Linear Programming and Integer Linear Programming

Kurt Mehlhorn

May 26, 2013
revised, May 20, 2014

1 Introduction

A linear program (LP) is the task of optimizing a linear function under linear constraints. Formally, it is an optimization problem of the form

$$\begin{aligned} &\max c^T x \\ &\text{subject to } Ax \leq b \\ &\qquad\qquad x \geq 0, \end{aligned}$$

where A is an $m \times n$ matrix, b is an m vector, and c is an n vector. The entries of A and the vectors b and c are integral. x is an n vector of real unknowns. Each of the $m+n$ inequalities in $Ax \leq b$ and $x \geq 0$ is called a constraint. Geometrically, each constraint is a half-space in $\mathbb{R}^n$. The term linear in linear program refers to the fact that the objective function and the constraints are linear functions of the variables.

The set $P := \{x; Ax \leq b \text{ and } x \geq 0\}$ is called the *set of feasible solutions*. An LP is *feasible* if the set of feasible solution is non-empty. The set of feasible solutions is a convex set as it is the intersection of half-spaces. Intersections of half-spaces are usually called polyhedra. A feasible LP is *bounded* if the set $\{c^T x; x \text{ is feasible}\}$ is bounded from above. If an LP is bounded, there is an optimal feasible x, namely an x^* with x^* is feasible and $c^T x^* \geq c^T x$ for every feasible x. A feasible LP is *unbounded* if the set $\{c^T x; x \text{ is feasible}\}$ contains arbitrarily large values.

We will see that the language of linear programs is very expressive (Section 3). Every LP has a dual LP which sheds additional light on the optimization problem (Section 4). There are many algorithms for solving linear programs (Section 5). In particular, the Simplex and the Interior Point algorithms are of great practical importance, and the Ellipsoid method is of great theoretical importance. LPs can be solved in polynomial time in n, m, and $\log L$, where L is the absolute value of the largest entry of A, b, and c. It is open, whether there is a strongly polynomial algorithm, i.e., an algorithm whose running time is polynomial in n and m. *Integer linear programs (ILP)* are linear programs with the additional constraint that x is integral (Section 6. Integer linear programming is NP-complete. If the number of variables is fixed, integer linear programming is in P. Some LPs are guaranteed to have optimal integral solutions. The linear programming relaxation of an ILP is obtained by dropping the integrality constraint on the variables. LP relaxations are a useful tool for solving ILPs exactly and approximately.

In the formulation of LPs, we may allow inequalities and equalities and unconstrained variables. This does not add to the expressive power. For example, an unconstrained variable x may be replaced by $x_p - x_n$ with $x_p \geq 0$ and $x_n \geq 0$. An equality $a^T x = b$ is equivalent to the two inequalities $a^T \geq b$ and $a^T x \leq b$. An inequality $a^T x \leq b$ can be turned into an equality $a^T x + s = b$ by the introduction of a nonnegative slack variable.

Open-source solvers for linear programs and integer linear programs are available.

The books [Sch03] are bibles on linear and integer linear programming. The books are demanding. [Chv93] is an easy going introduction to linear programming. [CCPS98] emphasizes the connection between linear programming and combinatorial optimization.

2 History

Already, Fourier (French Mathematician of the 18th century) investigated the problem of solving systems of linear inequalities. Leonid Kantorovish used linear programs to model logistic problems during World War II. Georg Dantzig published the Simplex method for solving linear programs in 1947 and John von Neumann invented duality theory in the context of game theory. Leonid Khachiyan obtained the first polynomial-time algorithm in 1979 and Narenda Karmakar introduced the interior-point method in 1984. Linear programming solvers usually offer the simplex algorithm and some interior-point method.

3 Expressiveness

Many important problems can be formulated as LPs. We give three examples.

Max Flow: We are given a directed graph $G = (V,E)$ with two special vertices s and t and a non-negative capacity of each edge $cap : E \mapsto \mathbb{R}_{\geq 0}$. We are searching for a flow f from s to t of maximum value. A flow is a function $f : E \mapsto \mathbb{R}_{\geq 0}$ obeying the capacity constraints (the flow across an edge is never more that the capacity of the edge) and flow conservation (the flow into a vertex is equal to the flow out of a vertex) for every vertex distinct from s and t, i.e.,

$$\begin{aligned}
\max \quad & \sum_{e;e=(s,_)} f_e - \sum_{e;e=(_,s)} f_e & \\
\text{subject to} \quad & f_e \leq cap_e & \text{for all } e \\
& \sum_{e;e=(_,v)} f_e = \sum_{e;e=(v,_)} f_e & \text{for all } v \neq s,t \\
& f_e \geq 0 & \text{for all } e
\end{aligned}$$

The value of a flow is the net-flow out of s.

Max Flow, Alternative Formulation: We have a variable f_p for every path p from s to t. Its value is the flow along p. Then flow conservation is guaranteed and we only have to enforce the capacity constraint. Thus

$$\begin{aligned}
\max \quad & \sum_{p} f_p & \\
\text{subject to} \quad & \sum_{p;e\in p} f_e \leq cap_e & \text{for all } e \\
& f_p \geq 0 & \text{for all } p
\end{aligned}$$

Observe that this formulation uses an exponential number of variables; exponential in the graph parameters $n = |V|$ and $m = |E|$.

Shortest Paths: The formulation of a problem as an LP may be counterintuitive at first. The shortest path problem is an example.

