NOTES ON

Combinatorial Optimization

Hao Su

August 8, 2026

Contents

1 Linear Programming 2
- 1.1 The Simplex Method 2
- 1.2 Bases and Polyhedra 8
- 1.3 Convex Geometry and Alternatives 10
- 1.4 Duality 12

2 Submodular Optimization 17
- 2.1 Submodular Set Functions 17
- 2.2 Greedy Analysis under Different Constraints 18
- 2.3 Fractional Extensions 23

3 Configuration Linear Programs 24
- 3.1 Nash Social Welfare 24
- 3.2 The Configuration LP 25
- 3.3 Dual Prices and Separation 26
- 3.4 Recovering a Small Primal LP 28

Chapter 1

Linear Programming

Let $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, $b = (b_i) \in \mathbb{R}^m$, and $c = (c_j) \in \mathbb{R}^n$.

Definition 1.1 (LP). A *linear program* in *standard form* has the equivalent representations:

$$\begin{aligned} &\text{(A) maximize } \sum_{j=1}^{n} c_j x_j && && \text{and (B) maximize } c^{\mathsf{T}} x \\ &\text{subject to } \sum_{j=1}^{n} a_{ij} x_j \le b_i && (i = 1, \dots, m) && \text{subject to } Ax \le b \\ &\qquad x_j \ge 0 && (j = 1, \dots, n) && \qquad x \ge 0. \end{aligned}$$

A proposal of specific values for the decision variables is called a *solution*. A solution x is called *feasible* if it satisfies all of the constraints. It is called *optimal* if in addition it attains the desired maximum. Problems exist that are *infeasible*. A problem is *unbounded* if it has feasible solutions with arbitrarily large objective values.

1.1 The Simplex Method

The feasible region of a linear program is a *polyhedron*; its corners are called *vertices* (definition 1.5). A linear objective—unless it is unbounded—attains its maximum at a vertex. Rather than search the whole polyhedron, simplex walks along its edges from one vertex to an adjacent vertex, always choosing a direction in which the objective increases.

For example, consider

$$\begin{aligned} \text{maximize} \quad & 3x_1 + 2x_2 && && \\ \text{subject to} \quad & -x_1 + 3x_2 && \le && 12, \\ & x_1 + x_2 && \le && 8, \\ & 2x_1 - x_2 && \le && 10, \\ & x_1 \,, \quad x_2 && \ge && 0. \end{aligned} \tag{1.1}$$

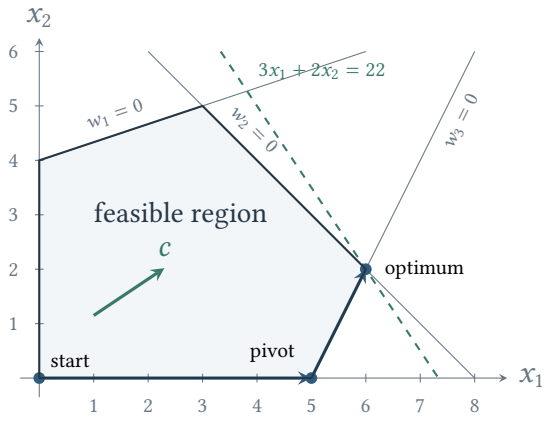

Figure 1.1: The simplex method follows edges between adjacent vertices.

1.1.1 Phase II

The dictionary formulation below is the classical primal simplex method introduced by Dantzig [1]. Introduce a slack variable $w_i \geq 0$ for each constraint and denote the objective value by ζ:

$$\zeta = \sum_{j=1}^{n} c_j x_j, \qquad w_i = b_i - \sum_{j=1}^{n} a_{ij} x_j \quad (i = 1, \dots, m). \tag{1.2}$$

Thus the inequalities have become equations together with the requirement that all x_j and w_i be nonnegative. It is convenient to regard the slacks as additional variables, $x_{n+i} = w_i$.

At any stage, row operations put these equations into a *dictionary*

$$\zeta = \bar{\zeta} + \sum_{j \in N} \bar{c}_j x_j, \qquad x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j \quad (i \in B), \tag{1.3}$$

where B and N partition the variable indices; B is called the current *basis*. The variables on the left are *basic*; those on the right are *nonbasic*. Setting every nonbasic variable to zero gives the dictionary's *basic solution*, with $x_i = \bar{b}_i$ for $i \in B$. It is a *basic feasible solution* precisely when $\bar{b}_i \geq 0$ for every $i \in B$.

Setting the nonbasic variables to zero selects a vertex. Geometrically, each equation $x_j = 0$ for $j \in N$ records a tight boundary there: $x_j = 0$ is a coordinate boundary when $j \leq n$, and $w_i = 0$ means that the corresponding original inequality is tight.

Pivot. A *pivot* exchanges one index in N with one in B, which is exactly the exchange of one tight boundary for another. Suppose $\bar{c}_k > 0$ for some $k \in N$. Increasing x_k from zero improves ζ, so x_k is an *entering variable*. While all other nonbasic variables remain zero,

$$x_i = \bar{b}_i - \bar{a}_{ik} x_k \geq 0 \qquad (i \in B).$$

Only rows with $\bar{a}_{ik} > 0$ restrict the step. The largest feasible step and a corresponding *leaving variable* x_ℓ are therefore

$$\theta = \min_{i \in B\colon \bar{a}_{ik} > 0} \frac{\bar{b}_i}{\bar{a}_{ik}}, \qquad \ell \in \arg\min_{i \in B\colon \bar{a}_{ik} > 0} \frac{\bar{b}_i}{\bar{a}_{ik}}. \tag{1.4}$$

At $x_k = \theta$, the variable x_ℓ reaches zero. Solving its row for x_k and substituting in all other rows performs the pivot: k enters B and ℓ enters N.

The stopping tests are certificates. If every $\bar{c}_j \le 0$, then for any feasible solution

$$\zeta = \bar{\zeta} + \sum_{j \in N} \bar{c}_j x_j \le \bar{\zeta},$$

so the current basic feasible solution is optimal. At the other extreme, suppose $\bar{c}_k > 0$ but $\bar{a}_{ik} \le 0$ for every $i \in B$. Then the ray

$$x(t) = x(0) + t d, \qquad d_k = 1, \quad d_j = 0 \ (j \in N \setminus \{k\}), \quad d_i = -\bar{a}_{ik} \ (i \in B) \tag{1.5}$$

is feasible for every $t \ge 0$, while its objective value increases by $t\bar{c}_k$. Thus any improving column with no positive entry in a basic row is a certificate of unboundedness.

1.1.2 Phase I

The initial dictionary (1.2) is feasible when $b \ge 0$: take $x = 0$ and $w = b$. Otherwise, Phase I introduces one artificial variable and solves

$$\begin{aligned} \text{maximize} \quad & -x_0 \\ \text{subject to} \quad & Ax - x_0 \mathbf{1} \le b, \\ & x_0, x \ge 0. \end{aligned} \tag{1.6}$$

Its slack dictionary is $w_i = b_i - a_i^{\mathsf{T}} x + x_0$. If some $b_i < 0$, pivoting x_0 into the row with the smallest b_i immediately produces a feasible dictionary for the auxiliary problem. Apply algorithm 1 to it. If the optimum is negative, the original linear program is infeasible; otherwise set $x_0 = 0$ and remove it from the dictionary. If it is basic, pivot it out when possible; a row in which this is impossible ($x_0 = 0$) is redundant and may be deleted. Restore the original objective $\zeta = c^{\mathsf{T}} x$ and run Phase II.

1.1.3 Degeneracy

A feasible dictionary is *degenerate* when $\bar{b}_i = 0$ for some basic variable x_i. Geometrically, a vertex is ordinarily determined by just enough independent tight boundaries. A zero basic variable makes an additional nonnegativity boundary tight, so different choices of tight boundaries—and hence different dictionaries—may describe the same vertex.

