorithm on the problem are huge. If there exists an efficient procedure for finding the reduced cost of minimum value, we can insert columns in the simplex tableau as the need arises, thus dispensing from dealing with all variables at once. The problem of determining the reduced cost of minimum value is called the *pricing problem*. Column generation techniques are only useful if the pricing problem can be solved efficiently. The procedure stops when the pricing problem determines a minimum reduced cost of non-negative value. One example of application of column generation is the multicommodity network flow problem formulated so that each variable corresponds to a path on the network. There are exponentially many paths, but the pricing problem is a shortest path problem, which can be solved very efficiently.

6 Interior Point methods

Material for this section (except for Subsection 6.1) has been taken from [BSS93, Fle91, Tah92, JRTV93, BV04].

Kachiyan showed in 1979 that finding an optimal solution to a LP problem has polynomial worst-case complexity. Even though Kachiyan's Ellipsoid algorithm is of polynomial worst-case complexity, it does not work well in practice. In 1984, Karmarkar presented another polynomial algorithm for solving LP problems which was claimed to have useful practical applicability. Gill et al. showed in 1985 that Karmarkar's algorithm was in fact an Interior Point Method (IPM) with a log-barrier function applied to an LP problem. IPMs had been applied to nonlinear problems with considerable success in the late 60's (see Fiacco and McCormick's 1968 book). This spawned considerable interest in IPMs applied to LP problems; barrier optimizers for LP were incorporated in most commercial software codes (CPLEX, XPressMP). Barrier optimizers compete well with Simplex-based implementations specially on large-scale problems.

6.1 Ellipsoid Algorithm and complexity of LP

Material for this section has been taken from [NW88, KV00, PS98].

Any LP problem can be polynomially reduced to the problem of finding a feasible solution to a system of strict linear inequalities in n variables with rational coefficients. Let P_ε be the polyhedron corresponding to the given system of strict linear inequalities, and let E_0 be an ellipsoid such that $P_\varepsilon \subseteq E_0$. The Ellipsoid algorithm either finds a feasible solution $x^* \in P_\varepsilon$ or determines that P_ε is empty; we assume that the volume of P_ε is strictly positive if P_ε is nonempty, i.e. $P_\varepsilon \neq \emptyset \rightarrow \mathrm{Vol}(P_\varepsilon) > 0$. Let v be a number such that $\mathrm{Vol}(P_\varepsilon) > v$ if $P_\varepsilon \neq \emptyset$ and $V = \mathrm{Vol}(E_0)$, and let

$$K = \lceil 2(n+1)(\ln V - \ln v) \rceil.$$

Initially, $k = 0$.

1. Let $x^{(k)}$ be the centre of the ellipsoid E_k. If $x^{(k)} \in P_\varepsilon$, the algorithm terminate with solution $x^{(k)}$. Otherwise, there exists a violated strict constraint $A_i x < b_i$ such that $A_i x^{(k)} \geq b_i$. The hyperplane $A_i x = b_i$ slices E_k through the centre; the half that does not contain P can be discarded.
2. Find the minimum volume ellipsoid E_{k+1} containing the half of E_k containing P, and let $x^{(k+1)}$ be the centre of E_{k+1}.
3. If $k \geq K$, stop: $P_\varepsilon = \emptyset$.
4. Repeat from 1.

An ellipsoid E_k with centre $x^{(k)}$ is described by $\{x \in \mathbb{R}^n \mid (x - x^{(k)})^\top D_k^{-1}(x - x^{(k)}) \leq 1\}$ where D_k is an $n \times n$ positive definite symmetric matrix. We compute $x^{(k+1)}$ and D_{k+1} as follows:

$$\begin{aligned}
\Gamma_k &= \frac{D_k A_i}{\sqrt{{A_i}^\top D_k A_i}} \\
x^{(k+1)} &= x^{(k)} + \frac{\Gamma_k}{n+1} \\
D_{k+1} &= \frac{n^2}{n^2-1}\left(D_k - \frac{2\Gamma_k {\Gamma_k}^\top}{n+1}\right).
\end{aligned}$$