We have a directed graph G. Each edge e has a length ℓ_e. We are interested in the shortest distance from s to t. We have a variable d_v for every vertex v. Consider

$$\begin{aligned} \max\ & d_t \\ \text{subject to}\quad & d_v \leq d_u + \ell_e && \text{for every edge } e = (u,v) \\ & d_s \leq 0 \end{aligned}$$

Observe that we formulated the shortest path problem as a maximization problem. Intuitively, this formulation can be understood as follows. Build a physical model of the graph by replacing any edge e by a string of length ℓ_e. Lift up the graph at s and let the other vertices fall down. They will come to rest at some vertical distance below s. This distance is exactly the shortest path distance from s.

Formally, we can argue as follows. Let μ_v be the length of the shortest path from s to v. The vector μ satisfies all inequalities. Conversely, if d is a vector satisfying all inequalities, then $d_v \leq \mu_v$ for all v. This can be proved by induction on k_v, the number of edges in a shortest path from s to v.

Exercise 1 *Show that the LP is infeasible if and only if the graph contains a negative cycle.*

4 Duality

With every linear program

$$\begin{aligned} \max\ & c^T x \\ \text{subject to}\quad & Ax \leq b \\ & x \geq 0, \end{aligned}$$

there is an associated linear program, called its dual. It is a linear program in m variables, which we collect into a vector y.

$$\begin{aligned} \min\ & y^T b \\ \text{subject to}\quad & y^T A \geq c \\ & y \geq 0 \end{aligned}$$

In this context, the original LP is called the primal LP.

Observe that the primal has m constraints $Ax \leq b$ and n non-negativity constraints $x \geq 0$. The dual has one variable for each primal constraint in $Ax \leq b$. The objective function of the dual is determined by the right hand side b. Besides the non-negativity constraints for the variables, the dual has one constraint for each variable in the primal. Let A have entries A_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$. Then A_{ij} is the coefficient of x_j is the i-th constraint and c_j is the coefficient of x_j in the primal objective function. The constraint of the dual corresponding to the j-th variable of the primal is $\sum_i y_i A_{ij} \geq c_j$. The dual has remarkable properties.

Theorem 1 (Weak Duality) *If x and y are feasible solutions of the dual and primal respectively, then*

$$c^T x \leq y^T b.$$

Proof:

$$c^T x \leq y^T A x \leq y^T b$$

where the first inequality holds since $x \geq 0$ and $y^T A \geq c^T$ and the second inequality holds since $y \geq 0$ and $Ax \leq b$. ∎

Theorem 2 (Strong Duality) *The following four possibilities are exhaustive.*

- *The primal and the dual are feasible and bounded. Then they have the same optimal values and there are feasible x^* and y^* with $c^T x^* = (y^*)^T b$.*
- *The primal is feasible and unbounded and the dual is infeasible.*
- *The dual is feasible and unbounded and the primal is infeasible.*
- *Primal and dual are infeasible.*

In particular, if the primal is feasible and bounded then the dual is also feasible and bounded. Moreover, the optimal objective values are the same.

Theorem 3 (Complimentary Slackness) *Assume primal and dual are feasible and x^* and y^* are optimal solutions of the primal and dual, respectively. Then*

$$\begin{aligned} x_j^* > 0 \quad &\Rightarrow \quad \sum_i y_i A_{ij} = c_j \\ y_i^* > 0 \quad &\Rightarrow \quad \sum_j A_{ij} x_j = b_i, \end{aligned}$$

i.e., the constraints corresponding to positive variables are tight.

Proof: By strong duality, $c^T x^* = (y^*)^T b$. Thus both inequalities in the proof of weak duality must be equalities. The claim follows. ∎

Let us have a look at the duals of the programs of Section 3.

Max Flow, Alternative Formulation: We have a variable f_p for every path p from s to t. Its value is the flow along p. Then flow conservation is guaranteed and we only have to enforce the capacity constraint. Thus

$$\begin{aligned} \max \sum_p f_p & \\ \text{subject to} \quad \sum_{p;e\in p} f_e \le cap_e & \qquad \text{for all } e \\ f_p \ge 0 & \qquad \text{for all } p. \end{aligned}$$

In the dual, we have a variable y_e for each edge and a constraint for each path.

$$\begin{aligned} \min \sum_e cap_e y_e & \\ \text{subject to} \quad \sum_{e;e\in p} y_e \ge 1 & \qquad \text{for all } p \\ y_e \ge 0 & \qquad \text{for all } e. \end{aligned}$$

What does the dual say? We have to assign nonnegative values to the y_e such that for every path p the sum of the y_e for the edges in the path sum to at least one (we say the path is cut). We call such a vector y a fractional cut. The goal is to minimize the capacity of the cut, i.e., the sum $\sum_e cap_e y_e$.

Let (f_p^*) and (y_e^*) be optimal solutions. Complementary slackness tells us that only saturated edges (edges e with $\sum_{p;e\in p} f_p^* = cap_e$) can be used in the minimum cut, i.e., have $y_e^* > 0$. Similarly, if $f_p^* > 0$, then $\sum_{e\in p} y_e^* = 1$.