If such a row wins the ratio test, then $\theta = 0$: the pivot changes the basis but neither the point nor ζ. An unfortunate pivot rule can keep exchanging these descriptions and eventually repeat a basis. This is *cycling*. Conversely, nontermination must be cycling because there are only finitely many bases; every pivot in a cycle is degenerate because ζ never decreases.

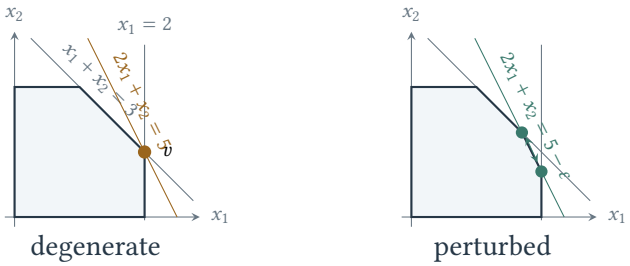

Figure 1.2: Three boundaries meet at a degenerate vertex (left). An infinitesimal perturbation separates it into nearby nondegenerate vertices (right).

Two standard devices prevent cycling.

Symbolic perturbation and the lexicographic rule. Replace the initial right-hand sides by

$$b_i + \varepsilon_i, \qquad 0 < \varepsilon_m \ll \cdots \ll \varepsilon_2 \ll \varepsilon_1 \ll \text{the original data},$$

and carry the ε_i symbolically. Equivalently, attach the vector e_i to row i and perform the same row operations on these vectors as on the right-hand sides. After these operations, write the perturbed basic row as

$$x_i = \bar{b}_i + \sum_{h=1}^{m} r_{ih}\varepsilon_h - \bar{a}_{ik}x_k - \cdots .$$

If x_k enters, row i would become tight after the symbolic step

$$\frac{\bar{b}_i + r_{i1}\varepsilon_1 + \cdots + r_{im}\varepsilon_m}{\bar{a}_{ik}} \qquad (\bar{a}_{ik} > 0).$$

The leaving row is the one reached first. Since ε_1 dominates ε_2, and so on, this is found by comparing

$$\frac{(\bar{b}_i, r_{i1}, \ldots, r_{im})}{\bar{a}_{ik}} \qquad (\bar{a}_{ik} > 0)$$

lexicographically: first compare the usual ratios $\bar{b}_i/\bar{a}_{ik}$; if they tie, compare $r_{i1}/\bar{a}_{ik}$, then $r_{i2}/\bar{a}_{ik}$, and so forth. Thus the method imagines separating coincident boundaries by successively smaller displacements and chooses the unique boundary encountered first. It changes only ties in the leaving-variable test; the entering variable may still be chosen by any improving rule, such as the largest reduced cost.

Theorem 1.2 (Lexicographic anti-cycling). *The lexicographic leaving rule makes every pivot nondegenerate and hence makes the simplex method terminate.*

Proof. Initially the coefficient vectors of $(\varepsilon_1, \dots, \varepsilon_m)$ form the identity matrix. After any sequence of pivots they form a matrix obtained by reversible row operations, so it still has rank m. In particular, no row has zero coefficients for every ε_i; because the scales are independent, no basic right-hand side is symbolically zero. Thus every step has $\theta > 0$ and strictly increases ζ. A basis can therefore never repeat, and only finitely many bases exist. At termination, taking every positive perturbation back to zero gives an optimal dictionary for the original problem. ⊣

Bland's rule. Give all original and slack variables distinct indices. Among improving nonbasic variables choose the one of smallest index; among rows tied in the minimum-ratio test, choose the basic variable of smallest index.

Algorithm 1 Simplex method with Bland's rule (Phase II)

Require: A feasible dictionary of the form (1.3)

while there exists $k \in N$ with $\bar{c}_k > 0$ **do**
  $k \leftarrow \min\{j \in N : \bar{c}_j > 0\}$
  if $\bar{a}_{ik} \le 0$ for every $i \in B$ **then**
    return UNBOUNDED and the ray (1.5)
  end if
  $\theta \leftarrow \min_{i \in B:\, \bar{a}_{ik} > 0} \bar{b}_i / \bar{a}_{ik}$
  $\ell \leftarrow \min\{i \in B : \bar{a}_{ik} > 0,\ \bar{b}_i / \bar{a}_{ik} = \theta\}$
  Pivot on row ℓ and column k; exchange k and ℓ
end while
return $x_N = 0$, $x_i = \bar{b}_i$ for $i \in B$, and value $\bar{\zeta}$

Theorem 1.3 (Bland's anti-cycling rule). *The simplex method with Bland's rule never cycles and therefore terminates.*

Proof. Suppose instead that some dictionaries form a cycle. Every pivot in it is degenerate, so all dictionaries represent the same point and objective value. Call a variable *fickle* if it is basic in some dictionaries of the cycle and nonbasic in others. Let x_t be the fickle variable of largest index, and choose a dictionary D in which x_s enters and x_t leaves. Then $s < t$. Choose a later dictionary D^* in which x_t enters, and extend its reduced costs $\bar{c}^*_j$ by setting them to zero on its basic variables.

Formally increase x_s by y from D, without imposing nonnegativity. Writing the same objective

using D and D^* and comparing the coefficient of y gives

$$\bar{c}_s - \bar{c}^*_s + \sum_{i \in B} \bar{c}^*_i \bar{a}_{is} = 0. \tag{1.7}$$

Here $\bar{c}_s > 0$, while $\bar{c}^*_s \le 0$ because $s < t$ and Bland chose t to enter in D^*. Hence some $r \in B$ satisfies $\bar{c}^*_r \bar{a}_{rs} < 0$. Such an x_r is fickle. Moreover $r < t$: the term with $r = t$ is positive, since x_t enters in D^* and leaves in D. Bland's entering rule in D^* now gives $\bar{c}^*_r \le 0$, so $\bar{a}_{rs} > 0$. Every fickle variable is zero at the common point, hence $\bar{b}_r = 0$ in D. Thus x_r was a zero-ratio leaving candidate with $r < t$, contradicting Bland's choice of x_t. ⊣

Theorem 1.4 (Fundamental theorem of linear programming). *Every feasible linear program in standard form has a basic feasible solution, and whenever an optimal solution exists, a basic optimal solution exists. If no optimal solution exists, the program is either infeasible or unbounded.*

Proof. Run Phase I and Phase II using Bland's rule. By the preceding theorem both terminate. Phase I either certifies infeasibility or returns a basic feasible solution. From that basis, Phase II either returns the unbounded ray (1.5) or a basic optimal solution. ⊣

1.1.4 Running time

There are at most

$$\binom{m+n}{m} \le 2^{m+n}$$

bases. An anti-cycling rule visits no basis twice, so this is an exponential upper bound on the number of pivots. A dense tableau pivot takes $O(mn)$ arithmetic operations, giving the crude arithmetic bound

$$O\left(mn\binom{m+n}{m}\right).$$

This count deliberately ignores coefficient bit lengths.

The exponential behavior is real, not just an artifact of the bound. On the Klee–Minty family

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} 2^{n-j} x_j \\ \text{subject to} \quad & 2\sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \le 100^{i-1} \quad (i = 1, \dots, n), \\ & x \ge 0, \end{aligned} \tag{1.8}$$

the largest-reduced-cost rule visits all 2^n vertices of a distorted cube and requires $2^n - 1$ pivots. This proves an exponential lower bound for that pivot rule.

Informally, if the coefficients of an arbitrary linear program are perturbed independently by Gaussian noise of scale δ, then a suitable two-phase shadow-vertex simplex method terminates in an expected $\mathrm{poly}(1/\delta, \text{input size})$ number of iterations [3]. Thus small random noise can yield polynomial expected behavior even though exponential worst cases remain.

1.2 Bases and Polyhedra

The dictionary notation is well suited to hand calculations. Matrix notation exposes the linear algebra behind it and makes the connection between bases, vertices, and dual solutions transparent.