It turns out that E_{k+1} defined by the matrix D_{k+1} contains the half-ellipsoid of E_k containing P, and furthermore

$$\mathrm{Vol}(E_{k+1}) \leq e^{-\frac{1}{2(n+1)}} \mathrm{Vol}(E_k). \tag{31}$$

The volume of E_K is necessarily smaller than v: by (31) we have

$$\mathrm{Vol}(E_K) \leq e^{-\frac{K}{2(n+1)}} \mathrm{Vol}(E_0) \leq V e^{-\frac{K}{2(n+1)}} \leq V e^{-\ln(V/v)} = v,$$

hence by the assumption on v we must have $P_\varepsilon = \emptyset$ if $k \geq K$. Since it can be shown that K is polynomial in the size of the problem instance (which depends on the number of variables, the number of constraints as well as the bits needed to store A and b), the Ellipsoid algorithm is a polynomial algorithm which can be used to solve linear programming problems. One last point to be raised is that in Eq. (31) we rely on an irrational computation, which is never precise on a computer. It can be shown that computations carried out to a certain amount of accuracy can still decide whether P_ε is empty or not.

6.2 Karmarkar's algorithm

Karmarkar's algorithm is designed to solve a particular type of linear optimization problem; namely, one where the objective function $c^\top x$ is known *a priori* to have optimal objective function value $f^* = c^\top x^* =$

0, and whose constraints are $Ax = 0, x \geq 0$ and $e^\top x = 1$; it is assumed that $Ae = 0$, so that $\bar{x} = e/n$ is an initial feasible point (where $e = (1, \ldots, 1)^\top$ and $e/n = (1/n, \ldots, 1/n)^\top$). It can be shown that any LP can be polynomially reduced to this special form. Let $X = \text{diag}(\bar{x})$ and $B = \begin{pmatrix} AX \\ e \end{pmatrix}$. The algorithm is as follows:

1. Project Xc into the null space of B: $c^* = (I - B^\top(BB^\top)^{-1}B)Xc$.
2. Normalize the descent direction: $d = \gamma \frac{c^*}{||c^*||}$. If $c^* = 0$ terminate with optimal solution x^*.
3. Move in projected space: $y = e/n - sd$ where s is a fixed step size.
4. Project back into x-space: $\bar{x} \leftarrow \frac{Xy}{e^\top Xy}$.
5. Repeat from 1.

Taking $\gamma = \frac{1}{\sqrt{n(n-1)}}$ and $s = \frac{1}{4}$ guarantees that the algorithm has polynomial complexity.

Let us look at the first iteration of Karmarkar's algorithm in more detail. The initial point $\bar{x}$ is taken to be the feasible point $\bar{x} = e/n$; notice e/n is also the centre of the simplex $e^\top x = 1, x \geq 0$. Let $S_r = \{x \in \mathbb{R}^n \mid ||x - x^*|| \leq r \wedge e^\top x = 1 \wedge x \geq 0\}$ be the largest sphere that can be inscribed in the simplex, and let S_R be the smallest concentric sphere that circumscribes the simplex. It can be shown that $R/r = n - 1$. Let x_r be the minimizer of $c^\top x$ on $F_r = S_r \cap \{x \mid Ax = 0\}$ and x_R the minimizer on $F_R = S_R \cap \{x \mid Ax = 0\}$; let $f_r = c^\top x_r$, $f_R = c^\top x_R$ and $\bar{f} = c^\top \bar{x}$. By the linearity of the objective function, we have $\frac{\bar{f} - f_R}{\bar{f} - f_r} = n - 1$. Since F_R contains the feasible region of the problem, $f_R \leq f^* = 0$, and so $(n-1)\bar{f} - (n-1)f_r = \bar{f} - f_R \geq \bar{f} - f^* = \bar{f}$, whence