Lemma 1 *The dual always has an integral optimal solution, i.e., a solution with $y_e \in \{0,1\}$. An integral optimal solution of the dual is called a minimum cut.*

Proof: This proof is inspired by the proof of Theorem 3.20 in [CCPS98]. Let (f^*, y^*) be an optimal solution to the primal and dual, respectively. Consider the following shortest path problem. We assign length y_e^* to edge e. For any vertex v let z_v be the length of a shortest path from s to v. Then $z_s = 0$ and $z_t = 1$; the latter follows from complementary slackness. Order the vertices in order of increasing z-values as

$$v_1 = s, v_2, \ldots, v_k = t, v_{k+1}, \ldots v_n.$$

For i, $1 \le i < k$, let $R_i = \{v_1, \ldots, v_i\}$ and let C_i be the set of edges with exactly one endpoint in R_i. Observe that an edge $v_j v_\ell$ belongs to C_i if and only if $j \le i < \ell$. Then

$$\begin{aligned} \sum_{1\le i<k} cap(C_i)(z_{v_{i+1}} - z_{v_i}) &= \sum_{1\le i<k} \sum_{e\in C_i} cap(e)(z_{v_{i+1}} - z_{v_i}) \\ &\le \sum_{e=v_i v_j} cap(e)(z_{v_j} - z_{v_i}) \\ &\le \sum_e cap(e) y_e^*. \end{aligned}$$

Here the first inequality follows the observation above that an edge $v_j v_\ell$ belongs to the cuts C_j to $C_{\ell-1}$ ($\leq$ since we are only summing over the cuts C_1 to C_{k-1}) and the second inequality follows from the fact that the distance from s to v_j is at most the distance from s to v_i plus y^*_e.

We have now established

$$\sum_{1 \leq i < k} cap(C_i)(z_{v_{i+1}} - z_{v_i}) \leq \sum_e cap(e) y^*_e.$$

Since $z_{v_{i+1}} - z_{v_i} \geq 0$ for all i and $\sum_{1 \leq i < k} z_{v_{i+1}} - z_{v_i} = z_{v_k} - z_{v_1} = z_t - z_s = 1$, there must be an i with $cap(C_i) \leq \sum_e cap(e) y^*_e$, i.e., an integral cut whose capacity is most the minimum capacity of a fractional cut. ∎

Corollary 4 (Max-Flow-Min-Cut) *The value of a maximum flow is equal to the capacity of a minimum cut.*

Proof: By strong duality, the value of a maximum flow is equal to the minimum capacity of a fractional cut. By the preceding Lemma, the minimum capacity of a fractional cut is the minimum capacity of a cut. ∎

Shortest Paths: The formulation of a problem as an LP may be counterintuitive at first. The shortest path problem is an example.

We have a directed graph G. Each edge e has a length ℓ_e. We are interested in the shortest distance from s to t. We have a variable d_v for every vertex v. Consider

$$\begin{array}{lll} \max & d_t & \\ \text{subject to} & d_v \leq d_u + \ell_e & \text{for every edge } e = (u,v) \\ & d_s \leq 0 & \end{array}$$

Observe that the edge inequalities can also be written as $d_v - d_u \leq \ell_e$ for every edge $e = (u,v)$. In the dual, we have a variable y_e for each edge and a variable z corresponding to the inequality $d_s \leq 0$, and one constraint for each vertex. The constraints are:

$$\begin{array}{lll} z - \sum_{e;e=(s,\_)} y_e + \sum_{e;e=(\_,s)} y_e \geq 0 & & \\ - \sum_{e;e=(v,\_)} y_e + \sum_{e;e=(\_,v)} y_e \geq 0 & & \text{for } v \neq s,t \\ - \sum_{e;e=(t,\_)} y_e + \sum_{e;e=(\_,t)} y_e \geq 1 & & \end{array}$$

and the objective is $\min \sum_e \ell_e y_e$. Let p be any s-t path. If we set $y_e = 1$ for $e \in p$ and $y_e = 0$ for $e \notin p$, and $z = 1$, we obtain a feasible solution with objective value "length of p". Since by weak

duality, the shortest path distance from s to t is a lower bound on the objective value of the dual, the shortest path from s to t is an optimal solution to the dual.

Further insight: the constraints of the dual are flow constraints. Call y_e the flow across e. We want a net-flow of at least one into t, flow conservation in all vertices different from s to t, and call z the net-flow into s. The goal is to minimize the cost $\sum_e \ell_e y_e$ of the flow.

5 Algorithms

There are numerous algorithms for solving linear programs. Some of them only decide feasibility. We will see in Section 5.7 that optimality can be reduced to feasibility.

5.1 Fourier-Motzkin Elimination

This is the simplest algorithm. It decides feasibility and is extremely simple to state.

Choose an arbitrary variable, say x, and solve all inequalities for x. We obtain inequalities I_u that upper bound x as a linear function of the other variables, I_ℓ that lower bound x as a linear function of the other variables, and inequalities I that do not mention x.

We generate for each pair $x \leq u$ of inequality in I_u and $x \geq \ell$ of inequality in I_ℓ a new inequality $\ell \leq u$ and add it to I. Let I' be the resulting set of inequalities.

Lemma 2 *I' is feasible if and only if the original system $I \cup I_u \cup I_\ell$ is feasible.*

Proof: If the original system is feasible, clearly the derived system is feasible.

If the derived system is feasible consider a feasible solution of I. We show how to choose the value for x such that all inequalities in $I \cup I_u \cup I_\ell$ hold. If I_u and I_ℓ are both empty, set x to an arbitrary value. If I_u is non-empty, set x to the minimum of the right-hand sides of inequalities in I_u. Since I' is feasible, this minimum satisfies all the inequalities in I_ℓ. If I_u is empty, but I_ℓ is not, proceed symmetrically. ∎

If all variables are eliminated, we have a set of inequalities between numbers. Such a system is readily decided.

