

Linear Programming

Contents

1	Heuristics	1
1.a	General framework	1
1.b	Constructive heuristics	1
1.c	Local search	2
1.d	Metaheuristics	2
1.e	Practical considerations	2
2	Modeling with linear functions	2
2.a	Polyhedra	2
2.b	Linear optimization problems	3
2.c	Where to find an optimum?	3
3	Solution algorithms for LPs	4
3.a	Simplex algorithm	4
3.a.a	General idea	4
3.a.b	Reduced costs	4
3.b	Ellipsoid and interior point methods	5
4	Lagrangian duality	5
5	Total unimodularity	6
5.a	When MILPs become easy to solve	6
5.b	Total Unimodularity	6
6	Homework	7

1 Heuristics

1.a General framework

A heuristic is a problem-specific algorithm that returns a solution

- rather quickly
- of decent quality
- without optimality guarantee

How to evaluate it? Mostly experimental.

1.b Constructive heuristics

Create a solution from nothing. Often greedy: while solution not complete, add the next most convenient item.

Examples:

- TSP: nearest neighbor, best insertion
- Knapsack: largest value per kg
- Scheduling: earliest deadline

Other constructive heuristics are based on a relaxation.

Exercise: Vertex cover through ILP rounding. (1) Propose a heuristic to solve the minimum vertex cover from the LP relaxation of its ILP formulation. (2) Prove that this actually yields a 2-approximation algorithm. (3) Turn it into an iterative procedure for vertex addition.

1.c Local search

Improve a solution with small moves.

Define a neighborhood function $\mathcal{N}(x)$ which lists all solutions that can be attained from x with one move. In a neighborhood, pick either the first improvement or the best improvement.

Examples:

- TSP: vertex exchange, 2-opt
- Knapsack: exchange 2 items

When the neighborhood is too big: Large Neighborhood Search uses mathematical programming to explore it.

1.d Metaheuristics

- Iterated local search (random perturbation)
- Variable neighborhood search (increasing radius)
- Simulated annealing (temperature determines acceptance)
- Evolutionary algorithms (analogy with genetics & natural selection)
- Tabu search (avoid last visited solutions)

1.e Practical considerations

- Solution encoding & evaluation
 - make small changes easy to apply and evaluate
- Handling constraints:
 - reject infeasible solutions
 - penalize infeasible solutions
 - repair them

2 Modeling with linear functions

2.a Polyhedra

A polyhedron P is a finite intersection of closed half-spaces:

$$P = \bigcap_{i \in [m]} \{x \in \mathbb{R}^n : a_i x \leq b_i\} = \{x \in \mathbb{R}^n : Ax \leq b\} \quad (1)$$

Here the inequality $Ax \leq b$ must be understood entrywise.

A polytope is a bounded polyhedron.

We say that $x \in P$ is a vertex (or extreme point) of the polyhedron if x cannot be written as a non-trivial mean of two other points of P :

$$x = \frac{y+z}{2} \quad \text{with} \quad y, z \in P \implies y = z = x \quad (2)$$

2.b Linear optimization problems

A Linear Program is an optimization problem with linear objective and affine constraints:

$$\min_{x \in \mathbb{R}^n} c^\top x \quad \text{s.t.} \quad Ax \leq b \iff \min_{x \in P} c^\top x \quad (3)$$

In standard form, it is written

$$\min_x c^\top x \quad \text{s.t.} \quad Ax = b, x \geq 0 \quad (\text{LP}) \quad (4)$$

and we assume without loss of generality that $\text{rank}(A) = m \leq n$ (no redundant constraints).

Exercise (Meunier - PL1): Prove that both forms above are equivalent.

A base $B \subset [n]$ is a subset of m linearly independent columns of A : the submatrix $A_B = (a_{i,j})_{i \in [m], j \in B}$ is nonsingular. Every base defines a basic solution of (LP)

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} \quad \text{with} \quad \begin{cases} x_B = A_B^{-1}b \\ x_N = 0 \end{cases} \quad (5)$$

which is called basic feasible if $A_B^{-1}b \geq 0$.

2.c Where to find an optimum?

Theorem (Interpretation of bases): x is a basic feasible solution of (LP) iff it is a vertex of the polyhedron P .

