NOTES ON

Combinatorial Optimization

Hao Su

July 30, 2026

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:

$$
\text{(A) maximize } \sum_{j=1}^{n} c_j x_j \qquad\qquad \text{and (B) maximize } c^\top x
$$

$$
\begin{aligned}
\text{subject to } \sum_{j=1}^{n} a_{ij} x_j &\le b_i \quad (i = 1, \dots, m) & \qquad \text{subject to } Ax &\le b \\
x_j &\ge 0 \quad (j = 1, \dots, n) & x &\ge 0.
\end{aligned}
$$

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 an intersection of half-spaces, called a *polyhedron*; its corners are called *vertices*. 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

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

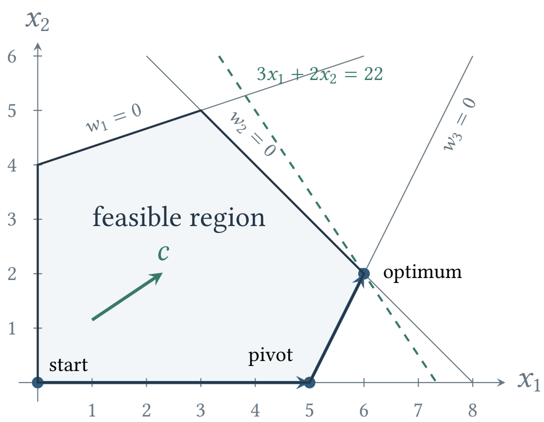

Figure 1.1: The simplex method follows edges between adjacent vertices.

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:

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

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

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

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

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

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 $w_i = b_i - a_i^\top x + x_0$. 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 $\zeta = c^\top x$ 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.

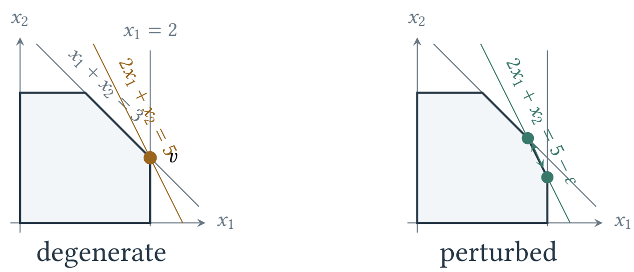

Figure 1.2: Three boundaries meet at a degenerate vertex (left). An infinitesimal perturbation separates it into nearby nondegenerate vertices (right).

