Chapter 3

Linear Programming

Linear programs (LP) play an important role in the theory and practice of optimization problems. Many COPs can directly be formulated as LPs. Furthermore, LPs are invaluable for the design and analysis of approximation algorithms. Generally speaking, LPs are COPs with linear objective function and linear constraints, where the variables are defined on a continous domain. We will be more specific below.

3.1 Introduction

We begin our treatment of linear programming with an example of a transportation problem to illustrate how LPs can be used to formulate optimization problems.

Example 3.1. There are two brickworks w_1, w_2 and three construction sites s_1, s_2, s_3. The works produce $b_1 = 60$ and $b_2 = 30$ tons of bricks per day. The sites require $c_1 = 30$, $c_2 = 20$ and $c_3 = 40$ tons of bricks per day. The transportation costs t_{ij} per ton from work w_i to site s_j are given in the following table:

t_{ij}	s_1	s_2	s_3
w_1	40	75	50
w_2	20	50	40

Which work delivers which site in order to minimize the total transportation cost? Let us write the problem as a mathematical program. We use variables x_{ij} that tell us how much we deliver from work w_i to site s_j.

$$\begin{aligned}
\text{minimize} \quad & 40x_{11} + 75x_{12} + 50x_{13} + 20x_{21} + 50x_{22} + 40x_{23} \\
\text{subject to} \quad & x_{11} + x_{12} + x_{13} \leq 60 \\
& x_{21} + x_{22} + x_{23} \leq 30 \\
& x_{11} + x_{21} = 30 \\
& x_{12} + x_{21} = 20 \\
& x_{13} + x_{23} = 40 \\
& x_{ij} \geq 0 \quad i = 1, 2,\ j = 1, 2, 3.
\end{aligned}$$

How do we find the best x_{ij}?

The general Linear Programming task is given in Problem 3.1.

As a shorthand we shall frequently write $\max\{c^\top x : Ax \leq b\}$. We can assume that we deal with a maximization problem without loss of generality because we can treat a minimization problem if we replace c with $-c$.

Problem 3.1 LINEAR PROGRAMMING

Instance. Matrix $A \in \mathbb{R}^{m\times n}$, vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

Task. Solve the problem

$$\begin{aligned} \text{maximze} \quad & c^\top x, \\ \text{subject to} \quad & Ax \leq b, \\ & x \in \mathbb{R}^n. \end{aligned}$$

That means answer one of the following questions.

(1) Find a vector $x \in \mathbb{R}^n$ such that $Ax \leq b$ and $\text{val}(x) = c^\top x$ is maximum, or

(2) decide that the set $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is empty, or

(3) decide that for all $\alpha \in \mathbb{R}$ there is an $x \in \mathbb{R}^n$ with $Ax \leq b$ and $c^\top x > \alpha$.

The function $\text{val}(x) = c^\top x$ is the *objective function*. A feasible x^* which maximizes val is an *optimum solution* and the value $z^* = \text{val}(x^*)$ is called *optimum value*. Any $x \in \mathbb{R}^n$ that satisfies $Ax \leq b$ is called *feasible*. The set $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is called the *feasible region*, i.e., the set of *feasible solutions*. If P is empty, then the problem is *infeasible*. If for every $\alpha \in \mathbb{R}$, there is a feasible x such that $c^\top x > \alpha$ then the problem is *unbounded*. This simply means that the maximum of the objective function does not exist.

3.2 Polyhedra

Consider the vector space $\mathbb{R}^n$. A *(linear) subspace* S of R^n is a subset of $\mathbb{R}^n$ closed under vector addition and scalar multiplication. Equivalently, S is the set of all points in $\mathbb{R}^n$ that satisfy a set of homogeneous linear equations:

$$S = \{x \in \mathbb{R}^n : Ax = 0\},$$

for some matrix $A \in \mathbb{R}^{n\times m}$. The *dimension* $\dim(S)$ is equal to the maximum number of linear independent vectors in S, i.e., $\dim(S) = n - \text{rank}(A)$. Here $\text{rank}(A)$ denotes the number of linear independent rows of A. An *affine subspace* S_b of $\mathbb{R}^n$ is the set of all points that satisfy a set of inhomogeneous linear equations:

$$S_b = \{x \in \mathbb{R}^n : Ax = b\}.$$

We have $\dim(S_b) = \dim(S)$. The *dimension* $\dim(X)$ of any subset $X \subseteq \mathbb{R}^n$ is the smallest dimension of any affine subspace which contains it.