$$f_r \leq \frac{n-2}{n-1}\bar{f} < e^{-\frac{1}{n-1}}\bar{f}. \tag{32}$$

Finally, we update $\bar{x}$ with x_r and repeat the process. If this reduction in objective function value could be attained at each iteration, $O(nL)$ iterations would be required to get within 2^{-L} tolerance from $f^* = 0$, and the algorithm would be polynomial. The only trouble lies in the fact that after the first iteration the updated current point x_r is not the centre of the simplex anymore. Thus, we use a projective transformation (step 4 of the algorithm) to reshape the problem so that the updated current point is again at the centre of the simplex. The linearity of the objective function is not preserved by the transformation, so a linear approximation is used (step 1 of the algorithm). Karmarkar showed that Eq. (32) holds for the modified objective function too, thus proving the polynomiality of the algorithm.

6.3 The Log-Barrier function

Gill et al. showed in 1985 that Karmarkar's algorithm is a special case of a log-barrier function applied to an LP problem, and also that various types of log-barrier methods yielded polynomial algorithms to solve LP problems. We consider the LP problem (6) in standard form, and reformulate it as follows:

$$\left.\begin{array}{rl} \min_x & c^\top x - \beta \sum\limits_{j=1}^{n} \ln x_j \\ \text{s.t.} & Ax = b \\ & x > 0, \end{array}\right\} \tag{33}$$

where $\beta > 0$ is a parameter. Notice that since $-\ln x_j \to \infty$ as $x_j \to 0^+$, minimizing the objective function of (33) automatically implies that $x > 0$. Furthermore, as β decreases towards 0, (33) describes a continuous family of optimization problems which converge to (6). This suggests a solution method based on solving a discrete sequence of convex problems

$$\left.\begin{array}{rl} \min_{x(\beta)} & c^\top x(\beta) - \beta \sum_{j=1}^{n} \ln x_j(\beta) \\ \text{s.t.} & Ax(\beta) = b \end{array}\right\} \tag{34}$$

for $\beta = \beta_1, \ldots, \beta_k, \ldots$, whose solutions $x^*(\beta_k) \to x^*$ as $k \to \infty$, where x^* is the solution of (6). The set $\{x^*(\beta) \mid \beta > 0\}$ is a path in $\mathbb{R}^n$ called the *central path.*

6.3.1 Primal-Dual feasible points

We show that each point $x(\beta)$ on the central path yields a dual feasible point. The Lagrangian $L_1(x, \lambda, \nu)$ for problem (6) (where λ are the dual variables for the constraints $-x \leq 0$ and ν for $Ax = b$) is:

$$L_1(x, \lambda, \nu) = c^\top x - \sum_{j=1}^{n} \lambda_j x_j + \nu(Ax - b), \tag{35}$$

while the Lagrangian $L_2(x, \nu)$ for problem (34) is:

$$L_2(x, \nu) = c^\top x(\beta) - \beta \sum_{j=1}^{n} \ln x_j(\beta) + \nu(Ax - b).$$

Deriving the stationarity KKT condition (21) from L_1 we get:

$$\forall j \leq n \ (c_j - \lambda_j + \nu A^j = 0),$$

where A^j is the j-th column of A. From L_2 we get:

$$\forall j \leq n \ (c_j - \frac{\beta}{x_j} + \nu A^j = 0).$$

Therefore, by letting

$$\lambda_j = \frac{\beta}{x_j} \tag{36}$$

for all $j \leq n$, we can show that each point $x(\beta)$ on the central path gives rise to a dual feasible point (λ, ν) for the dual of (6):

$$\left.\begin{array}{rrcl} \max\limits_{\nu} & b^\top \nu & & \\ \text{s.t.} & A^\top \nu + \lambda & = & c \\ & \lambda & \geq & 0. \end{array}\right\} \tag{37}$$