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 ∈ N with c̄_k > 0 do
    k ← min{j ∈ N : c̄_j > 0}
    if ā_ik ≤ 0 for every i ∈ B then
        return UNBOUNDED and the ray (1.5)
    end if
    θ ← min_{i∈B: ā_ik>0} b̄_i/ā_ik
    ℓ ← min{i ∈ B : ā_ik > 0, b̄_i/ā_ik = θ}
    Pivot on row ℓ and column k; exchange k and ℓ
end while
return x_N = 0, x_i = b̄_i for i ∈ B, and value ζ̄
```

> **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 $a^\top x \le \beta$ is *valid* for a polyhedron P if every point of P satisfies it. When the hyperplane $a^\top x = \beta$ meets P, it is a *supporting hyperplane*, and

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

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. $\dashv$

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

$$\{x \in P \;:\; c^\top x = \zeta^*\}.$$

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

1.3 Duality

1.3.1 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

$$y^\top A x \leq y^\top b.$$

If also $A^\top y \geq c$, then $x \geq 0$ implies $c^\top x \leq y^\top A x$. Seeking the best such bound produces the dual pair

$$(\mathrm{P}) \quad \max_{x \geq 0}\{c^\top x \;:\; Ax \leq b\} \qquad\qquad (\mathrm{D}) \quad \min_{y \geq 0}\{b^\top y \;:\; A^\top y \geq c\}. \tag{1.12}$$

The same upper-bound argument handles other constraint and variable signs directly. A row $a_i^\top x \leq b_i$ may be added with multiplier $y_i \geq 0$, because nonnegative multiplication preserves its direction; a $\geq$ row instead requires $y_i \leq 0$, while an equality permits either sign. After the rows are combined, comparison with $c^\top x$ similarly requires the resulting coefficient of x_j to be at least, equal to, or at most c_j according as $x_j \geq 0$, x_j is free, or $x_j \leq 0$. Thus the general dual is governed by exactly the same valid-bound construction; in particular, a flow-conservation equality naturally receives an unrestricted dual price.

After adding the primal slacks, the same pair takes the equality-form representation

$$(\widehat{\mathrm{P}}) \quad \max_{u \geq 0}\{\widehat{c}^\top u \;:\; \widehat{A} u = b\} \qquad\qquad (\widehat{\mathrm{D}}) \quad \min_{y \in \mathbb{R}^m}\{b^\top y \;:\; \widehat{A}^\top y \geq \widehat{c}\}. \tag{1.13}$$

The variable y is free in $(\widehat{\mathrm{D}})$, because it corresponds to equality constraints. There is no separate sign restriction on it; dual feasibility is the single condition $\widehat{A}^\top y \geq \widehat{c}$. Expanding the slack block shows that this condition is

$$\begin{bmatrix} A^\top \\ I_m \end{bmatrix} y \geq \begin{bmatrix} c \\ 0 \end{bmatrix} \quad \iff \quad A^\top y \geq c, \qquad y \geq 0.$$

Thus the sign condition has not vanished; it is encoded by the dual constraints belonging to the nonnegative primal slack variables. Let

$$w = b - Ax \geq 0, \qquad z = A^\top y - c \geq 0$$

denote the primal and dual slacks.

Theorem 1.7 (Weak duality)**.** *Every primal-feasible x and dual-feasible y satisfy $c^\top x \le b^\top y$.*

Proof. The *duality gap* decomposes into nonnegative complementary products:

$$b^\top y - c^\top x = y^\top(b - Ax) + x^\top(A^\top y - c) = y^\top w + x^\top z \ge 0. \tag{1.14}$$

⊣

Thus equal primal and dual values certify optimality without replaying an algorithm. Weak duality also implies that primal unboundedness forces dual infeasibility, and dual unboundedness forces primal infeasibility.

Theorem 1.8 (Strong duality)**.** *If either problem in* (1.12) *has a finite optimal solution, then so does the other, and their optimal values are equal.*

Proof. Suppose first that the primal has an optimum. By the theorem 1.4, simplex terminates at an optimal basis of the augmented formulation (1.9). Seek a dual upper bound that is exact on the basic variables. This forces their dual constraints to be tight:

$$B^\top y = \widehat{c}_{\mathcal{B}}, \qquad \text{so} \qquad y = B^{-\top}\widehat{c}_{\mathcal{B}}.$$

Because $Bu_{\mathcal{B}} = b$, this choice already makes the bound exact at the current basic solution:

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

It remains only to check that the bound is globally valid. On the nonbasic columns, the simplex optimality condition is exactly

$$\bar{c}_{\mathcal{N}} = \widehat{c}_{\mathcal{N}} - N^\top y \le 0 \quad \Longleftrightarrow \quad N^\top y \ge \widehat{c}_{\mathcal{N}}.$$

Hence every column constraint holds: tightly on the basic columns and weakly on the nonbasic ones. Therefore $\widehat{A}^\top y \ge \widehat{c}$, which is precisely dual feasibility for $(\widehat{\mathrm{D}})$ in (1.13). This dual-feasible solution has the same value as the primal solution, so weak duality makes both optimal. The converse follows by symmetry. ⊣

There are therefore four possible status patterns: both problems have equal finite optima; the primal is unbounded and the dual infeasible; the dual is unbounded and the primal infeasible; or both are infeasible.