The disadvantage of Fourier-Motzkin elimination is that the number of inequalities may square whenever a variable is eliminated.

5.2 The Simplex Algorithm

The Simplex algorithm was published by Georg Dantzig in 1947. It is still the most widely used method for solving linear programs. The algorithm is simple.

Recall that the feasible region R is a polytope in $\mathbb{R}^n$. A polytope has vertices and edges connecting two vertices. Geometric intuition tells us that the optimal solution is a vertex of the polytope (there may be others and we have to be a bit careful when the LP is unbounded). Moreover, we can find an optimal solution by walking along the edges of the polytope. This leads to the following algorithm.

find a vertex v of the feasible region
while true **do**
  if v has a neighbor with a higher objective value, let v be some such neighbor
  if v has no such neighbor, stop. v is optimal.
end while

If there is more than one neighbor with a higher objective value, the *pivot rule* determines the neighbor. Popular pivot rules are:

- any neighbor
- a random neighbor
- the neighbor across the steepest edge
- the neighbor with the best objective value
- ...

Some facts about the Simplex algorithm.

- In practice, the algorithms usually works very well and terminates in a number of iterations that grows only linearly with the dimension $\max(n,m)$ of the input.
- For most known pivot rules, there are examples for which the algorithms needs a super-polynomial number of iterations.
- There is a randomized Pivot rule that guarantees a running time of $O(mn^2 + e^{O(\sqrt{n \log n})})$. (Matousek/Sharir/Welzl).
- The smoothed complexity (Spielman and Teng) of the Simplex algorithm is polynomial: here, one perturbs the LP by adding small random noise to each entry of A, b, and c. The smoothed running time is then the expected running time on the perturbations.

5.3 Seidel's Randomized Algorithm

Raimund Seidel (UdS) invented a very simple randomized algorithm. Choose a constraint $a^T x \leq b$ at random and remove it temporarily. Find an optimal solution x^* of the smaller system. If $a^T x^* \leq b$, return x^*. Otherwise, the constraint $a^T x \leq b$ must be satisfied with equality at the optimum of the full system. Solve $a^T x = b$ for one of its variables and substitute into all other inequalities. This reduces the number of variables by one.

5.4 The Ellipsoid Method

The Ellipsoid method was invented by Khachiyan in 1979. It decides feasibility. Let $P = \{x \in \mathbb{R}^n; Ax \leq b \text{ and } x \geq 0\}$ be the feasible region. For simplicity, we assume that P is bounded and has non-empty interior. The algorithm maintains an ellipsoid E that contains all vertices of P.

Let R be large enough so that all vertices of P are contained in the ball of radius R centered at the origin. Initialize E to this ball.

while true **do**

Let z be the center of E. If z is feasible, stop and return z.

If z is not feasible, find an inequality that is violated by z, say $a^T z > b$.

Let E' be the smallest ellipsoid (smallest = minimum volume) containing the "half-ellipsoid" $E \cap \{ x;\, a^T x \leq a^T z \}$.

set E to E'.

If the volume of E is sufficiently small, stop and declare the LP infeasible.

end while

Some explanations are in order. We set E' to $E \cap \{ x;\, a^T x \leq a^T z \}$ instead of $E \cap \{ x;\, a^T x \leq b \}$. This simplifies the calculations. The hyperplane $a^T x = a^T z$ is parallel to the hyperplane $a^T x = b$ (since the normal vector is a in both cases) and passes through z.

We need to have an algorithm that given a point z decides whether z is feasible. If z is infeasible, the algorithm must return a violated inequality. Such an algorithm is called a *separation oracle*.

Clearly, $P \subseteq E$ is an invariant of the algorithm. There are three crucial ingredients to the correctness proof.

- R can be chosen such that $\ln R$ is a polynomial in n and $\log L$. Recall that L is the largest absolute value of any entry in A, b, or c. The volume of the initial ellipsoid E is bounded by R^n. Thus
$$\ln vol(E) \leq n \ln R \leq p_1(n, \log L)$$
for some polynomial p_1.

- The volume of E' is substantially smaller than the volume of E. Substantially smaller means, that the logarithm of the volume decreases by at least $1/p_2(n, \log L)$ for some polynomial p_2.
$$\ln vol(E') \leq \ln vol(E) - \frac{1}{p_2(n, \log L)}.$$

- If P is non-empty then
$$\ln vol(P) \geq -p_3(n, \log L)$$
for some polynomial p_3. This lower bound holds only if P has non-empty interior. So we stop, once $\ln vol(E) < -p_3(n, \log L)$.

  Can it not be the case that the feasible region is non-empty, but has empty interior. Yes, this can be the case. The work-around is to add a small positive quantity ε to all right-hand sides. The quantity must be small enough such that an infeasible problem stays infeasible and such that a feasible problem gets an interior for which the lower bound on the volume applies. It can be shown that such a quantity exists.

Theorem 5 *The number of iterations of the Ellipsoid method is polynomial.*

Proof: The logarithm of the volume of the initial ellipsoid is upper bounded by $p_1(n, \log L)$. In each step, the log of the volume decreases by at least $1/p_2(n, \log L)$. Thus the logarithm of the volume volume of the ellipsoid constructed in the t-th iteration is at most

$$p_1(n, \log L) - t/p_2(n, \log L).$$

If iteration t is not the last, we have

$$p_1(n, \log L) - t/p_2(n, \log L) \geq -p_3(n, \log L).$$

Thus

$$t \leq p_2(n, \log L)(p_1(n, \log L) + p_3(n, \log L)).$$

∎

Observe (!!!) that the upper bound does not depend on m, the number of constraints. As long as we have a separation oracle, the number of constraints is unimportant. It may be exponential in n or even infinite.