1.2.1 The matrix dictionary

After adding slacks, write the program as

$$\begin{aligned} &\text{maximize} && \widehat{c}^{\mathsf{T}} u \\ &\text{subject to} && \widehat{A} u = b, \qquad \widehat{A} = \begin{bmatrix} A & I_m \end{bmatrix}, \quad u = \begin{bmatrix} x \\ w \end{bmatrix}, \quad \widehat{c} = \begin{bmatrix} c \\ 0 \end{bmatrix}. \\ & && u \geq 0, \end{aligned} \tag{1.9}$$

Let $\mathcal{B}$ be a set of m column indices for which $B = \widehat{A}_{\mathcal{B}}$ is nonsingular, and let $\mathcal{N}$ contain the remaining indices, with $N = \widehat{A}_{\mathcal{N}}$. Partitioning u and $\widehat{c}$ accordingly gives

$$B u_{\mathcal{B}} + N u_{\mathcal{N}} = b.$$

Solving for the basic variables and substituting into the objective produces the matrix form of a dictionary:

$$\begin{aligned} u_{\mathcal{B}} &= B^{-1} b - B^{-1} N u_{\mathcal{N}}, \\ \zeta &= \widehat{c}_{\mathcal{B}}^{\mathsf{T}} B^{-1} b + \left(\widehat{c}_{\mathcal{N}}^{\mathsf{T}} - \widehat{c}_{\mathcal{B}}^{\mathsf{T}} B^{-1} N \right) u_{\mathcal{N}}. \end{aligned} \tag{1.10}$$

Thus

$$\bar{b} = B^{-1} b, \qquad \bar{c}_{\mathcal{N}} = \widehat{c}_{\mathcal{N}} - N^{\mathsf{T}} B^{-\mathsf{T}} \widehat{c}_{\mathcal{B}}. \tag{1.11}$$

The associated basic solution is feasible exactly when $B^{-1} b \geq 0$, and a feasible basis is optimal exactly when $\bar{c}_{\mathcal{N}} \leq 0$. Although B^{-1} makes the formulas concise, an implementation solves linear systems with B and B^{T}; it does not form the inverse explicitly.

1.2.2 Faces and valid inequalities

Definition 1.5 (Polyhedral language). A *polyhedron* is the intersection of finitely many closed half-spaces; a bounded polyhedron is a *polytope*. An inequality $a^{\mathsf{T}}x \le \beta$ is *valid* for a polyhedron P if every point of P satisfies it. When the hyperplane $a^{\mathsf{T}}x = \beta$ meets P, it is a *supporting hyperplane*, and

$$F = \{x \in P \,:\, a^{\mathsf{T}}x = \beta\}$$

is a *face* of P. A face of dimension $\dim P - 1$ is a *facet*; a zero-dimensional face is a *vertex*.

Equivalently, a point $v \in P$ is a vertex, or *extreme point*, when it cannot be written as $v = \lambda x + (1-\lambda)y$ for distinct $x, y \in P$ and $0 < \lambda < 1$. Faces are precisely the sets on which some linear objective attains its maximum over P. Facets are the maximal proper faces.

Theorem 1.6 (Basic feasible solutions are vertices). *Suppose $\widehat{A}$ has row rank m. A feasible solution of $\widehat{A}u = b,\ u \ge 0$ is a basic feasible solution if and only if it is a vertex of the feasible polyhedron.*

Proof. Let u be basic for the basis $\mathcal{B}$, so $u_{\mathcal{N}} = 0$. If $u = \lambda p + (1-\lambda)q$ for feasible p, q and $0 < \lambda < 1$, nonnegativity forces $p_{\mathcal{N}} = q_{\mathcal{N}} = 0$. Hence $Bp_{\mathcal{B}} = b = Bq_{\mathcal{B}}$, and nonsingularity of B gives $p = q = u$. Thus u is a vertex.

Conversely, let u be a vertex and put $S = \{j \,:\, u_j > 0\}$. If the columns $\widehat{A}_S$ were dependent, there would be a nonzero vector d, supported on S, with $\widehat{A}d = 0$. For sufficiently small $\varepsilon > 0$, both $u + \varepsilon d$ and $u - \varepsilon d$ would be feasible, expressing u as their midpoint. Therefore $\widehat{A}_S$ is independent. Extend its columns to a basis of $\mathbb{R}^m$; the added basic variables have value zero, so u is a basic feasible solution. ⊣

The slack map $x \mapsto (x, b - Ax)$ is an affine bijection from the original polyhedron onto its equality-form image, so it preserves vertices and edges. Although this image lies in $\mathbb{R}^{n+m}$, the slack coordinates add no independent directions: moving x by d forces w to move by $-Ad$. Thus simplex treats x and w symmetrically as variables, but geometrically it follows the same edge graph as the original polyhedron; its basic feasible points are precisely the embedded vertices.

The exact equality count now makes degeneracy precise. For $\widehat{A}u = b,\ u \ge 0$ in $\mathbb{R}^q$, with $\operatorname{rank}\widehat{A} = m$, a basis determines the basic solution through the m row equalities together with the $q - m$ independent equalities $u_j = 0$ for the nonbasic variables. If a basic variable is also zero, an additional nonnegativity constraint is active. The active equalities are then dependent, and different bases may represent the same vertex; a pivot between them changes the basis without moving the point.

If an LP has finite attained optimum ζ^*, its optimal solutions form the face

$$\{x \in P \,:\, c^{\mathsf{T}}x = \zeta^*\}.$$

The theorem 1.4 and theorem 1.6 say that, in standard form, this face contains an optimal vertex.

1.3 Convex Geometry and Alternatives

1.3.1 Convex sets and cones

Definition 1.7 (Convexity and cone). A set $C \subseteq \mathbb{R}^d$ is *convex* if $\lambda x + (1-\lambda)y \in C$ whenever $x, y \in C$ and $0 \le \lambda \le 1$. For $S \subseteq \mathbb{R}^d$, its *convex hull* is

$$\operatorname{conv}(S) = \left\{ \sum_{i=1}^{r} \lambda_i x_i \,:\, r \ge 1,\ x_i \in S,\ \lambda \ge 0,\ \sum_{i=1}^{r} \lambda_i = 1 \right\}.$$

A set C is a *cone* if $\alpha x \in C$ whenever $x \in C$ and $\alpha \ge 0$. A *convex cone* is a cone that is convex, equivalently a set closed under nonnegative linear combinations. The *conic hull* $\operatorname{cone}(S)$ consists of all finite nonnegative linear combinations of elements of S.

Intersections of convex sets are convex; in particular, every polyhedron is convex. If S is finite, $\operatorname{cone}(S)$ is a closed convex cone, called a *polyhedral cone*.

Definition 1.8 (Dual space and dual cone). The *dual space* V^* consists of the linear functionals $\varphi : V \to \mathbb{R}$, paired with vectors by $\langle \varphi, x \rangle = \varphi(x)$. In Euclidean space the inner product represents every functional by a vector y, so that $\varphi(x) = y^{\mathsf{T}}x$. For a cone $C \subseteq \mathbb{R}^d$, its *dual cone* is

$$C^* = \{y \,:\, y^{\mathsf{T}}x \ge 0 \text{ for every } x \in C\}.$$

The *polar cone* uses the opposite sign: $C^\circ = -C^* = \{y \,:\, y^{\mathsf{T}}x \le 0 \text{ for every } x \in C\}$.

Thus C^* contains the linear measurements that regard every direction of C as nonnegative. The angle interpretation is only Euclidean notation for this positivity. For example,

$$(\mathbb{R}^d_+)^* = \mathbb{R}^d_+, \qquad (\operatorname{cone}\{a\})^* = \{y \,:\, y^{\mathsf{T}}a \ge 0\}, \qquad L^* = L^\perp$$

for every linear subspace L, since L contains both x and $-x$.

1.3.2 Separation

Theorem 1.9 (Point–set separation). *If $C \subseteq \mathbb{R}^d$ is nonempty, closed, and convex and $p \notin C$, then some $y \neq 0$ satisfies*

$$y^{\mathsf{T}} x \geq y^{\mathsf{T}} z > y^{\mathsf{T}} p \qquad (x \in C)$$

for a closest point $z \in C$ to p.