1.3.2 Complementary slackness

Theorem 1.9 (Complementary slackness). *Let x and y be primal and dual feasible. They are optimal if and only if*

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

Proof. By (1.14), the feasible pair is optimal exactly when $y^\top w + x^\top z = 0$. Every summand is nonnegative, so the sum vanishes exactly when all the displayed products vanish. ⊣

Thus at an optimal primal-dual feasible pair, 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 an optimality proof into a structural matching between positive primal objects and tight dual inequalities.

1.3.3 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 the invariant $\bar{c} \leq 0$ at every pivot. If a negative basic row has no negative coefficient, that row remains negative for every nonnegative choice of the nonbasic variables and therefore certifies infeasibility. Bland's rule or lexicographic perturbation prevents cycling exactly as for primal simplex, so the algorithm terminates with either this certificate or a basis with $\bar{b} \geq 0$, which is optimal.

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

1.3.4 Lagrangian duality

Dual variables can also be understood as penalties for violating constraints. For the primal problem in (1.12), define

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

For a feasible x, the second term is nonnegative, so $L(x, y) \geq c^{\top} x$. Hence

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

is an upper bound on the primal optimum for every $y \geq 0$. Since

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

we have

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

Minimizing $g(y)$ over $y \geq 0$ is exactly the LP dual. Weak duality says that every multiplier gives a valid bound; strong duality says that for LP the best bound is exact.

The same construction remains useful when x is restricted to a discrete set. One removes difficult constraints, prices their violation with multipliers, and solves the resulting *Lagrangian relaxation*. The relaxed optimum supplies a bound and often decomposes into simpler combinatorial subproblems. Unlike linear programming duality, however, the best Lagrangian bound for a discrete problem may have a nonzero duality gap.

1.4 Convex and Polyhedral Separation

1.4.1 Convex sets and convex hulls

Definition 1.10 (Convexity). A set $C \subseteq \mathbb{R}^d$ is *convex* if $\lambda x + (1-\lambda)y \in C$ whenever $x, y \in C$ and $0 \leq \lambda \leq 1$. A *convex combination* of $x_1, \dots, x_r$ is a point

$$\sum_{i=1}^{r} \lambda_i x_i, \qquad \lambda_i \geq 0, \qquad \sum_{i=1}^{r} \lambda_i = 1.$$

The *convex hull* $\operatorname{conv}(S)$ is the smallest convex set containing S.

A set is convex if and only if it contains every finite convex combination of its points. Consequently,

$$\operatorname{conv}(S) = \left\{ \sum_{i=1}^{r} \lambda_i x_i \;:\; r \geq 1,\ x_i \in S,\ \lambda \geq 0,\ \mathbf{1}^\top \lambda = 1 \right\}. \tag{1.15}$$

Intersections of convex sets are convex; in particular, every polyhedron is convex.

Theorem 1.11 (Carathéodory). *Every point of* $\operatorname{conv}(S)$, *where* $S \subseteq \mathbb{R}^d$, *is a convex combination of at most* $d+1$ *points of* S.

Proof. By (1.15), write $x = \sum_{i=1}^{r} \lambda_i x_i$ with $x_i \in S$. Regard λ as a feasible solution of

$$\sum_{i=1}^{r} \lambda_i x_i = x, \qquad \sum_{i=1}^{r} \lambda_i = 1, \qquad \lambda \geq 0.$$

This system has at most $d+1$ independent equalities. By theorem 1.4 it has a basic feasible solution, which has at most $d+1$ positive components. Those components give the desired representation. ⊣

Carathéodory's theorem says that convex representations in fixed dimension always have sparse witnesses. The number $d+1$ is best possible, as witnessed by a point in the relative interior of a d-simplex.

1.4.2 Farkas certificates

Theorem 1.12 (Farkas' lemma). *Exactly one of the following systems has a solution:*

1. $Ax \le b$,
2. $A^\top y = 0, y \ge 0, b^\top y < 0$.

Proof. The two systems cannot both be feasible: multiplying $Ax \le b$ by $y \ge 0$ would give $0 = x^\top A^\top y \le b^\top y < 0$.

For the converse, view $Ax \le b$ as the feasibility LP with zero objective. Its dual is

$$\min\{b^\top y \,:\, A^\top y = 0,\ y \ge 0\},$$

which is feasible at $y = 0$. If the primal system is infeasible, strong duality rules out a finite dual optimum. The dual is therefore unbounded below, so it contains a feasible y with $b^\top y < 0$. ⊣

The vector y is a short, checkable certificate: it combines the proposed inequalities into the contradiction $0 < 0$. Many combinatorial min–max theorems can be viewed as structured instances of this alternative.

1.4.3 Separation

For two sets $C, D \subseteq \mathbb{R}^d$, a nonzero vector a *weakly separates* them if