5.5 Using the Ellipsoid Method to Put Problems into P

We formulate the traveling salesman problem as an integer linear program.

We have a variable x_e for every edge which we constrain to have values in $\{0, 1\}$. This is achieved by $0 \leq x_e \leq 1$ and $x_e \in \mathbb{N}$. A tour contains two edges incident to every vertex, i.e.,

$$\sum_{e; e \in \delta(v)} x_e = 2 \quad \text{for every } v,$$

where $\delta(v)$ is the set of edges incident to v. Any set of edges which contains exactly two edges incident to every vertex consists of a collection of cycles. We must exclude that a solution consists of a collection of subtours. This is the purpose of the subtour elimination constraints.

$$\sum_{e; e \in \delta(S)} x_e \geq 2 \quad \text{for every } S \subseteq V \text{ with } \emptyset \neq S \neq V,$$

where $\delta(S)$ is the set of edges with exactly one endpoint in S.

The objective function is $\min \sum_e c_e x_e$, where c_e is the cost of e.

The traveling salesman problem is NP-complete. The linear programming relaxation of the TSP is obtained by dropping the integrality constraint $x_e \in \mathbb{N}$.

Lemma 3 *The LP-relaxation of the TSP is in P.*

Proof: We use the Ellipsoid method to solve the LP. The LP has an exponential number of constraints. We need a separation oracle.

Let x_e^*, $e \in E$, be a vector of real values. We perform the following checks:

1. If there is an e, with $x_e^* < 0$ or $x_e^* > 1$, we have found a violated inequality.

2. If there is a vertex v with $\sum_{e;e\in\delta(v)} x_e \neq 2$, we have found a violated inequality.

3. Consider the graph, where we assign capacity $\cap_e = x_e^*$ to edge e. If this graph has a cut of capacity less than two, we have a violated inequality. We fix a vertex s and check for every other vertex t, whether there is a cut of value less than two separating s from t.

Assume we would not know that the min-cut problem (= max-flow problem) is in P. Recall the s-t-cut-LP. We have a variable y_e for each edge and a constraint for each path.

$$\begin{aligned} \min \sum_e cap_e y_e & \\ \text{subject to} \quad \sum_{e;e\in p} y_e \geq 1 & \qquad \text{for all } s\text{-}t \text{ paths } p \\ y_e \geq 0 & \qquad \text{for all } e. \end{aligned}$$

Let y_e^* be a vector of real values. We check $y_e \geq 0$ for every e and then compute a shortest path from s to t with respect to the edge lengths y_e^*. If the distance from s to t is less than one, the shortest path yields a violated inequality.

Finally, the shortest path problem is in P, because it is an LP with a polynomial number of constraints. ∎

5.6 Interior Point Methods

The interior point methods are alternative polynomial time algorithms for solving general LPs. They are inspired by methods used in convex optimization for many years. They are of great practical value. In recent years, interior point methods for specific LPs, e.g., max-flow and min-cost flow, have been determined. These methods are competitive with combinatorial algorithms for the problem. We might see a revolution of combinatorial optimization in the sense that combinatorial methods are replaced by numerical methods.

Consider $\max c^T x$ subject to $Ax \leq b$. The idea is to approximate the polyhedron P defined by $Ax \leq b$ by a hierarchy of convex sets. Let

$$\Phi(x) = \sum_{1\leq i\leq m} \ln(b_i - a_i^T x),$$

where a_i^T is the i-th row of A. The function $-\ln(b_i - a_i^T x)$ goes to minus infinity (in fact, very quickly) as x goes to the hyperplane $b_i = a_i^T x$ inside P. For $x \in \mathbb{R}$, let $L_z = \{x; \Phi(x) \leq z\}$. Then

$$L_z \subseteq P \text{ for all } z \in \mathbb{R} \quad \text{and} \quad \cup_{z\in\mathbb{R}} L_z = P, i.e.,$$

each level set is contained in P and the level sets exhaust P.

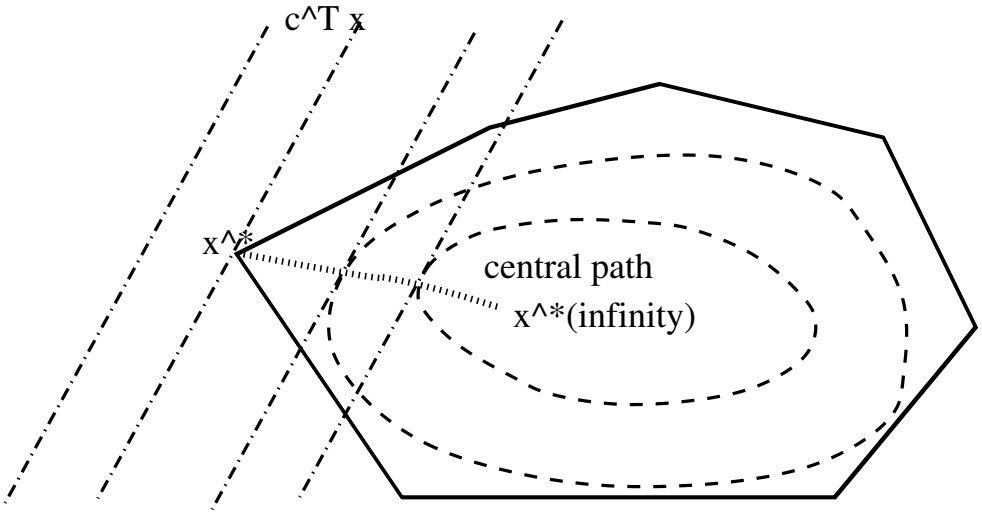


Figure 1: An illustration of the interior point method. The polygon P is indicated by a solid line. Dashed lines indicate level sets of logarithmic barrier function $\Phi(x)$. For fixed μ, $x^*(mu)$ lies on the boundary of some level set; the tangent to the level set at $x^*(\mu)$ is parallel to the level sets of the objective function. The central path is indicated by dots.