An affine subspace of $\mathbb{R}^n$ of dimension $n-1$ is called *hyperplane*, i.e., alternatively

$$H = \{x \in \mathbb{R}^n : a^\top x = b\},$$

for some vector $a \in \mathbb{R}^n, a \neq 0$ and scalar b. A hyperplane defines two (closed) *halfspaces*

$$H^+ = \{x \in \mathbb{R}^n : a^\top x \geq b\},$$
$$H^- = \{x \in \mathbb{R}^n : a^\top x \leq b\}.$$

As a halfspace is a convex set, the intersection of halfspaces is also convex.

A *polyhedron* in $\mathbb{R}^n$ is a set

$$P = \{x \in \mathbb{R}^n : Ax \leq b\}$$

for some matrix $A \in \mathbb{R}^{m \times n}$ and some vector $b \in \mathbb{R}^m$. A bounded polyhedron is called *polytope*.

Let $P = \{x : Ax \leq b\}$ be a non-empty polyhedron with dimension d. Let c be a vector for which $\delta := \max\{c^\top x : x \in P\} < \infty$, then

$$H_c = \{x : c^\top x = \delta\}$$

is called *supporting hyperplane* of P. A *face* of P is the intersection of P with a supporting hyperplane of P. Three types of faces are particular important, see Figure 3.1:

(1) A *facet* is a face of dimension $d - 1$,

(2) a *vertex* is a face of dimension zero (a point), and

(3) an *edge* is a face of dimension one (a line segment).

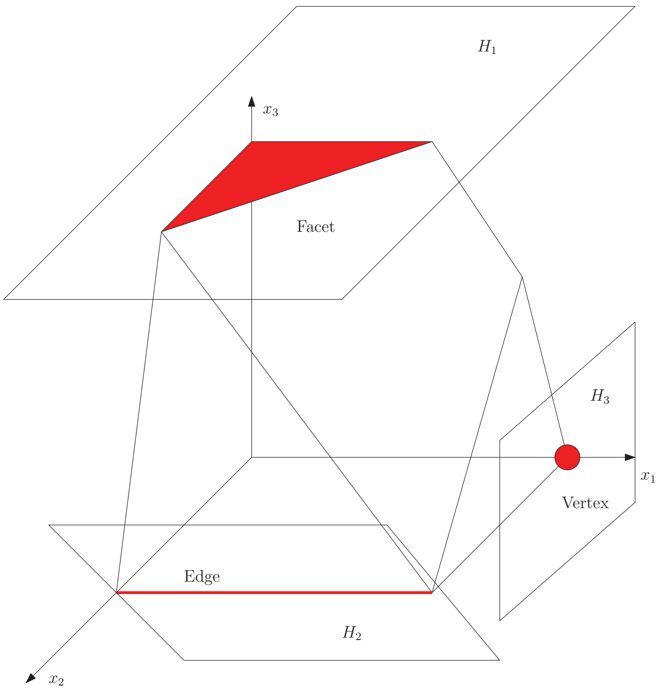

Figure 3.1: Facet, vertex, and edge.

The following lemma essentially states that a set $F \subseteq P$ is a face of a polyhedron P if and only if some of the inequalities of $Ax \leq b$ are satisfied with equality for all elements of F.

Lemma 3.2. *Let $P = \{x : Ax \leq b\}$ be a polyhedron and $F \subseteq P$. Then the following statements are equivalent:*

(1) F is a face of P.

(2) There is a vector c with $\delta := \max\{c^\top x : x \in P\} < \infty$ and $F = \{x \in P : c^\top x = \delta\}$.

(3) $F = \{x \in P : A'x = b'\} \neq \emptyset$ for some subsystem $A'x \leq b'$ of $Ax \leq b$.