Proof. A closest point exists because the distance minimization may be restricted to a sufficiently large compact ball. For $x \in C$, put

$$\phi(t) = \|z + t(x - z) - p\|^2, \qquad 0 \leq t \leq 1.$$

The segment $z + t(x - z)$ remains in C, so the squared distance $\phi(t)$ is minimized at $t = 0$. Its right derivative there is therefore nonnegative:

$$0 \leq \phi'_+(0) = 2(z - p)^{\mathsf{T}}(x - z),$$

or equivalently $(p - z)^{\mathsf{T}}(x - z) \leq 0$. Taking $y = z - p$ gives the first inequality, while $y^{\mathsf{T}}(z - p) = \|z - p\|^2 > 0$ gives the strict one. ⊣

If C is also a cone, then $0, 2z \in C$. Substituting these two points for x in the separating inequality gives $0 \geq y^{\mathsf{T}} z$ and $2y^{\mathsf{T}} z \geq y^{\mathsf{T}} z$. Thus $y^{\mathsf{T}} z = 0$, and consequently $y^{\mathsf{T}} x \geq 0$ for every $x \in C$. The theorem therefore becomes the fundamental conic alternative

$$p \notin C \quad \Longrightarrow \quad \exists\, y \in C^* \,:\; y^{\mathsf{T}} p < 0. \tag{1.12}$$

Exactly one of $p \in C$ and the displayed certificate can hold. Applying (1.12) to a point outside a closed convex cone also gives the *bipolar identity* $C^{**} = C$: the positive linear measurements in C^* determine C completely.

1.3.3 Farkas certificates

Theorem 1.10 (Farkas' lemma). *Exactly one of the following systems is feasible:*

1. *$Ax \leq b$, where x is unrestricted in sign;*
2. *$A^{\mathsf{T}} y = 0,\ y \geq 0,\ b^{\mathsf{T}} y < 0$.*

Proof. Introduce the slack $s = b - Ax \geq 0$. The first system is feasible exactly when

$$b \in C := \operatorname{im}(A) + \mathbb{R}^m_+ = \operatorname{cone}\{\pm A_{\cdot j}, e_i \,:\, j \leq n,\ i \leq m\}.$$

This is a closed polyhedral cone, and

$$C^* = (\mathrm{im}(A))^* \cap (\mathbb{R}^m_+)^* = \mathrm{im}(A)^\perp \cap \mathbb{R}^m_+ = \{y \,:\, A^\mathsf{T} y = 0,\ y \geq 0\}.$$

If $b \notin C$, the conic alternative supplies $y \in C^*$ with $b^\mathsf{T} y < 0$. So both cannot be infeasible. Also both cannot be feasible: multiplying $(Ax)^\mathsf{T}$ by y would give $0 \leq b^\mathsf{T} y < 0$. ⊣

Thus a Farkas vector is simply a separator of b from the cone of attainable right-hand sides.

Theorem 1.11 (Polyhedral separation). *Any two disjoint nonempty polyhedra can be strongly separated.*

Proof. Write $P = \{x \,:\, Ax \leq b\}$ and $Q = \{x \,:\, Cx \leq d\}$. Their disjointness makes the combined system infeasible. By theorem 1.10, there are $y, z \geq 0$ such that

$$A^\mathsf{T} y + C^\mathsf{T} z = 0, \qquad b^\mathsf{T} y + d^\mathsf{T} z < 0.$$

Put $a = A^\mathsf{T} y = -C^\mathsf{T} z$, $\alpha = b^\mathsf{T} y$, and $\beta = -d^\mathsf{T} z$. Then

$$a^\mathsf{T} x \leq \alpha < \beta \leq a^\mathsf{T} q \qquad (x \in P,\ q \in Q).$$

Moreover $a \neq 0$: otherwise nonemptiness would force $b^\mathsf{T} y, d^\mathsf{T} z \geq 0$, contradicting their negative sum. ⊣

This reverse reduction shows that Farkas and separation are the same alternative in polyhedral language. The general point–set theorem is broader; no such equivalence is claimed for arbitrary convex sets.

1.4 Duality

1.4.1 Bounds as certificates

The most useful interpretation of duality is that it manufactures upper bounds. For any $y \geq 0$, multiplying the primal inequalities by y_i and adding gives

$$y^\mathsf{T} Ax \leq y^\mathsf{T} b.$$

If also $A^\mathsf{T} y \geq c$, then $x \geq 0$ implies $c^\mathsf{T} x \leq y^\mathsf{T} Ax$. Seeking the best such bound produces

$$\text{(P)} \quad \max_{x \geq 0} \{c^\mathsf{T} x \,:\, Ax \leq b\} \qquad\qquad \text{(D)} \quad \min_{y \geq 0} \{b^\mathsf{T} y \,:\, A^\mathsf{T} y \geq c\}. \tag{1.13}$$

The same construction gives all sign rules. Multiplication by $y_i \geq 0$ preserves a $\leq$ row; a $\geq$ row requires $y_i \leq 0$, and an equality permits either sign. Comparing the combined coefficient with c_j gives $\geq$, $=$, or $\leq$ according as $x_j \geq 0$, x_j is free, or $x_j \leq 0$.

After adding primal slacks, the same pair is

$$(\widehat{\mathrm{P}}) \quad \max_{u \geq 0} \{\widehat{c}^{\mathsf{T}} u \,:\, \widehat{A} u = b\} \qquad\qquad (\widehat{\mathrm{D}}) \quad \min_{y \in \mathbb{R}^m} \{b^{\mathsf{T}} y \,:\, \widehat{A}^{\mathsf{T}} y \geq \widehat{c}\}. \tag{1.14}$$

Although y is formally free because it corresponds to equality constraints, the slack block encodes its original sign:

$$\begin{bmatrix} A^{\mathsf{T}} \\ I_m \end{bmatrix} y \geq \begin{bmatrix} c \\ 0 \end{bmatrix} \quad \Longleftrightarrow \quad A^{\mathsf{T}} y \geq c, \qquad y \geq 0.$$

Let

$$w = b - Ax \geq 0, \qquad z = A^{\mathsf{T}} y - c \geq 0$$

denote the primal and dual slacks.

Theorem 1.12 (Weak duality). *Every primal-feasible x and dual-feasible y satisfy $c^{\mathsf{T}} x \leq b^{\mathsf{T}} y$.*

Proof. Since $c = A^{\mathsf{T}} y - z$, the objective normal is a nonnegative combination of the normals a_i and $-e_j$ of the defining half-spaces. Adding those inequalities gives the objective-aligned ceiling $c^{\mathsf{T}} x \leq b^{\mathsf{T}} y$. Algebraically, its gap is

$$b^{\mathsf{T}} y - c^{\mathsf{T}} x = y^{\mathsf{T}}(b - Ax) + x^{\mathsf{T}}(A^{\mathsf{T}} y - c) = y^{\mathsf{T}} w + x^{\mathsf{T}} z \geq 0. \tag{1.15}$$

⊣

Thus primal points give achieved values, while dual points give certified ceilings parallel to the objective level sets.

Theorem 1.13 (Strong duality). *If either problem in (1.13) has a finite optimal solution, then so does the other, and their optimal values are equal.*

Geometric proof. Suppose the primal optimum p^* is attained. Introduce the attainable-value cone

$$K = \{(Ax + s, c^{\mathsf{T}} x - r) \,:\, x, s, r \geq 0\} \subseteq \mathbb{R}^{m+1}.$$

It is the polyhedral cone generated by $(A_{\cdot j}, c_j)$, $(e_i, 0)$, and $(0, -1)$. Membership has the direct meaning

$$(b, \alpha) \in K \quad \Longleftrightarrow \quad \exists x \geq 0 \,:\, Ax \leq b,\ c^{\mathsf{T}} x \geq \alpha.$$

Hence $(b, p^*) \in K$, whereas $(b, p^* + \varepsilon) \notin K$ for every $\varepsilon > 0$.