We could now proceed as follows. We maximize $c^T x$ subject to $x \in L_z$ for increasing values of z. For each fixed z, this is a convex optimization problem. One can use the method of Lagrange multipliers to solve the problem.

Alternatively, we maximize $c^T x + \mu\Phi(x)$, where $\mu > 0$ is a parameter. Let $x^*(\mu)$ be the unique optimal solution. The solution is unique since $c^T x + \mu\Phi(x)$ is concave. For fixed μ, one can find $x^*(\mu)$ by standard methods, e.g., Newton's method.

The idea is to compute $x^*(\mu)$ for decreasing value of μ. A near optimal solution is easy to find for large μ. As one decreases μ, a near-optimal solution for the previous value of μ is taken as the starting value for the search for the next $x^*(\mu)$. For $\mu \to 0$, the solution $x^*(mu)$ traces a path (called the central path) converging to the optimal solution of the LP.

5.7 Optimality and Feasibility

In this section we show that computing an optimal solution can be reduced to deciding feasibility. For this purpose, we have a closer look at the vertices of the feasible region.

Let A be $m \times n$ and let b be an m-vector. We assume for simplicity that the polytope $P = \{x \in \mathbb{R}^n; Ax \leq b \text{ and } x \geq 0\}$ is full-dimensional. In each vertex of P, n out of the $m+n$ constraints are tight, i.e., satisfied with equality. Each vertex of the feasible region is the solution of a $n \times n$ system of linear equations, say $A'x = b'$, consisting of n tight constraints (tight = satisfiedwith equality). Let L be the largest absolute value of the entry in A, b, and c. The entries are integral. By Cramer's rule each coordinate of a vertex is the quotient of two determinants. The denominator is $\det A'$, i.e., all coordinates have the same denominator. For the numerator of the i-th coordinate the i-th column of A' is replaced by b'. The determinants are determinants of

$n \times n$ matrices with integer entries bounded by L and hence have value in $[1, n!L^n]$. Recall that a determinant is a sum of $n!$ terms each bounded by L^n. Thus the coordinates are rational numbers $\pm p/q$ with $0 \leq p \leq n!L^n \leq (nL)^n$ and $1 \leq q \leq n!L^n \leq (nL)^n$ and therefore the maximal objective value at a vertex is $nn!L^{n+1} \leq (nL)^{n+1}$ and the minimal objective value is the negative of this number.

An LP is unbounded iff the LP $Ax \leq b$, $x \geq 0$, $c^T x \geq 1 + (nL)^{n+1}$ is feasible. Observe that $\log(nL)^{n+1} \leq (n+1)(\log n + \log L)$ is polynomial in n and $\log L$.

Assume now we have a bounded and feasible LP. We can find the maximal objective value by binary search. Let $U = (nL)^{n+1}$ and $L = -U$. Let $M = (L+U)/2$. We test whether $Ax \leq b$, $x \geq 0$, $c^T x \geq M$ is feasible. If so, we set $L = M$. If not, we set $U = M$. When can we stop? We know at all times, that there is a vertex having an objective value of at least L and there is no vertex that has a value of U or more. We can stop if the interval is small enough so that no two distinct values are contained in it.

Consider a vertex. It's i-th coordinate is a rational number of the form p_i/q with $q \leq (nL)^n$; recall that all coordinates have the same denominator. Consider two vertices v and v' and their objective values $c^T v$ and $c^T v'$. Let q be the common denominator of the coordinates of v and q' be the common denominator of the coordinates of the vertices of v'. If the objective values are different, they differ by at least $1/(qq') \geq 1/(nL)^{2n}$.

How long does the binary search take? We start with an interval of length no more than $(nL)^{n+1}$ and end with an interval of length $1/(nL)^{2n}$. Thus the number of iterations is at most

$$\log_2 \frac{(nL)^{n+1}}{(nL)^{2n}} = (3n+1)\log(nL) = poly(n, \log L).$$

We can now also compute a lower bound on the volume of P. Observe that the vertices have denominators q with $q \leq (nL)^n$. Consider the simplex spanned by any $(n+1)$ vertices v_1 to v_{n+1} of P. It has volume $1/(n!)$ times the determinant of the $n \times n$ matrix with columns $v_2 - v_1$, $v_3 - v_1$, ..., $v_{n+1} - v_1$. The entries in column i are rational numbers with denominators $q_{i+1}q_1$, where q_i is the common denominator of the coordinates of the i-th vertex. We can move the denominators out of the determinant and obtain $1/(n!(\prod_{2 \leq i \leq n+1} q_i)q_1^n)$ times the determinant of a matrix with integer entries. The latter is at least one. Thus

$$vol(P) \geq 1/(n!(\prod_{2 \leq i \leq n+1} q_i)q_1^n) \geq \frac{1}{L^{poly(n)}}.$$

6 Integer Linear Programs

A integer linear program (ILP) is a linear program with the additional constraint that the values of the variables must be integral.