As important corollaries we have:

Corollary 3.3. *If $\max\{c^\top x : x \in P\} < \infty$ for a non-empty polyhedron P and a vector c, then the set of points where the maximum is attained is a face of P.*

Corollary 3.4. *Let P be a polyhedron and F a face of P. Then F is again a polyhedron. Furthermore, a set $F' \subseteq F$ is a face of P if and only if it is a face of F.*

A important class of faces are *minimal faces*, i.e., faces that do not contain any other face. For these we have:

Lemma 3.5. *Let $P = \{x : Ax \leq b\}$ be a polyhedron. A non-empty set $F \subseteq P$ is a minimal face of P if and only if $F = \{x \in \mathbb{R}^n : A'x = b'\}$ for some subsystem of $Ax \leq b$.*

Corollary 3.3 and Lemma 3.5 already imply that LINEAR PROGRAMMING can be solved by solving the linear *equation* system $A'x = b'$ for each subsystem $A'x \leq b'$. This approach obviously yields an exponential time algorithm. An algorithm which is more practicable (although also exponential in the worst case) is the SIMPLEX algorithm. The algorithm is based on the following important consequence of Lemma 3.5.

Corollary 3.6. *Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a polyhedron. Then all minimal faces of P have dimension $n - \operatorname{rank}(A)$. The minimal faces of polytopes are vertices.*

Thus, it suffices to search an optimum solution among the *vertices* of the polyhedron. This is what the SIMPLEX algorithm is doing. We do not explain the algorithm in detail here, but it works as follows. Provided that the polyhedron is not empty, it finds an initial vertex. If the current vertex is not optimal, find another vertex with strictly larger objective value (pivot rule). Iterate until an optimal vertex is found or the polyhedron can be shown to be unbounded. See Figure 3.2.

The algorithm terminates after at most $\binom{m}{n}$ iterations (which is not polynomial). It was conjectured that SIMPLEX is polynomial until Klee and Minty gave an example where the algorithm (with Bland's pivot rule) uses 2^n iterations on an LP with n variables and $2n$ constraints. It is not known if there is a pivot rule that leads to polynomial running time. Nonetheless, SIMPLEX with Bland's pivot rule is frequently observed to terminate after few iterations when run on "practical instances".

However, there are algorithms, e.g., the ELLIPSOID method and KARMAKAR's algorithm that solve LINEAR PROGRAMMING in polynomial time. But these algorithms are mainly of interest from a theoretical point of view. We conclude with the statement that one can solve LINEAR PROGRAMMING in polynomial time with "black box" algorithms.

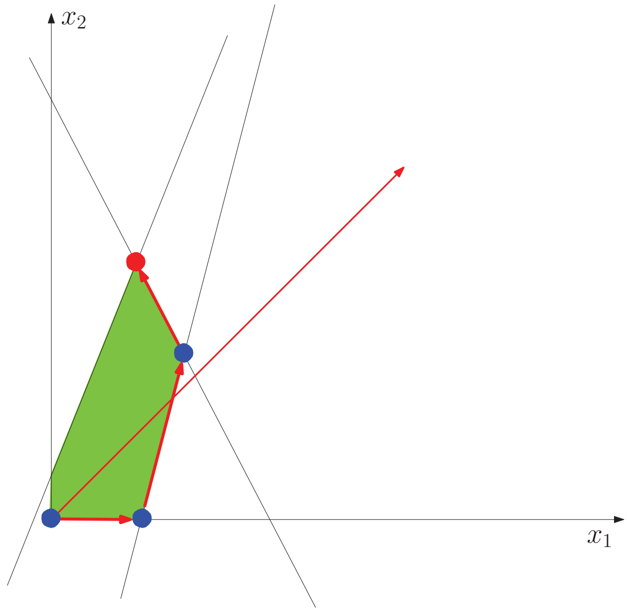

Figure 3.2: A SIMPLEX path.

3.3 Duality

Intuition behind Duality

Consider the following LP, which is illustrated in Figure 3.3

$$\begin{aligned} \text{maximize} \quad & x_1 + x_2 && (3.1)\\ \text{subject to} \quad & 4x_1 - x_2 \leq 8 && (3.2)\\ & 2x_1 + x_2 \leq 10 && (3.3)\\ & -5x_1 + 2x_2 \leq 2 && (3.4)\\ & -x_1 \leq 0 && (3.5)\\ & -x_2 \leq 0 && (3.6) \end{aligned}$$

and notice that this LP is in the maximization form

$$\max\{c^\top x : Ax \leq b\}.$$

Because we are dealing with a maximization problem, every feasible solution x provides the *lower bound* $c^\top x$ on the value $c^\top x^*$ of the optimum solution x^*, i.e., we know $c^\top x \leq c^\top x^*$.