By conic separation there is $q = (u, \eta) \in K^*$ with $q^\mathsf{T}(b, p^* + \varepsilon) < 0$. Since $q^\mathsf{T}(b, p^*) \geq 0$, necessarily $\eta < 0$. Put $\lambda = -\eta > 0$ and $y = u/\lambda$. Testing q on the generators of K shows

$$u \geq 0, \qquad A^\mathsf{T} u + \eta c \geq 0,$$

or equivalently $y \geq 0$ and $A^\mathsf{T} y \geq c$. Thus y is dual feasible, and the two evaluations give

$$p^* \leq b^\mathsf{T} y < p^* + \varepsilon.$$

The dual infimum is therefore p^*. Its feasible region is a polyhedron, so the theorem 1.4 makes this finite infimum attained. The converse follows symmetrically. ⊣

Basis proof. At an optimal basis of (1.9), seek a dual ceiling exact on the basic variables. This forces

$$B^\mathsf{T} y = \widehat{c}_\mathcal{B}, \qquad y = B^{-\mathsf{T}} \widehat{c}_\mathcal{B}.$$

Since $B u_\mathcal{B} = b$,

$$b^\mathsf{T} y = u_\mathcal{B}^\mathsf{T} B^\mathsf{T} y = \widehat{c}_\mathcal{B}^\mathsf{T} u_\mathcal{B} = \widehat{c}^\mathsf{T} u.$$

On the nonbasic columns, simplex optimality is exactly

$$\bar{c}_\mathcal{N} = \widehat{c}_\mathcal{N} - N^\mathsf{T} y \leq 0 \quad \Longleftrightarrow \quad N^\mathsf{T} y \geq \widehat{c}_\mathcal{N}.$$

Hence $\widehat{A}^\mathsf{T} y \geq \widehat{c}$, so the exact ceiling is globally valid. Weak duality makes both solutions optimal. ⊣

Within finite-dimensional polyhedral theory, separation, Farkas, and strong duality are equivalent forms of the same alternative. Separation gave Farkas and strong duality above; Farkas gave polyhedral separation. Conversely, strong duality applied to the zero-objective feasibility LP gives Farkas, and separating a point from a polyhedral cone recovers the conic alternative. Directly, Farkas applied to the proposed dual certificate

$$A^\mathsf{T} y \geq c, \qquad y \geq 0, \qquad b^\mathsf{T} y \leq p^*$$

would otherwise produce either a primal point of value greater than p^* or an improving unbounded ray.

There are four possible status patterns: equal finite optima; primal unbounded and dual infeasible; dual unbounded and primal infeasible; or both infeasible.

1.4.2 Conic and LP complementarity

If $s \in C$ and $y \in C^*$, then $y^\mathsf{T} s \geq 0$. They are *complementary* when $y^\mathsf{T} s = 0$; geometrically, s lies in the exposed face $C \cap y^\perp$. For the self-dual orthant,

$$s, y \geq 0, \qquad y^\mathsf{T} s = 0 \quad \Longleftrightarrow \quad y_i s_i = 0 \quad \text{for every } i.$$

Theorem 1.14 (Complementary slackness). *Let x and y be primal and dual feasible. They are optimal if and only if*

$$x_j z_j = 0 \quad (j = 1, \dots, n), \qquad y_i w_i = 0 \quad (i = 1, \dots, m).$$

Proof. By (1.15), the feasible pair is optimal exactly when the two orthant pairings $y^{\mathsf{T}} w$ and $x^{\mathsf{T}} z$ vanish. Their nonnegative summands then vanish coordinatewise. ⊣

Thus a positive primal variable forces its dual constraint to be tight, and a positive dual variable forces its primal constraint to be tight. In combinatorial optimization this often turns optimality into a structural matching between primal objects and dual inequalities.

1.4.3 The dual simplex method

Primal simplex maintains $\bar{b} \geq 0$ and pivots until $\bar{c} \leq 0$. Dual simplex is its mirror image: it maintains $\bar{c} \leq 0$ and repairs negative basic values until $\bar{b} \geq 0$.

Suppose $x_\ell = \bar{b}_\ell - \sum_{j \in N} \bar{a}_{\ell j} x_j$ has $\bar{b}_\ell < 0$. To raise x_ℓ to zero, an entering variable must have $\bar{a}_{\ell j} < 0$. Among these columns, the dual ratio test chooses a minimum of

$$\frac{\bar{c}_j}{\bar{a}_{\ell j}}.$$

Algorithm 2 Dual simplex method with Bland tie-breaking

Require: A dual-feasible dictionary (1.3), so $\bar{c}_j \leq 0$ for every $j \in N$

while there exists $i \in B$ with $\bar{b}_i < 0$ **do**
  $\ell \leftarrow \min\{i \in B : \bar{b}_i < 0\}$
  if $\bar{a}_{\ell j} \geq 0$ for every $j \in N$ **then**
    return INFEASIBLE
  end if
  $\rho \leftarrow \min_{j \in N : \bar{a}_{\ell j} < 0} \bar{c}_j / \bar{a}_{\ell j}$
  $k \leftarrow \min\{j \in N : \bar{a}_{\ell j} < 0,\ \bar{c}_j / \bar{a}_{\ell j} = \rho\}$
  Pivot on row ℓ and column k; exchange k and ℓ
end while
return the current basic optimal solution

The dual ratio test preserves $\bar{c} \leq 0$ at every pivot. If a negative basic row has no negative coefficient, it remains negative for every nonnegative choice of the nonbasic variables and certifies infeasibility. Bland's rule or lexicographic perturbation prevents cycling exactly as for primal simplex.

Dual simplex is useful when a change preserves reduced-cost signs but destroys primal feasibility. It also gives an alternative Phase I: choose a temporary objective for which the initial dictionary is dual feasible, run dual simplex to obtain a feasible basis, restore the intended objective, and run primal simplex.

1.4.4 Lagrangian duality

Dual variables can also be understood as penalties for violating constraints. For the primal problem in (1.13), define

$$L(x, y) = c^\top x + y^\top (b - Ax), \qquad x, y \geq 0.$$

For feasible x, $L(x, y) \geq c^\top x$, so

$$g(y) = \sup_{x \geq 0} L(x, y)$$

is an upper bound for every $y \geq 0$. Since

$$L(x, y) = b^\top y + (c - A^\top y)^\top x,$$

$$g(y) = \begin{cases} b^\top y, & A^\top y \geq c, \\ +\infty, & \text{otherwise.} \end{cases}$$

Minimizing g is exactly the LP dual: weak duality says every multiplier gives a bound, and strong duality says the best LP bound is exact.

For a discrete feasible set, the same construction prices violations of difficult constraints and produces a *Lagrangian relaxation*. Its optimum remains a useful bound and often decomposes into simpler combinatorial subproblems, but it may have a nonzero duality gap.

Chapter 2

Submodular Optimization

A set function is submodular when additional elements become less valuable as the set grows. This diminishing-returns principle is weak enough to model many objectives, but strong enough to support approximation algorithms using only value queries.

2.1 Submodular Set Functions

Let U be a finite ground set, let $f : 2^U \to \mathbb{R}$, and let $S \subseteq U$. For $e \in U \setminus S$, the *marginal value* of e at S is

$$\Delta(e \mid S) = f(S \cup \{e\}) - f(S).$$

Definition 2.1 (Submodularity). The function f is *submodular* if it has diminishing returns:

$$\Delta(e \mid A) \geq \Delta(e \mid B) \qquad (A \subseteq B,\ e \notin B).$$

Equivalently, f is submodular if

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \qquad (A, B \subseteq U).$$

Call f *normalized* if $f(\emptyset) = 0$, *nonnegative* if $f(S) \geq 0$ for every $S \subseteq U$, *monotone* if $A \subseteq B$ implies $f(A) \leq f(B)$, and *symmetric* if $f(S) = f(U \setminus S)$ for every $S \subseteq U$.

Example 2.2. Submodular functions.