Deciding feasibility of ILPs is NP-complete. Every NP-complete problem can be formulated as an ILP. For many NP-complete problems, this formulation is elegant.

The Traveling Salesman Problem: We have a variable x_e for every edge which we constrain to have values in $\{0,1\}$. This is achieved by $0 \leq x_e \leq 1$ and $x_e \in \mathbb{N}$. A tour contains two edges incident to every vertex, i.e.,

$$\sum_{e;e\in\delta(v)} x_e = 2 \quad \text{for every } v,$$

where $\delta(v)$ is the set of edges incident to v. Any set of edges which contains exactly two edges incident to every vertex consists of a collection of cycles. We must exclude that a solution consists of a collection of subtours. This is the purpose of the subtour elimination constraints.

$$\sum_{e;e\in\delta(S)} x_e \geq 2 \quad \text{for every } S \subseteq V \text{ with } \emptyset \neq S \neq V,$$

where $\delta(S)$ is the set of edges with exactly one endpoint in S.

The objective function is $\min \sum_e w_e x_e$, where w_e is the weight of e.

The Vertex Cover Problem: We have a variable x_v for every vertex v which we constrain to have values in $\{0,1\}$. For each edge $e = uv$, we have the constraint $x_u + x_v \geq 1$. The objective is $\min \sum_v x_v$.

The *LP-relaxation* of an ILP is obtained by dropping the integrality constraints on the variables. The optimal solution of an LP-relaxation is, in general, fractional. Nevertheless, LP-relaxations are interesting for several reasons.

6.1 Some LPs are guaranteed to have integral optimal solutions

We give two examples:

Maximum Weight Bipartite Matching: Consider a bipartite graph G. Let n be the number of nodes of G. Each edge e has an associated positive weight w_e. We want to find a matching of maximum weight. A matching is a set of edges no two of which share an endpoint. We introduce a variable x_e for each edge. The goal is to maximize $\sum_e w_e x_e$ subject to $\sum_{e;e\in\delta(v)} \leq 1$ for all v and $x_e \in \{0,1\}$.

In the LP-relaxation, we replace $x_e \in \{0,1\}$ by $x_e \geq 0$. Observe that $x_e \leq 1$ in any optimal solution.

Lemma 4 *The LP-relaxation has an optimal integral solution.*

Proof: We show that the vertices of the feasible region are integral. A vertex of the feasible region is a solution of a $m \times m$ linear system. Each row in this system is either of the form $x_e = 0$ for some e or of the form $\sum_{e;e\in\delta(v)} = 1$ for some node v of the graph. We show that the determinant of such a linear system is in $\{-1,0,+1\}$. The rows corresponding to equations of the form $x_e = 0$ contain a single one and hence we can eliminate them. We are left with a system coming from equations of the form $\sum_{e;e\in\delta(v)} = 1$. Consider any column of the system (columns are indexed by edges), say the column corresponding to $e = uv$. If neither the constraint for u

nor the constraint for v is present, the column contains no non-zero entry and the determinant is zero. If exactly one of the constraints is present, the column contains a single one, and we can delete the column and the row containing the one without changing the value of the determinant. We are left with the case that each column contains exactly two ones. At this point, we use the fact that our graph is bipartite. Let U and V be the two sides of the bipartite graph. Multiply the equations for V with -1 and then add all equations. We get the all-zero vector. Thus the determinant is zero.

Let me give an alternative proof. Let $(x_e)_{e\in E}$ be an optimal solution with a maximum number of integral entries. If all entries are integral, we are done. Otherwise, consider the subgraph G_f (f stands for fractional) formed by the edges with fractional values, i.e., by the edges $\{e; 0 < x_e < 1\}$. Let V' be the vertex set of G_f. The vertices in V' have degree at least two and hence G_f contains a cycle C. C has even length since G is bipartite. Split C into two set of edges C_1 and C_2 by putting edges alternatingly into C_1 and into C_2. Consider the solution

$$x'_e = x_e + \begin{cases} \varepsilon & \text{if } e \in C_1 \\ -\varepsilon & \text{if } e \in C_2 \\ 0 & \text{if } e \notin C. \end{cases}$$

If ε is small enough, the vector $x' = (x'_e)_{e\in E}$ is feasible. It weight is the weight of x plus $\varepsilon(w(C_1) - w(C_2))$. If $w(C_1) \neq w(C_2)$, the solution x is not optimal. If the weight are equal, we can get an optimal solution with an additional integral entry, a contradiction to the choice of x. ∎

Curve Reconstruction: The LP-relaxation of the TSP-ILP given above (the sub-tour LP) is not integral. It may have fractional solutions. Figure 2 shows an example.

However, for the following class of instances, the LP-relaxation always has an integral optimal solution [AM02]. There is an unknown closed curve in the plane and the vertices of the graph correspond to points on this curve. The distances between the vertices are the Euclidean distances between the points. If the points are sufficiently dense sampled, the LP-relaxation has an integral optimal solution and the Traveling Salesmen Tour reconstructs the curve. Figure 3 shows an example.