Can we also obtain *upper bounds* on $c^\top x^*$? For any feasible solution x, the constraints (3.2)–(3.6) are satisfied. Now compare the objective function (3.1) with the constraint (3.3) *coefficient-by-coefficient* (where we remember that $x_1, x_2 \geq 0$ in this example):

$$\begin{array}{ccccc} 1 \cdot x_1 & + & 1 \cdot x_2 & & \\ \leq & & \leq & & \\ 2 \cdot x_1 & + & 1 \cdot x_2 & \leq 10 \end{array}$$

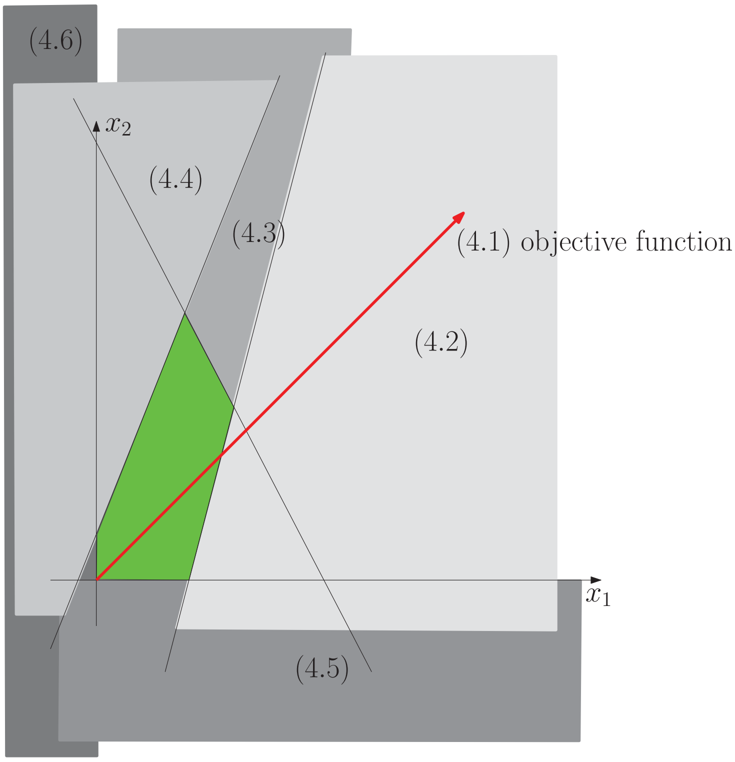

Figure 3.3: An LP.

Thus for *every* feasible solution x we have the upper bound $x_1 + x_2 \leq 10$, i.e., the optimum value can be at most 10. Can we improve on this? We could try $\frac{7}{9} \cdot (3.3) + \frac{1}{9} \cdot (3.4)$:

$$\begin{array}{ccccc} 1 \cdot x_1 & + & 1 \cdot x_2 & & \\ \text{\rotatebox{90}{$\leq$}} & & \text{\rotatebox{90}{$\leq$}} & & \\ (\frac{7}{9} \cdot 2 + \frac{1}{9} \cdot (-5))x_1 & + & (\frac{7}{9} \cdot 1 + \frac{1}{9} \cdot 2)x_2 & \leq \frac{7}{9} \cdot 10 + \frac{1}{9} \cdot 2 = \frac{72}{9} = 8 \end{array}$$

Hence we have $x_1 + x_2 \leq 8$ for every feasible x and thus an upper bound of 8 on the optimum value. If we look closely, our choices 7/9 and 1/9 give $\frac{7}{9} \cdot 2 + \frac{1}{9} \cdot (-5) = 1$ and $\frac{7}{9} \cdot 1 + \frac{1}{9} \cdot 2 = 1$, i.e., we have combined the coefficients of the objective function $c^\top x$ with *equality*. This is also the best bound this approach can give here.

This suggests the following genear approach for obtining upper bounds on the optimal value. Combine the constraints with non-negative multipliers $y = (y_1, y_2, y_3, y_4, y_5)$ such that each coefficient in the result equals the corresponding coefficient in the objective function, i.e., we want $y^\top A = c^\top$. We associate y_1 with (3.2), y_2 with (3.3), y_3 with (3.4), y_4 with (3.5), and y_5 with (3.6). Notice that the y_i must be non-negative because we are multiplying an *inequality* of the system $Ax \leq b$, i.e., if a multiplier y_i were negative we change the corresponding inequality from "$\leq$" to "$\geq$". Now $y_1(3.2) + y_2(3.3) + y_3(3.4) + y_4(3.5) + y_5(3.6)$ evaluates to