Proof:

- Suppose x is a feasible basic solution. There is a basis B such that $x_B = A_B^{-1}b \geq 0$ and $x_N = 0$. In addition, since $Ay = A_B y_B = b$ we see that $y_B = A_B^{-1}b = x_B$ and the same goes for z_B . Therefore, $y = z = x$.
- Now suppose that x is a vertex of P . Let $K = \{j \in [n] : x_j > 0\}$. Let $d_K \in \ker A_K$ and $d \in \mathbb{R}^n$ equal to d_K on K and equal to 0 elsewhere. Then for $\epsilon > 0$ small enough, $x \pm \epsilon d \in P$, which means that $d = 0$. Thus A_K has independent columns, and since $\text{rank} A = m$, K can be completed into a set B of size m such that A_B is invertible. As a consequence, B is a basis and $x = (x_B, x_N)$ with $x_N = 0$ and $x_B = A_B^{-1}b \geq 0$. ■

Theorem (Existence of an optimum): If $P \neq \emptyset$ and $\inf_{x \in P} c^\top x > -\infty$, then (LP) has an optimal solution.

Proof (for the bounded case): If P is a polytope, P is compact and the result follows. ■

Theorem (Optimal basis): If (LP) has an optimal solution, then at least one of the optimal solutions is basic (feasible).

Proof (for the bounded case): We admit the Minkowski-Weyl theorem (every polytope is the convex hull of its vertices). Let x be an optimal solution of (LP), then $x = \sum_k \lambda_k v_k$ where the v_k are the vertices of P and the $\lambda_k \geq 0$ sum to 1. By concavity of $x \mapsto c^\top x$, we know that $c^\top x \geq \sum_k \lambda_k (c^\top v_k)$, hence there is a k such that $c^\top x \geq c^\top v_k$. Then v_k is an optimal solution, and it is basic feasible. ■

Exercise (Meunier - PL6): Computing sums of order statistics. For any vector $x \in \mathbb{R}^n$, we denote by $x^{[i]}$ the i -th largest element of the set $\{x_1, \dots, x_n\}$. For instance, $x^{[1]}$ is the largest component of x , and $x^{[n]}$ is the smallest.

1. Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We fix an integer $k \in [n]$. Show that the problem

$$\min \sum_{i=1}^k x^{[i]} \quad \text{s.t.} \quad Ax = b, x \in \mathbb{R}_+^n \quad (6)$$

can be formulated as a Linear Program. Hint: introduce one constraint per subset $S \subset [n]$ of size k .

2. (Harder) Show that there is a formulation with at most $m + n$ constraints.

3 Solution algorithms for LPs

Naive algorithm: enumerate all bases, discard the infeasible ones and compare the feasible ones. Unfortunately, this method has exponential runtime.

3.a Simplex algorithm

3.a.a General idea

Starting from a given vertex, try to find one of its neighbors with smaller cost. The way we choose the neighbor is called a pivot rule.

Theorem: There exists a pivot rule such that the simplex doesn't cycle.

To find the initial vertex, we run a first simplex on an auxiliary problem which is easier to initialize.

3.a.b Reduced costs

Going from one vertex to the next is like going from one basis to the next. We can rewrite the constraints of (LP) by decomposing x along the current basis B :

$$Ax = b \iff A_B x_B + A_N x_N = b \iff x_B = A_B^{-1}(b - A_N x_N) \quad (7)$$

Starting from B , we rewrite the objective as:

$$\begin{aligned}
c^\top x &= c_B^\top x_B + c_N^\top x_N \\
&= c_B^\top (A_B^{-1}(b - A_N x_N)) + c_N^\top x_N \\
&= \underbrace{c_B^\top A_B^{-1} b}_{\text{cost of the basic solution}} + \underbrace{\left(c_N - A_N^\top (A_B^{-1})^\top c_B \right)^\top}_{\text{reduced cost } r_N^\top} x_N \\
&= \text{cost}(\text{sol}(B)) + r_N^\top x_N
\end{aligned} \tag{8}$$

Proposition (Role of the reduced cost): If $r_N \geq 0$, then B is an optimal basis. Otherwise, we can decrease the objective value by increasing x_j where $j \in N$ is such that $r_j < 0$. This either leads to a new basis $B' \subset B \cup \{j\}$ or to a direction of unboundedness.