Notice that the dual problem (37) has been derived from (27) by setting $\nu = -\mu$ and using λ as slack variables for the inequalities in (27). Since λ, ν also depend on β, we indicate them by $\lambda(\beta), \nu(\beta)$. That $(\lambda(\beta), \nu(\beta))$ is dual feasible in (37) follows because $(x(\beta), \lambda(\beta), \nu(\beta))$ satisfies (21) as shown above. Now, notice that by definition $\lambda_j(\beta) = \frac{\beta}{x_j(\beta)}$ implies

$$\forall j \leq n \ (\lambda_j(\beta) x_j(\beta) = \beta) \tag{38}$$

which means that as β converges to 0, the conditions (38) above imply the KKT slack complementarity conditions (23). Since $\lambda(\beta) \geq 0$ by definition (36), we have that $(x^*, \lambda^*, \nu^*) = (x(0), \lambda(0), \mu(0))$ is an optimal primal-dual pair solving (6) and (37).

6.3.2 Optimal partitions

In a linear problem there may be more than one optimal solution. If that is the case, all optimal solutions are on one face of the polyhedron. When such a case arises in the simplex method, more than one bfs may be optimal. The analysis of the IPM provides a unique characterization of the optimal solutions even when there is no unique optimal solution. We show that the central path converges to a strictly complementary optimal solution, i.e. an optimal solution for which $(x^*)^\top \lambda^* = 0$ and $x^* + \lambda^* > 0$. These solutions are used to construct the *optimal partition*, i.e. a partition of the n vector components in two index sets B, N such that:

$$\begin{aligned} B &= \{j \le n \mid x_i^* > 0\} \\ N &= \{j \le n \mid \lambda_i^* > 0\}. \end{aligned}$$

The partition obtained in this way is unique, and does not depend on the strictly complementary optimal solution used to define it. Optimal partitions provide a unique and well-defined characterization of optimal faces.

In the remainder of this section we write $x = x(\beta)$ and $\lambda = \lambda(\beta)$ to simplify notation. The proof of the following theorem was taken from [JRTV93].

6.1 Theorem (Goldman-Tucker, 1956)
With the same notation of this section, (x^*, λ^*), the limiting point of the primal-dual central path $(x(\beta), \lambda(\beta))$, is a strictly complementary primal-dual optimal solution for (6).

Proof. By (38) we have that $x^\top \lambda = n\beta$, thus $(x^*)^\top \lambda^*$ converges to 0 as $\beta \to 0$. Now, both x^* and x are primal feasible, so $Ax^* = Ax = b$, hence $A(x^* - x) = 0$. Furthermore, since both λ^* and λ are dual feasible, we have $\nu^* A + \lambda^* = \nu A + \lambda = c$, i.e. $(\nu - \nu^*)A = \lambda^* - \lambda$. In other words, $x^* - x$ is in the null space of A and $\lambda^* - \lambda$ is in the range of $A^\top$: thus, the two vectors are orthogonal. Hence we have:

$$0 = (x^* - x)^\top (\lambda^* - \lambda) = (x^*)^\top \lambda^* + x^\top \lambda - (x^*)^\top \lambda - x^\top \lambda^*,$$

and thus

$$(x^*)^\top \lambda + x^\top \lambda^* = n\beta.$$

Dividing throughout by $\beta = x_j \lambda_j$ we obtain:

$$\sum_{j=1}^{n} \left(\frac{x_j^*}{x_j} + \frac{\lambda_j^*}{\lambda_j} \right) = n.$$

Since

$$\lim_{\beta \to 0} \frac{x_j^*}{x_j} = \begin{cases} 1 & \text{if } x_i^* > 0 \\ 0 & \text{otherwise} \end{cases}$$

and similarly for the λ component, for each $j \le n$ exactly one of the two components of the pair (x_j^*, λ_j^*) is zero and the other is positive. □

6.3.3 A simple IPM for LP

The prototypical IPM for LP is as follows:

1. Consider an initial point $x(\beta_0)$ feasible in (34), a parameter $\alpha < 1$ and a tolerance $\varepsilon > 0$. Let $k = 0$.
2. Solve (34) with initial point $x(\beta_k)$, to get a solution x^*.