$$y_1(4x_1 - 1x_2) + y_2(2x_1 + x_2) + y_3(-5x_1 + 2x_2) + y_4(-x_1) + y_5(-x_2) \\ \leq y_1 8 + y_2 10 + y_3 2 + y_4 0 + y_5 0,$$

where rearranging yields

$$(4y_1 + 2y_2 - 5y_3 - y_4)x_1 + (-y_1 + y_2 + 2y_3 - y_5)x_2 \leq 8y_1 + 10y_2 + 2y_3 + 0y_4 + 0y_5$$

and want to find values for $y_1, y_2, y_3, y_4, y_5 \geq 0$ that satisfy:

$$\begin{array}{ccccc} 1 \cdot x_1 & + & 1 \cdot x_2 & & \\ \| & & \| & & \\ (4y_1 + 2y_2 - 5y_3 - y_4)x_1 & + & (-y_1 + y_2 + 2y_3 - y_5)x_2 & \leq & 8y_1 + 10y_2 + 2y_3 + 0y_4 + 0y_5 \end{array}$$

Of course, we are interested in the best choice for $y = (y_1, y_2, y_3, y_4, y_5) \geq 0$ the approach can give. This means that we want to minimize the upper bound $8y_1+10y_2+2y_3+0y_4+0y_5$. We simply write down this task as a mathematical program, which turns out to be an LP.

$$\begin{aligned} \text{minimize} \quad & 8y_1 + 10y_2 + 2y_3 + 0y_4 + 0y_5 && (3.7) \\ \text{subject to} \quad & 4y_1 + 2y_2 - 5y_3 - y_4 = 1 && (3.8) \\ & -y_1 + y_2 + 2y_3 - y_5 = 1 && (3.9) \\ & y_1, y_2, y_3, y_4, y_5 \geq 0 && (3.10) \end{aligned}$$

Further note that the new objective function is the right hand side $(8, 10, 2, 0, 0)^\top$ of the original LP and that the new right hand side is the objective function $(1, 1)^\top$ of the original LP. Thus the above LP is of the form

$$\min\{y^\top b : y^\top A = c^\top, y \geq 0\}.$$

Notice that there is a feasible solution $x = (2, 6)^\top$ for the original LP that gives $c^\top x = 8$. Further note that the multipliers $y = (0, 7/9, 1/9, 0, 0)^\top$ yield $y^\top b = 8$, i.e.,

$$c^\top x = y^\top b.$$

Hence we have a certificate that the solution $x = (2, 6)$ is indeed optimal (because we have a matching upper bound). Not surprisingly this is no exception but the principal statement of the *strong duality* theorem.

Weak and Strong Duality

Given an LP

$$P = \max\{c^\top x : Ax \leq b\}$$

called *primal*, we define the *dual*

$$D = \min\{y^\top b : y^\top A = c^\top, y \geq 0\}.$$

Lemma 3.7. *The dual of the dual of an LP is (equivalent to) the original LP.*

Now we can say that the LPs P and D are dual to each other or a *primal-dual pair.* The following forms of primal-dual pairs are standard:

$$\begin{aligned} \max\{c^\top x : Ax \leq b\} &\sim \min\{y^\top b : y^\top A = c^\top, y \geq 0\} \\ \max\{c^\top x : Ax \leq b, x \geq 0\} &\sim \min\{y^\top b : y^\top A \geq c^\top, y \geq 0\} \\ \max\{c^\top x : Ax = b, x \geq 0\} &\sim \min\{y^\top b : y^\top A \geq c^\top\} \end{aligned}$$

The following lemma is called *weak duality.*

Lemma 3.8 (Weak Duality)**.** *Let x and y be respective feasible solutions of the primal-dual pair $P = \max\{c^\top x : Ax \leq b\}$ and $D = \min\{y^\top b : y^\top A = c^\top, y \geq 0\}$. Then $c^\top x \leq y^\top b$.*

Proof. $c^\top x = (y^\top A)x = y^\top (Ax) \le y^\top b$. □

The following *strong duality* theorem is the most important result in LP theory and the basis for a lot of algorithms for COPs.