$$\sup_{x \in C} a^\top x \le \inf_{y \in D} a^\top y.$$

The separation is *strong* when the inequality is strict. Geometrically, level sets of $a^\top x$ place the two sets on opposite sides of a hyperplane; strong separation leaves a positive gap.

Closedness of both sets alone does not guarantee a positive gap when they are unbounded. For example,

$$\{(s,t) \,:\, t \ge e^s\} \quad \text{and} \quad \{(s,0) \,:\, s \in \mathbb{R}\}$$

are disjoint closed convex sets whose distance is zero.

Theorem 1.13. *Any two disjoint nonempty polyhedra can be strongly separated.*

Proof. Write $P = \{x \,:\, Ax \le b\}$ and $Q = \{x \,:\, Cx \le d\}$. Their disjointness says that the combined system is infeasible. By theorem 1.12, there are $y, z \ge 0$ such that

$$A^\top y + C^\top z = 0, \qquad b^\top y + d^\top z < 0.$$

Put $a = A^\top y = -C^\top z$, $\alpha = b^\top y$, and $\beta = -d^\top z$. Then

$$a^\top x \le \alpha < \beta \le a^\top q \qquad (x \in P,\ q \in Q).$$

Moreover $a \neq 0$: otherwise nonemptiness of P and Q would imply $b^\top y \ge 0$ and $d^\top z \ge 0$, contradicting their negative sum. Hence the displayed inequalities give a genuine separating hyperplane. ⊣

For polyhedra, separation is algorithmic as well as geometric: a point outside P can be excluded by a valid inequality. Finding such inequalities is the *separation problem*, a central viewpoint in polyhedral combinatorics.

1.4.4 The ellipsoid method

The ellipsoid method, introduced for linear programming by Khachiyan [2], was the first polynomial-time algorithm for LP.

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

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

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

Shrinking uncertainty. An ellipsoid has the form

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

Begin with $E_0 \supseteq P$ and query its center. If the center is infeasible, the oracle discards the half containing it; the surviving half contains P. Replace that half by its minimum-volume enclosing ellipsoid and repeat.

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

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

Suppose the initial ellipsoid is a ball of radius R, while every nonempty feasible region under consideration contains a ball of radius r. After N failed queries,

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

This becomes smaller than the volume of the radius-r ball once

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

The system must then be infeasible. For rational input of encoding length L, one can choose bounds with $\log R, \log(1/r) = \mathrm{poly}(d, L)$; hence only polynomially many oracle queries are needed. Standard perturbation and dimension reduction handle feasible sets with empty interior.

Reduction from optimization. Test feasibility of

$$P_\gamma = P \cap \{x \,:\, c^\top x \geq \gamma\}$$

and search over γ, using the rational encoding bounds.

Exponentially many inequalities. Suppose

$$P = \{x \in \mathbb{R}^d \,:\, a_s^\top x \leq b_s \text{ for every } s \in \mathcal{S}\},$$

where d is polynomial but $\mathcal{S}$ may be exponentially large. A seperation oracle that can either verify all the inequalities at a query point or find a violated index s in polynomial time is desirable in this case.

The ellipsoid method is mainly theoretical: its iterates can become badly elongated, precision must be controlled carefully, and simplex or interior-point methods (to do) are usually much faster in practice. Its running time is also only *weakly polynomial*, since it depends on L. Whether general LP admits a strongly polynomial algorithm, independent of coefficient magnitudes, remains a major open problem.

1.4.5 Strict complementarity

Ordinary complementary slackness permits both members of a complementary pair to vanish. Nevertheless, an optimal pair can always be chosen to avoid this.

> **Theorem 1.14** (Strict complementarity). *If the primal and dual have optimal solutions, then they have optimal solutions (x^*, w^*) and (y^*, z^*) satisfying*
>
> $$x^* + z^* > 0, \qquad y^* + w^* > 0.$$
>
> *Together with complementary slackness, exactly one member of each pair (x_j^*, z_j^*) and (y_i^*, w_i^*) is positive.*

Geometrically, one may choose the primal and dual solutions in the relative interiors of their optimal faces. Strict complementarity is an existence theorem: an arbitrary optimal pair need not be strict, and the theorem does not say that the optimal basis is unique or nondegenerate.

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 ← ∅
for t = 1, …, k do
    Choose e ∈ U \ S maximizing Δ(e | S)
    S ← S ∪ {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 ← ∅
for t = 1, …, q do
    Choose e ∈ U ∖ A maximizing Δ(e | A), subject to A ∪ {e} being color-feasible
    A ← A ∪ {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

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.