1. Additive functions $f(S) = \sum_{e \in S} w_e$ have constant marginals. Budget-additive functions $\min\{B, \sum_{e \in S} w_e\}$ exhibit diminishing returns.
2. If each $e \in U$ represents a set C_e, then $f(S) = |\bigcup_{e \in S} C_e|$ is a monotone coverage function.
3. The number of edges $f(S)$ crossing $(S, U \setminus S)$ is symmetric, but not monotone.
4. Given jointly distributed discrete random variables $(X_e)_{e \in U}$, their *entropy function* is $f(S) = H(X_e : e \in S)$. Its marginal $\Delta(e \mid S) = H(X_e \mid X_s : s \in S)$ is the conditional entropy and decreases as more variables are conditioned upon.

Lemma 2.3 (Consequences of submodularity). *Let $f : 2^U \to \mathbb{R}$ be submodular.*

1. *For all $S, T \subseteq U$,*
$$f(S \cup T) - f(S) \le \sum_{e \in T \setminus S} \Delta(e \mid S).$$
2. *For every $C \subseteq U$, the residual function*
$$f_C(S) = f(C \cup S) - f(C)$$
is submodular.
3. *If f is nonnegative, then it is* subadditive*; that is,*
$$f(A \cup B) \le f(A) + f(B) \qquad (A, B \subseteq U).$$

2.2 Greedy Analysis under Different Constraints

2.2.1 A Cardinality Constraint

Consider
$$\max\{f(S) : S \subseteq U,\ |S| \le k\},$$
where f is normalized, monotone, and submodular. The natural algorithm repeatedly takes the largest current marginal.

Algorithm 3 Greedy submodular maximization under a cardinality constraint

Require: A value oracle for $f : 2^U \to \mathbb{R}_{\ge 0}$ and $k \le |U|$

$S \leftarrow \emptyset$
for $t = 1, \dots, k$ **do**
  Choose $e \in U \setminus S$ maximizing $\Delta(e \mid S)$
  $S \leftarrow S \cup \{e\}$
end for
return S

Theorem 2.4 (Cardinality greedy). *If S_k is returned by algorithm 3 and O is optimal, then*
$$f(S_k) \ge \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(O) \ge \left(1 - \frac{1}{e}\right) f(O).$$

Proof. At the start of iteration t, lemma 2.3 and monotonicity give

$$f(O) - f(S_{t-1}) \le f(S_{t-1} \cup O) - f(S_{t-1}) \le \sum_{e \in O \setminus S_{t-1}} \Delta(e \mid S_{t-1}).$$

There are at most k summands, so greedy gains at least $(f(O) - f(S_{t-1}))/k$. Hence the remaining gap satisfies

$$f(O) - f(S_t) \le \left(1 - \frac{1}{k}\right)\big(f(O) - f(S_{t-1})\big).$$

Iterate this recurrence from $f(S_0) = 0$ and use $(1 - 1/k)^k \le e^{-1}$. ⊣

2.2.2 A Knapsack Constraint

Give each element a positive cost c_e and let $c(S) = \sum_{e \in S} c_e$. The knapsack problem is

$$\max\{f(S) \,:\, S \subseteq U,\ c(S) \le B\}.$$

where f is normalized, monotone, and submodular. Elements with $c_e > B$ may be discarded. Cardinality greedy becomes *density greedy*: choose the largest gain per unit cost, $\Delta(e \mid S)/c_e$. If O is optimal, then at a current set S, lemma 2.3 implies that some $e \in O \setminus S$ has density at least

$$\frac{f(O) - f(S)}{B}.$$

Let e_t be the t-th greedy element, $S_t = S_{t-1} \cup \{e_t\}$, and $R_t = f(O) - f(S_t)$. As long as greedy may compare itself with every element of O,

$$f(S_t) - f(S_{t-1}) \ge \frac{c_{e_t}}{B} R_{t-1}, \qquad R_t \le \left(1 - \frac{c_{e_t}}{B}\right) R_{t-1}.$$

Consequently,

$$R_t \le f(O) \prod_{j=1}^{t} \left(1 - \frac{c_{e_j}}{B}\right) \le e^{-c(S_t)/B} f(O).$$

The corresponding continuous picture is also useful intuition: writing x for cost spent and y for value accumulated suggests

$$\frac{dy}{dx} \ge \frac{f(O) - y}{B}, \qquad y(x) \ge (1 - e^{-x/B}) f(O).$$

Both calculations give the familiar $1 - 1/e$ bound once the accumulated cost reaches B, but the item that crosses B makes the set infeasible.

The following modification guesses the first three important elements, filters the remaining elements by the third guessed marginal, and removes the final overflowing item. It is due to Sviridenko [4].

Algorithm 4 Density greedy with partial enumeration

Require: A value oracle for $f : 2^U \to \mathbb{R}_{\geq 0}$, costs $c > 0$, and budget B

Choose $S^* \in \arg\max\{f(S) : |S| \leq 2,\ c(S) \leq B\}$
for all ordered triples (i_1, i_2, i_3) of distinct elements with $c(I) \leq B$, where $I = \{i_1, i_2, i_3\}$ **do**
    $\tau \leftarrow \Delta(i_3 \mid \{i_1, i_2\})$
    $V \leftarrow \{e \in U \setminus I : \Delta(e \mid \{i_1, i_2\}) \leq \tau\}$
    $A \leftarrow \emptyset$
    while $c(A) < B - c(I)$ and $V \setminus A \neq \emptyset$ **do**
        Choose $e \in V \setminus A$ maximizing $\Delta(e \mid I \cup A)/c_e$
        $A \leftarrow A \cup \{e\}$
    end while
    Remove from A the last element added
    if $f(I \cup A) > f(S^*)$ **then**
        $S^* \leftarrow I \cup A$
    end if
end for
return S^*

Theorem 2.5 (Knapsack greedy with partial enumeration). *For a normalized monotone submodular function, algorithm 4 returns a feasible set S satisfying*

$$f(S) \geq \left(1 - \frac{1}{e}\right) f(O).$$

Proof. If $|O| \leq 2$, the initialization of algorithm 4 considers O itself. Suppose $|O| \geq 3$, and order three elements of O by

$$o_1 \in \arg\max_{e \in O} f(\{e\}),$$
$$o_2 \in \arg\max_{e \in O \setminus \{o_1\}} \Delta(e \mid \{o_1\}),$$
$$o_3 \in \arg\max_{e \in O \setminus \{o_1, o_2\}} \Delta(e \mid \{o_1, o_2\}).$$

Consider the iteration that guesses the ordered triple (o_1, o_2, o_3). Put $I = \{o_1, o_2, o_3\}$, $B' = B - c(I)$, and

$$\tau = \Delta(o_3 \mid \{o_1, o_2\}).$$

If $O = I$, this iteration already considers the optimal set I, so assume $O \setminus I \neq \emptyset$; in particular, $B' > 0$. Every element of $O \setminus I$ survives the filtering step by the choice of o_3. Moreover, every surviving element e satisfies

$$\Delta(e \mid I) \leq \Delta(e \mid \{o_1, o_2\}) \leq \tau.$$

The three displayed maximizing choices and diminishing returns also imply

$$f(\{o_1\}) \geq \Delta(o_2 \mid \{o_1\}) \geq \tau,$$

and hence

$$\tau \leq \frac{1}{3} f(I).$$

Let $h(T) = f(I \cup T) - f(I)$, and let A be the residual set constructed by density greedy before its last element e° is discarded. If the residual budget is reached, the preceding discrete density analysis applies to h, budget B', and optimum $O \setminus I$. If instead the surviving elements are exhausted first, then $A \supseteq O \setminus I$, and monotonicity gives an even stronger bound. Hence

$$h(A) \geq \left(1 - \frac{1}{e}\right) h(O \setminus I) = \left(1 - \frac{1}{e}\right) \big(f(O) - f(I)\big).$$

The algorithm always returns the residual set $A^- = A \setminus \{e^\circ\}$. The filtering step and diminishing returns give

$$h(A) - h(A^-) = \Delta(e^\circ \mid I \cup A^-) \leq \Delta(e^\circ \mid I) \leq \tau \leq \frac{1}{3} f(I).$$

Thus discarding the last item is harmless to the claimed factor:

$$\begin{aligned} f(I \cup A^-) &\geq f(I) + \left(1 - \frac{1}{e}\right) \big(f(O) - f(I)\big) - \frac{1}{3} f(I) \\ &= \left(1 - \frac{1}{e}\right) f(O) + \left(\frac{1}{e} - \frac{1}{3}\right) f(I) \\ &\geq \left(1 - \frac{1}{e}\right) f(O). \end{aligned}$$

This feasible candidate is among those compared by the algorithm. ⊣

2.2.3 A Partition Matroid Constraint

Definition 2.6 (Matroid). A *matroid* is a pair $M = (U, \mathcal{I})$, where U is a finite ground set and $\mathcal{I} \subseteq 2^U$ is a family of *independent sets* satisfying:

1. $\emptyset \in \mathcal{I}$;
2. if $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$;
3. if $A, B \in \mathcal{I}$ and $|A| < |B|$, then some $e \in B \setminus A$ satisfies $A \cup \{e\} \in \mathcal{I}$.