Theorem 3.9 (Strong Duality). *For any primal-dual pair* $P = \max\{c^\top x : Ax \le b\}$ *and* $D = \min\{y^\top b : y^\top A = c^\top, y \ge 0\}$ *we have:*

(1) If P *and* D *have respective optimum solutions* x *and* y*, say, then*

$$c^\top x = y^\top b.$$

(2) If P *is unbounded, then* D *is infeasible.*

(3) If P *is infeasible, then* D *is infeasible or unbounded.*

Before we prove the theorem, we establish the *fundamental theorem of linear inequalities*. The heart of the proof actually gives a basic version of the Simplex algorithm. The result also implies Farkas' Lemma.

Theorem 3.10. *Let* $a_1, \ldots, a_m, b$ *be vectors in* n*-dimensional space. Then*

either (I): $b = \sum_{i=1}^m a_i \lambda_i$ *with* $\lambda_i \ge 0$ *for* $i = 1, \ldots, m$,

or (II): *there is a hyperplane* $\{x : c^\top x = 0\}$*, containing* $t-1$ *linearly independent vectors from* $a_1, \ldots, a_m$ *such that* $c^\top b < 0$ *and* $c^\top a_1, \ldots, c^\top a_m \ge 0$*, where* $t = \operatorname{rank}\{a_1, \ldots, a_m, b\}$.

Proof. We may assume that $a_1, \ldots, a_m$ span the n-dimensional space. Clearly, (I) and (II) exclude each other as we would otherwise have the contradiction

$$0 > c^\top b = \lambda_1 c^\top a_1 + \cdots + \lambda_m c^\top a_m \ge 0.$$

To see that at least one of (I) and (II) holds, choose linearly independent $a_{i_1}, \ldots, a_{i_n}$ from $a_1, \ldots, a_m$ and set $B = \{a_{i_1}, \ldots, a_{i_n}\}$. Next apply the following iteration:

(i) Write $b = \lambda_{i_1} a_{i_1} + \cdots + \lambda_{i_n} a_{i_n}$. If $\lambda_{i_1}, \ldots, \lambda_{i_n} \ge 0$ we are in case (I).

(ii) Otherwise, choose the smallest h among $i_1, \ldots, i_n$ with $\lambda_h < 0$. Let $\{x : c^\top x = 0\}$ be the hyperplane spanned by $B - \{a_h\}$. We normalize c so that $c^\top a_h = 1$. (Hence $c^\top b = \lambda_h < 0$.)

(iii) If $c^\top a_1, \ldots, c^\top a_m \ge 0$ we are in case (II).

(iv) Otherwise, choose the smallest s such that $c^\top a_s < 0$. Then replace B by $(B - \{a_h\}) \cup \{a_s\}$. Restart the iteration anew.

We are finished if we have shown that this process terminates. Let B_k denote the set B as it is in the k-th iteration. If the process does not terminate, then $B_k = B_\ell$ for some $k < \ell$ (as there are only finitely many choices for B). Let r be the highest index for which a_r has been removed from B at the end of one of the iterations $k, k+1, \ldots, \ell-1$, say in iteration p. As $B_k = B_\ell$, we know that a_r also has been added to B in some iteration q with $k \le q \le \ell$. So

$$B_p \cap \{a_{r+1}, \ldots, a_m\} = B_q \cap \{a_{r+1}, \ldots, a_m\}.$$

Let $B_p = \{a_{i_1}, \dots, a_{i_n}\}$, $b = \lambda_{i_1} a_{i_1} + \dots + \lambda_{i_n} a_{i_n}$, and let d be the vector c found in iteration q. Then we have the contradiction

$$0 > d^\top b = d^\top(\lambda_{i_1} a_{i_1} + \dots + \lambda_{i_n} a_{i_n}) = \lambda_{i_1} d^\top a_{i_1} + \dots + \lambda_{i_n} d^\top a_{i_n} + \dots + \lambda_{i_n} d^\top a_{i_n} > 0,$$

where the second inequality follows from: If $i_j < r$ then $\lambda_{i_j} \geq 0, d^\top a_{i_j} \geq 0$, if $i_j = r$ then $\lambda_{i_j} < 0, d^\top a_{i_j} < 0$, and if $i_j > r$ then $d^\top a_{i_j} = 0$. □

Lemma 3.11 (Farkas' Lemma)**.** *There is a vector*

$$\begin{cases} x & \text{with } Ax \leq b \\ x \geq 0 & \text{with } Ax \leq b \\ x \geq 0 & \text{with } Ax = b \end{cases} \quad \text{if and only if } y^\top b \geq 0 \text{ for all} \quad \begin{cases} y \geq 0 & \text{with } y^\top A = 0 \\ y \geq 0 & \text{with } y^\top A \geq 0 \\ y & \text{with } y^\top A \geq 0 \end{cases}.$$

Proof. We first show the case $x \geq 0$ with $Ax = b$ if and only if $y^\top b \geq 0$ for each y with $y^\top A \geq 0$.