3.b Ellipsoid and interior point methods

The simplex is efficient in practice, but has exponential worst-case running time for all known pivot rules. Other methods, discovered much later, have a much better theoretical complexity (polynomial even in the worst case): ellipsoid algorithm, interior point algorithms.

In particular, the ellipsoid algorithm runs in polynomial time even with an exponential number of constraints, as long as the separation problem can be solved efficiently.

4 Lagrangian duality

Proposition (Expression of the dual): The dual of the Linear Program in standard form (LP) is:

$$\max_{y \in \mathbb{R}^m} b^\top y \quad \text{s.t.} \quad A^\top y \leq c \tag{LPD} \tag{9}$$

Exercise: Prove it.

Proof: The Lagrangian for (LP) writes:

$$\mathcal{L}(x, \lambda, \mu) = c^\top x + \lambda^\top (b - Ax) + \mu^\top (-x) \tag{10}$$

where $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}_+^n$ are Lagrange multipliers associated with each constraint. The dual problem is defined as

$$\max_{\lambda, \mu \geq 0} \underbrace{\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \mu)}_{\text{dual function } g(\lambda, \mu)} \tag{11}$$

■

Theorem (Weak duality): We always have $\text{val}(\text{LPD}) \leq \text{val}(\text{LP})$.

Theorem (Strong duality in Linear Programming): If either (LP) or (LPD) has a feasible solution, then the values of both problems are equal.

Proof: Based on the termination of the simplex algorithm and a correspondence between the final reduced costs & the optimal dual variables $y^* = A_{B^*}^{-1} b$. ■

Proposition (Complementary slackness): Let x and y be feasible solutions of the primal and the dual respectively. Then x and y are both optimal solutions iff

$$(c - A^\top y)^\top x = 0 \quad \text{and} \quad y^\top (b - Ax) = 0 \quad (12)$$

Proof: Once we note that

$$y^\top (b - Ax) \leq 0 \leq x^\top (c - A^\top y) \quad (13)$$

the result follows directly from strong duality. ■

Remark: The dual variables have an economic interpretation. In the standard form LP, we replace the constraint $Ax = b$ by $Ax = b + \varepsilon$, then the value of the problem will change by approximately $-(y^*)^\top \varepsilon$, where y^* is the optimal dual solution.

5 Total unimodularity

5.a When MILPs become easy to solve

A polyhedron P is integral if $P = \text{conv}(P \cap \mathbb{Z}^n)$. In particular, all its vertices have integer coordinates, which means that

- any LP with P as feasible set has one integral optimal solution
- the simplex algorithm finds such a solution

This implies that solving a MILP is just as easy as solving its continuous relaxation.

5.b Total Unimodularity

A matrix A is Totally Unimodular (TU) if the determinant of all its square submatrices (obtained by removing rows and/or columns) is in $\{-1, 0, 1\}$. In particular, all of its coefficients must be in $\{-1, 0, 1\}$ too.

Theorem (Importance of TU matrices): If A is TU and $b \in \mathbb{Z}^m$, then $P = \{Ax = b, x \geq 0\}$ is integral.

Proof: For any vertex x of P , there is a basis B such that x is the basic feasible solution associated with B . This implies that $x_N = 0$ and $x_B = A_B^{-1}b \geq 0$. Since A is TU and A_B is invertible, the determinant of A_B is in $\{-1, 1\}$. By the comatrix formula, $A_B^{-1} = \frac{1}{\det A_B} \text{com}(A_B)^\top$, which means A_B^{-1} has integer coefficients. Since b does too, we conclude that $x \in \mathbb{Z}^n$. ■

Proposition (Poincaré criterion for TU): Let $A \in \{-1, 0, 1\}^{m \times n}$. If A contains at most one 1 and one -1 per column, then A is TU.

Proof: By recursion. ■

Corollary (Integrality of the flow polyhedron): The incidence matrix of a directed graph is TU.

How to build new TU matrices? If A is TU, then the following matrices are too: $A^\top$, $-A$, $(A, \pm A)$, (A, e_j) , any matrix obtained multiplying a row or column of A by -1 .

6 Homework

Exercise: Prove that min cut is the dual of max flow.

Exercise (Meunier - course notes): Nash equilibria in zero-sum games