For $X \subseteq U$, the *rank* of X in M is

$$r_M(X) = \max\{|I| : I \subseteq X,\ I \in \mathcal{I}\}.$$

Thus $r_M(X)$ is the maximum number of elements of X that can be chosen independently. The rank function $r_M : 2^U \to \mathbb{Z}_{\geq 0}$ is monotone and submodular.

A *matroid constraint* requires the chosen set to belong to $\mathcal{I}$. The cardinality constraint $|S| \leq k$ is the uniform matroid. Another special case is a *partition matroid*: partition U into classes $U_1, \dots, U_q$, assign a capacity b_j to each class, and declare

$$\mathcal{I} = \{S \subseteq U : |S \cap U_j| \leq b_j \text{ for every } j \in [q]\}.$$

Now let $\chi : U \to [q]$ assign a color to each element, with every color used at least once. A set is *color-feasible* if it contains at most one element of each color:

$$|\{e \in S : \chi(e) = j\}| \leq 1 \qquad (j \in [q]).$$

This is precisely the partition matroid above with $U_j = \{e \in U : \chi(e) = j\}$ and $b_j = 1$. Given a value oracle for a nonnegative monotone submodular function $f : 2^U \to \mathbb{R}_{\geq 0}$, the objective is

$$\max\{f(S) : S \text{ is color-feasible}\}.$$

Algorithm 5 Greedy submodular maximization under a partition matroid

Require: A value oracle for $f : 2^U \to \mathbb{R}_{\geq 0}$ and a color map $\chi : U \to [q]$

$A \leftarrow \emptyset$
for $t = 1, \dots, q$ **do**
    Choose $e \in U \setminus A$ maximizing $\Delta(e \mid A)$, subject to $A \cup \{e\}$ being color-feasible
    $A \leftarrow A \cup \{e\}$
end for
return A

Theorem 2.7 (Greedy under a partition matroid). *Let A be the set returned by algorithm 5, and let O be an optimal color-feasible set. Then*

$$f(A) \geq \frac{1}{2} f(O).$$

Proof. By monotonicity, extend O, if necessary, so that it contains one element of every color. Let g_i be the element chosen by greedy in iteration i, and write $G_i = \{g_1, \dots, g_i\}$, with $G_0 = \emptyset$. For

each i, let o_i be the element of O having the same color as g_i. This color does not occur in G_{i-1}, so $G_{i-1} \cup \{o_i\}$ is feasible. The greedy choice therefore gives

$$\Delta(g_i \mid G_{i-1}) \geq \Delta(o_i \mid G_{i-1}).$$

Monotonicity, submodularity, and diminishing returns now give

$$\begin{aligned} f(O) &\leq f(G_q \cup O) \\ &\leq f(G_q) + \sum_{i=1}^{q} \Delta(o_i \mid G_q) \\ &\leq f(G_q) + \sum_{i=1}^{q} \Delta(o_i \mid G_{i-1}) \\ &\leq f(G_q) + \sum_{i=1}^{q} \Delta(g_i \mid G_{i-1}) \\ &= 2f(G_q) - f(\emptyset) \leq 2f(G_q). \end{aligned}$$

Since $G_q = A$, the result follows. ⊣

Example 2.8 (Tightness). Let $U = \{a, b, c\}$, with a, b of one color and c of another. Consider two unit-weight features x, y: element a covers x, while both b and c cover y. Thus $f(S)$ is the number of features covered by S. This is a monotone submodular coverage function, with

$$f(\{a\}) = f(\{b\}) = f(\{c\}) = 1, \qquad f(\{a, c\}) = f(\{a, b, c\}) = 2, \qquad f(\{b, c\}) = 1.$$

Greedy may break its initial tie in favor of b, after which it selects c and returns value 1. The feasible set $\{a, c\}$ has value 2, so the factor $1/2$ is attained.

2.3 Fractional Extensions

to do

Chapter 3

Configuration Linear Programs

A configuration linear program uses one variable for each complete combinatorial choice, such as a machine's schedule or an agent's bundle of items. This creates exponentially many variables, but it preserves information that item-by-item variables would lose. Nash social welfare provides a clean example: whole bundles are the natural choices, dual prices lead to a bundle-optimization oracle, and section 3.3.1 lets us use that oracle without writing down the entire LP.

3.1 Nash Social Welfare

Let N be a set of agents and M a set of indivisible items. Agent i has a valuation $v_i : 2^M \to \mathbb{R}_{\geq 0}$. An allocation $\mathcal{A} = (A_i)_{i \in N}$ assigns pairwise disjoint bundles:

$$A_i \subseteq M, \qquad A_i \cap A_{i'} = \emptyset \quad (i \neq i').$$

Items need not all be allocated. Given weights $w_i > 0$ with $\sum_i w_i = 1$, the *weighted Nash social welfare* is

$$\mathrm{NSW}(\mathcal{A}) = \prod_{i \in N} v_i(A_i)^{w_i}.$$

For equal weights this is the geometric mean of the agents' values. The product balances efficiency and fairness: for example, utility vectors $(10, 1)$ and $(6, 4)$ have total values 11 and 10, respectively, but products 10 and 24. Nash welfare prefers the second outcome because its modest loss in total value is outweighed by its better balance.

Because every $w_i > 0$, an allocation has positive Nash welfare exactly when $v_i(A_i) > 0$ for every agent. On such allocations, taking logarithms gives the equivalent objective

$$\log \mathrm{NSW}(\mathcal{A}) = \sum_{i \in N} w_i \log v_i(A_i). \tag{3.1}$$

If no positive-welfare allocation exists, the optimum Nash welfare is zero and must be handled separately. We henceforth assume that a positive-welfare allocation exists.

Two important valuation classes are:

$$\text{additive:} \quad v_i(S) = \sum_{j \in S} v_{ij}, \qquad \text{submodular:} \quad v_i \text{ has diminishing returns.}$$

3.2 The Configuration LP

For each agent i, let

$$C_i = \{S \subseteq M \,:\, v_i(S) > 0\}$$

be its possible *configurations*. Introduce a variable $y_{i,S}$ for every $S \in C_i$. Integrally, $y_{i,S} = 1$ means that agent i receives exactly S. Relaxing these variables gives

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in N} \sum_{S \in C_i} w_i \log v_i(S)\, y_{i,S} \\ \text{subject to} \quad & \sum_{i \in N} \sum_{\substack{S \in C_i \\ j \in S}} y_{i,S} \le 1 && (j \in M), \\ & \sum_{S \in C_i} y_{i,S} = 1 && (i \in N), \\ & y_{i,S} \ge 0 && (i \in N,\ S \in C_i). \end{aligned} \tag{3.2}$$

The two families of constraints have a simple probabilistic interpretation. For each agent, $y_{i,\cdot}$ is a distribution over bundles, so its total mass is one. The load of item j, obtained by adding the masses of all bundles containing j, is at most one.