Necessity is clear since $y^\top b = y^\top(Ax) \geq 0$ for all x and y with $x \geq 0$, $y^\top A \geq 0$, and $Ax = b$. For suffience, assume that there is no $x \geq 0$ with $Ax = b$. Then, by Theorem 3.10 and denoting $a_1, \dots, a_m$ be the colums of A, there is a hyperplane $\{x : y^\top x = 0\}$ with $y^\top b < 0$ for some y with $y^\top A \geq 0$.

For the case x with $Ax \leq b$ if and only if $y^\top b \geq 0$ for each $y \geq 0$ with $y^\top A = 0$ consider $A' = [I, A, -A]$. Observe that $Ax \leq b$ has a solution x if and only if $A'x' = b$ has a solution $x' \geq 0$. Now apply what we have just proved.

For the case $x \geq 0$ with $Ax \leq b$ if and only if $y^\top b \geq 0$ for each $y \geq 0$ with $y^\top A \geq 0$ consider $A' = [I, A]$. Observe that $Ax \leq b$ has a solution $x \geq 0$ if and only if $A'x' = b$ has a solution $x' \geq 0$. Now apply what we have just proved. □

Proof of Theorem 3.9. For (1) both optima exist. Thus, if $Ax \leq b$ and $y \geq 0$, $y^\top A = c^\top$, then $c^\top x = y^\top Ax \leq y^\top b$. Now it suffices to show that there are x, y such that $Ax \leq b$, $y \geq 0$, $y^\top A = c^\top$, $c^\top x \geq y^\top b$, i.e., that

$$\text{there are } x, y \text{ such that } y \geq 0 \text{ and } \begin{pmatrix} A & 0 \\ -c^\top & b^\top \\ 0 & A^\top \\ 0 & -A^\top \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} b \\ 0 \\ c^\top \\ -c^\top \end{pmatrix}$$

By Lemma 3.11 this is equivalent to: If $u, \lambda, v, w \geq 0$ with $uA - \lambda c^\top = 0$ and $\lambda b^\top + vA^\top - wA^\top \geq 0$ then $ub + vc - wc \geq 0$.

Let u, λ, v, w satisfy this premise. If $\lambda > 0$ then $ub = \lambda^{-1}\lambda b^\top u^\top \geq \lambda^{-1}(w - v)A^\top u^\top = \lambda^{-1}\lambda(w - v)c = (w - v)c$. If $\lambda = 0$, let $Ax_0 \leq b$ and $y_0 \geq 0$, $y_0^\top A = c^\top$. (x_0, y_0 exist since P and D are not empty.) Then $ub \geq uAx_0 = 0 \geq (w - v)A^\top y_0^\top = (w - v)c$.

The claim (2) directly follows from Lemma 3.8. For (3), if D is infeasible there is nothing to show. Thus let D be feasible. From Lemma 3.11 we get: Since $Ax \leq b$ is infeasible, there is a vector $y \geq 0$ with $y^\top A = 0$ and $y^\top b < 0$. Let z be such that $z^\top A = c^\top$ and $\alpha > 0$. Then $\alpha y + z$ is feasible with objective value $\alpha y^\top b + z^\top b$, which can be made arbitrarily small since $y^\top b < 0$ and $\alpha > 0$. □

The theorem has a lot of implications but we only list two of them. The first one is called *complementary slackness* (and gives another way of proving optimality).

Corollary 3.12. *Let* $\max\{c^\top x : Ax \le b\}$ *and* $\min\{y^\top b : y^\top A = c, y \ge 0\}$ *be a primal-dual pair and let* x *and* y *be respective feasible solutions. Then the following statements are equivalent:*

(1) x *and* y *are both optimum solutions.*

(2) $c^\top x = y^\top b$.

(3) $y^\top (b - Ax) = 0$.

Secondly, the fact that a system $Ax \le b$ is infeasible can be proved by giving a vector $y \ge 0$ with $y^\top A = 0$ and $y^\top b < 0$ (Farkas' Lemma).