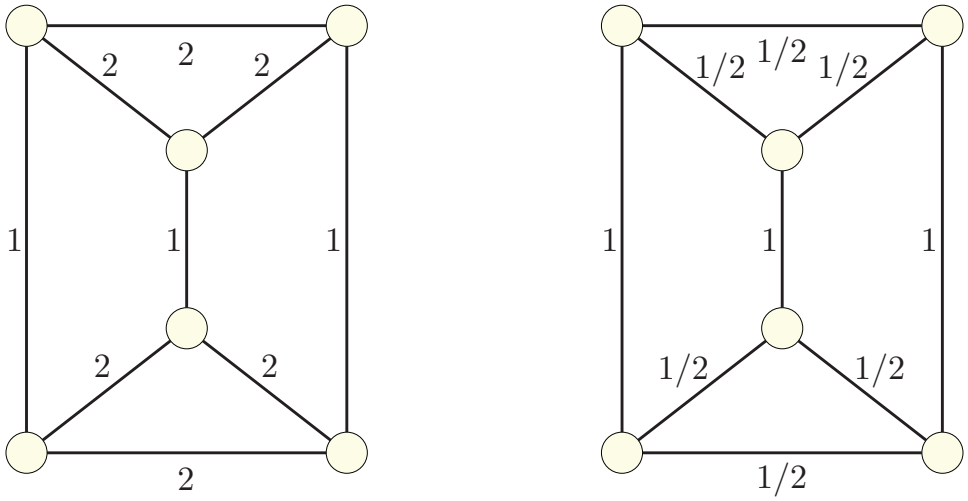
6.2 ILPs can be solved by repeatedly solving LP-Relaxations

How can one solve ILPs? Popular methods are branch-and-bound and branch-and-cut. The principle is simple. I assume for simplicity that all variables are constrained to be either 0 or 1.

Assume we have computed an optimal solution to an LP-relaxation. If the solution is integral, we are done. If the solution is fractional, we have two choices.

Branch: Take one of the fractional variables, say x_i, and consider two subproblems. In the first problem, one sets x_i to zero, and in the second problem, one sets x_i to one.

Fractional Optimal Solution



- left side: edges weights right side: optimal solution to LP
- optimal tour has cost $4 \cdot 2 + 2 \cdot 1 = 10$.
- fractional solution has cost $6 \cdot 0.5 \cdot 2 + 3 \cdot 1 \cdot 1 = 9$.

10

Figure 2: The left side shows an instance of the Traveling Salesman Problem. The right hand side shows an optimal solution to the Sub-tour-LP.

Cut: Try to find an additional inequality that is satisfied by all integral solutions, but is violated by the fractional solution. Add the inequality and resolve. There are systematic procedures for finding additional valid inequalities.

The book [ABCC06] discusses computational approaches to the Traveling Salesman Problem.

6.3 LP-relaxations are the basis for approximation algorithms

The book [Vaz03] discusses the use of linear programming for approximation algorithms.

I give one example to show you the flavor of what can be done. Recall the vertex cover problem. The goal is to find a minimum number of vertices that cover all edges. We have variables x_v, one for each vertex, and the constraints $x_u + x_v \geq 1$ for each edge uv and $x_v \geq 0$ for

Figure 3: The left hand side shows a point set. The right hand side shows the optimum Traveling Salesman tour through this set of points. It is obtained as a solution of the sub-tour LP.

each vertex v. Moreover, $x_v \in \{0,1\}$. The objective is to minimize $\sum_v x_v$. In the LP-relaxation, we replace the integrality constraints on the variables by $x_v \geq 0$ for all v.

Let x_v^*, $v \in V$ be an optimal solution to the LP-relaxation and let $s^* = \sum_v x_v^*$. We refer to s^* as the size of an optimum fractional vertex cover. Clearly s^* is at most the cardinality of an optimum vertex cover. Then $0 \leq x_v^* \leq 1$ for all v. We interpret the x_v^* as probabilities. Consider the following algorithm.

initialize C to the empty set.
for all vertices v **do**
  flip a coin with success probability x_v^* k times. If at least one success, put v into C.
end for

Lemma 5 *Let s^* be the size of an optimum (fractional) vertex cover, and let $k = \lceil 1 + \log_4 m \rceil$, where m is the number of edges. Then with probability at least $1/2$, the algorithm above constructs a vertex cover C with $|C| \leq 4ks^*$.*

Proof: v is added to C an expected number of kx_v^* times. Thus the expected size of C is $k\sum_v x_v^* \leq ks^*$. The probability that the size of C is four times its expectation or more is at most $1/4$. If C is not a vertex cover, there must be an edge $e = uv$ that is not covered. The probability that neither u nor v is put into C is bounded by

$$(1-x_u^*)^k \cdot (1-x_v^*)^k = (1-x_u^*-x_v^*+x_u^*x_v^*)^k \leq \frac{1}{4^k},$$

where the last inequality uses $1-(a+b)+ab \leq 1/4$ for $a+b \geq 1$. The probability that some edge is not covered is at most m times this. By the choice of k, the probability that some edge is not covered is at most $1/4$.

We conclude that the probability that C is either not a node cover or has size more than $4ks^*$ is bounded by $1/2$. ∎

References

[ABCC06] D. Applegate, B. Bixby, V. Chvatal, and W. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

[AM02] Ernst Althaus and Kurt Mehlhorn. Traveling Salesman-Based Curve Reconstruction in Polynomial Time. *SIAM Journal on Computing*, 31(1):27–66, February 2002.

[CCPS98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, Inc, 1998.

[Chv93] V. Chvatal. *Linear Programming*. Freeman, 93.

[Sch03] A. Schrijver. *Combinatorial Optimization (3 Volumes)*. Springer Verlag, 2003.

[Vaz03] V. Vazirani. *Approximation Algorithms*. Springer, 2003.