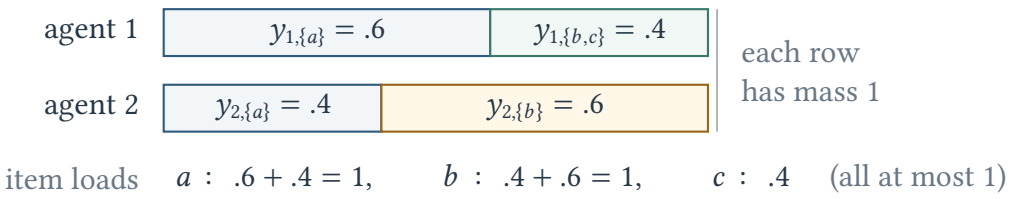

Figure 3.1: A fractional configuration solution. Each agent mixes among whole bundles; item capacities couple the agents' distributions.

Every allocation of positive welfare gives an integral feasible solution by setting $y_{i,A_i} = 1$. Conversely, the bundles selected by any integral feasible solution are pairwise disjoint. Hence the integer version of (3.2) is exactly the allocation problem, while the LP is a relaxation and its optimum is an upper bound on the optimal log Nash welfare.

The relaxation does not itself produce an allocation. A fractional solution specifies one marginal distribution for each agent, but these distributions need not be jointly realizable as disjoint bundles. Turning them into an allocation is a separate *rounding* problem. to do

3.3 Dual Prices and Separation

The primal has exponentially many variables but only $|N| + |M|$ structural constraints. Duality reverses this imbalance. Give item j a nonnegative dual price α_j, and give agent i's equality constraint a free variable β_i. The dual is

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in M} \alpha_j + \sum_{i \in N} \beta_i \\ \text{subject to} \quad & \sum_{j \in S} \alpha_j + \beta_i \geq w_i \log v_i(S) && (i \in N,\ S \in C_i), \\ & \alpha_j \geq 0 && (j \in M), \\ & \beta_i \in \mathbb{R} && (i \in N). \end{aligned} \tag{3.3}$$

Write $\alpha(S) = \sum_{j \in S} \alpha_j$. A dual constraint says that the logarithmic value of every bundle is covered by its item prices plus the agent-specific offset. This is directly an upper-bound certificate: multiplying these inequalities by $y_{i,S}$, summing, and using the primal capacities gives

$$\sum_{i,S} w_i \log v_i(S) y_{i,S} \leq \sum_j \alpha_j \underbrace{\sum_{i, S \ni j} y_{i,S}}_{\leq 1} + \sum_i \beta_i \underbrace{\sum_S y_{i,S}}_{=1} \leq \sum_j \alpha_j + \sum_i \beta_i.$$

3.3.1 Optimization from separation

The ellipsoid method, introduced for linear programming by Khachiyan [2], was the first polynomial-time algorithm for LP. Its importance here is that it accesses a polyhedron through separation rather than an explicit list of inequalities.

Separation oracles. A *strong separation oracle* for $P \subseteq \mathbb{R}^d$, queried at z, either certifies $z \in P$ or returns $a \neq 0$ and β such that

$$a^\top x \leq \beta < a^\top z \qquad (x \in P).$$

For an explicitly listed $P = \{x : Ax \leq b\}$, scanning the rows is such an oracle. The useful case is when the inequalities are too numerous to list but a violated one can still be found efficiently.

Shrinking uncertainty. An ellipsoid has the form

$$E(z, Q) = \{x : (x - z)^\top Q^{-1} (x - z) \leq 1\}.$$

Begin with $E_0 \supseteq P$ and query its center. If the center is infeasible, the separating hyperplane discards the half containing it while preserving P. Replace the surviving half by its minimum-volume enclosing ellipsoid and repeat.

Polynomiality. A central cut in dimension d has an enclosing ellipsoid satisfying

$$\mathrm{vol}(E_{k+1}) \le \exp\left(-\frac{1}{2(d+1)}\right)\mathrm{vol}(E_k).$$

If the initial ellipsoid has radius R, while every nonempty feasible region under consideration contains a ball of radius r, then after N failed queries,

$$\mathrm{vol}(E_N) \le \exp\left(-\frac{N}{2(d+1)}\right)\mathrm{vol}(E_0).$$

This contradicts nonemptiness once

$$N > 2d(d+1)\log\frac{R}{r}.$$

For rational input of encoding length L, suitable bounds satisfy $\log R, \log(1/r) = \mathrm{poly}(d, L)$. Perturbation and dimension reduction handle feasible sets with empty interior.

From feasibility to optimization. To optimize $c^{\mathsf{T}}x$, test feasibility of

$$P_\gamma = P \cap \{x : c^{\mathsf{T}}x \ge \gamma\}$$

and search over γ using rational encoding bounds. Consequently, a polynomial-time separation oracle yields polynomial-time optimization, even for a polynomial-dimensional polyhedron with exponentially many implicit inequalities.

The method is mainly theoretical: its iterates may become badly elongated and its precision requires care. Its running time is weakly polynomial because it depends on L; whether general LP admits a strongly polynomial algorithm remains open.

Although (3.3) has exponentially many constraints, the ellipsoid method needs only a separation oracle. At a proposed (α, β), the oracle treats each agent independently and searches for

$$\max_{S \in C_i}\Big\{ w_i \log v_i(S) - \alpha(S)\Big\}. \tag{3.4}$$

If its value exceeds β_i, the maximizing bundle gives a violated dual constraint. Otherwise every constraint for agent i is satisfied. In column-generation language, (3.4) is the *pricing problem*: it looks for a bundle whose value justifies introducing its primal variable.

3.3.2 The bundle oracle

An equivalent view makes the underlying combinatorial problem clearer. For a target value V, ask for the cheapest bundle attaining it:

$$\min\{\alpha(S) : v_i(S) \ge V\}. \tag{3.5}$$

Scanning suitably discretized target values reveals the best tradeoff in (3.4). Reversing the question once more, for a price budget B define

$$\max\{v_i(S) \,:\, \alpha(S) \le B\}. \tag{3.6}$$

The dual variables have therefore turned separation into a familiar single-agent optimization problem.

Additive valuations. If $v_i(S) = \sum_{j \in S} v_{ij}$, then (3.6) is an ordinary knapsack problem: the dual prices are costs and the v_{ij} are profits. Rounding profits geometrically and applying dynamic programming gives an FPTAS-style separation oracle. Thus the configuration LP can be approximated without enumerating all bundles, even though its additive valuation formula still contains exponentially many variables.

Submodular valuations. If v_i is monotone submodular, then (3.6) is exactly monotone submodular maximization under a knapsack constraint. The algorithm in algorithm 4 supplies the approximate bundle oracle needed here. If it returns a bundle of value at least $q = 1 - 1/e$ times the best value under budget B, then its logarithmic value loses at most

$$w_i \log \frac{1}{q} \le \log \frac{e}{e-1}.$$

After the target values or budgets are discretized, this becomes an approximate separation oracle with an additional arbitrarily small error. This is the precise dependence between the two chapters: configuration LP is the framework, while submodular knapsack maximization implements its oracle for submodular valuations.

3.4 Recovering a Small Primal LP

The ellipsoid method makes only polynomially many oracle calls. Record every violated bundle constraint (i, S) returned during its execution, and let $\mathcal{K}$ be this polynomial-size collection. A dual constraint indexed by (i, S) corresponds to exactly one primal variable $y_{i,S}$. Standard optimization–separation equivalence therefore recovers a primal optimum by solving the restriction of (3.2) to the variables indexed by $\mathcal{K}$; with an approximate oracle, the same construction recovers the corresponding approximate LP solution.

Thus exponential size is not by itself fatal. What matters is whether the dual can recognize a violated bundle efficiently:

$$\boxed{\text{configuration LP}} \longrightarrow \boxed{\text{dual separation}} \longrightarrow \boxed{\text{single-agent bundle optimization}}.$$

For Nash social welfare, solving this relaxation and rounding it are distinct steps. The configuration LP organizes the global competition for items; the oracle exploits the structure of each agent's valuation; a rounding argument must finally coordinate the agents into one integral allocation.

Bibliography

[1] George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13(1):339–347, 1951.

[2] Leonid G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244(5):1093–1096, 1979.

[3] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.

[4] Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.