3. If $n\beta_k < \varepsilon$, stop with solution x^*.
4. Update $\beta_{k+1} = \alpha\beta_k$, $x(\beta_{k+1}) = x^*$ and $k \leftarrow k + 1$.
5. Go to step 2.

By (35), $L_1(x, \lambda, \nu) = c^\top x - n\beta_k$, which means that the duality gap is $n\beta_k$. This implies that $x(\beta_k)$ is never more than $n\beta_k$-suboptimal. This is the basis of the termination condition in step (3). Each subproblem in step 2 can be solved by using Newton's method.

6.3.4 The Newton step

In general, the Newton descent direction d for an unconstrained problem $\min f(x)$ at a point $\bar{x}$ is given by:

$$d = -(\nabla^2 f(\bar{x}))^{-1}\nabla f(\bar{x}). \tag{39}$$

If $\nabla^2 f(\bar{x})$ is positive definite, we obtain

$$(\nabla f(\bar{x}))^\top d = -(\nabla f(\bar{x}))^\top (\nabla^2 f(\bar{x}))^{-1}\nabla f(\bar{x}) < 0,$$

so d is a descent direction. In this case, we need to find a feasible descent direction such that $Ad = 0$. Thus, we need to solve the system

$$\begin{pmatrix} \nabla^2 f(\bar{x}) & A^\top \\ A & 0 \end{pmatrix} \begin{pmatrix} d \\ \nu^\top \end{pmatrix} = \begin{pmatrix} -\nabla f(\bar{x}) \\ 0 \end{pmatrix},$$

for (d, ν), where ν are the dual variables associated with the equality constraints $Ax = b$. Step 4 in the IPM algorithm of Section 6.3.3 becomes $x(\beta_{k+1}) = x(\beta_k) + \gamma d$, where γ is the result of a line search, for example

$$\gamma = \mathrm{argmin}_{s \geq 0} f(\bar{x} + sd). \tag{40}$$

Notice that feasibility with respect to $Ax = b$ is automatically enforced because $\bar{x}$ is feasible and d is a feasible direction.

7 Conclusion

In this didactical paper we have discussed some of the very basic theoretical notions and methods in linear programming. Starting from some geometrical notions about convexity and polyhedra, we explained the primal simplex algorithm. We then derived the dual of a linear program via the Lagrangian function; next we briefly examined some well-known variants of the simplex algorithm. Finally, we gave some sketchy notes about the Ellipsoid algorithm, Karmarkar's algorithm, and Interior Point algorithms based on the log-barrier function.

References

[BSS93] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms.* Wiley, Chichester, second edition, 1993.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, 2004.

[Dan63] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963.

[Fis99] M. Fischetti. *Lezioni di Ricerca Operativa (in Italian).* Edizioni Libreria Progetto, Padova, 1999.

[Fle91] R. Fletcher. *Practical Methods of Optimization.* Wiley, Chichester, second edition, 1991.

[JRTV93] B. Jansen, C. Roos, T. Terlaky, and J.-Ph. Vial. Interior-point methodology for linear programming: duality, sensitivity analysis and computational aspects. Technical Report 28, 1993.

[KV00] B. Korte and J. Vygen. *Combinatorial Optimization, Theory and Algorithms.* Springer-Verlag, Berlin, 2000.

[Lee04] J. Lee. *A First Course in Combinatorial Optimization.* Cambridge University Press, Cambridge, 2004.

[NW88] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York, 1988.

[PS98] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover, New York, 1998.

[Pug02] C.C. Pugh. *Real Mathematical Analysis.* Springer-Verlag, Berlin, 2002.

[Tah92] H.A. Taha. *Operations Research: An Introduction.* MacMillan, New York, 1992.