CMPUT 675: Topics in Algorithms and Combinatorial Optimization (Fall 2009)

Lecture 8,9: Ellipsoid Algorithm, Totally Unimodular Matrices

Lecturer: Mohammad R. Salavatipour **Scriber:** Zhong Li

Date: Sept 29, 2009

1 Ellipsoid Algorithm

Simplex was the first algorithm developed for solving LP's. It is still the most practical algorithm although the running time of it is exponential in worst case. Ellipsoid was the first polytime algorithm for LPs. It was known and used earlier for general convex optimiation problems but it was Khachyan in 1979 who showed that it can solve LP in polynomial time. It is not a practical algorithm as it has a large (although polynomial) time bound but has big consequences in combinatorial optimization and design of algorithms (including approximation). Since then, there have been other polynomial algorithms developed to solve LP's (a.k.a. interior point methods).

The question of sloving LP's is that of determining whether a given LP is feasible or not. It can be proved (an exercise) that the optimization problem can be reduced to a polynomial number of calls to an oracle that would decide feasibility of a given LP. Therefore, it is sufficient to present a polynomial time algorithm for deciding feasibility of an LP:

Given: Given a polyhedron $P = \{x : Cx \leq d\}$ find a feasible solution $x \in \mathbb{R}^n$ if there exists one.

The general idea of the algorithm is as follows. Start with a big ellipsoid E that is guaranteed to contain P. We then check if the centre of this ellipsoid is inside P or not. If yes, we return that as the feasible solution. If not, then there is a constraint $c^T x \leq d_i$ which is voilated. This defines a separating hyperplane such that the centre of the elliposid is to one side and P is to the other side of this hyperplane. We then find a smaller ellipsoid based on the intersection of this half-space (containing P) with the previous ellipsoid. This becomes one iteration of the algorithm. If the starting ellipsoid is E_0 and a_0 is the center of this ellipsoid then:

- Let E_0 be the starting ellipsoid containint P and a_0 be its centre.
- While a_i is not in P do
  - Let $c^T x \leq d_i$ be a voilating constraint.
  - Let E_{k+1} be the minimum volume ellipsoid containing $E_k \cap \{x : c^T x \leq c^T a_k\}$
  - Let $k \leftarrow k + 1$.

Figure 1 shows one iteration of the Ellipsoid algorithm. The way ellipsoids are built is guaranteed their sizes to shrink in volume. Thus if P has positive volume, we eventually find a point in P.

Definition 1.1 *A sphere in $\mathbb{R}^n$ centered at the origin is the set*

$$\{x : x_1^2 + x_2^2 + ... + x_n^2 \leq 1\} \equiv \{x | x^T \cdot x \leq 1\}$$

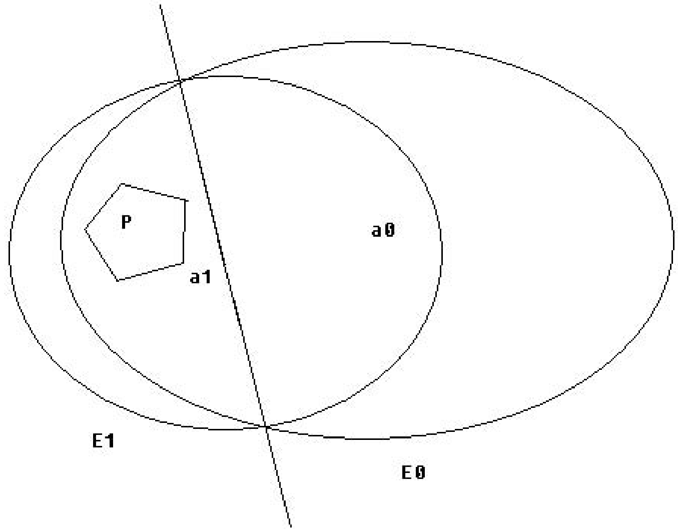


Figure 1: One iteration of Ellipsoid Algorithm

Definition 1.2 *An ellipsoid in $\mathbb{R}^n$ centered at $c \in \mathbb{R}^n$ is obtained by an affine transformation of unit sphere:*

$$\{x \in \mathbb{R}^n | (x-c)^T A^T A(x-c) \leq 1\}$$

For each ellipsoid E, by half ellipsoid $\frac{1}{2}E$ we mean the intersection of E and a halfspace going through the centetr of E, i.e. all x that satisfy the condition $(x-c)^T A^T A(x-c) \leq 1$ as well as $a^T x \geq a^T d$ for some vector d. The following Lemma (whose proof we omit) is the key lemma in bounding the running time of Ellipsoid algorithm.

Lemma 1.3 (Key lemma) *Each half ellipsoid $\frac{1}{2}E$ is an ellpsoid $\tilde{E}$ with:*

$$\frac{Vol(\tilde{E})}{Vol(E)} < e^{-\frac{1}{2(n+1)}}.$$

Therefore, after at most $2(n+1)\ln\frac{Vol(E_0)}{Vol(P)}$ iterations, we should arrive at a point in P. It can be shown that one can find an initial E_0 such that $\frac{Vol(E_0)}{Vol(P)}$ is bounded by a polynomial in terms of length of encoding of P. In typical applications, we have $0 \leq x \leq 1$, so $P = conv(X)$ for some $X \subseteq \{0,1\}^n$. For such applications we can start with E_0 being a ball (sphere) centered at $(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$ and radius $\frac{1}{2}\sqrt{n}$. This ball contains all the points of $\{0,1\}^n$ and has volume bounded by $vol(E_0) = \frac{1}{2^n}(\sqrt{n})^n Vol(B_n)$ where B_n is the unit ball. A crude upper bound for $Vol(B_n)$ is 2^n. Thus $\log(Vol(E_0)) = O(n \log n)$.

Separation Oracle and exponential size LP's:

One important feature of Ellipsoid algorithm is that we do not need to have an explicit representation of the LP for the algorithm. What we need to have is to be able to decide whether a gien point x belongs to P or not, and if not find a violated constraint. This enables us to solve even LP's with exponentially many constraints as long as we can check in polynomial time whether a given proposed solution is feasible or not. In other words, the above analysis actually proves that Ellipsoid algorithm makes a polynomial number of calls to a separation oracle for checking feasibility of a point x.

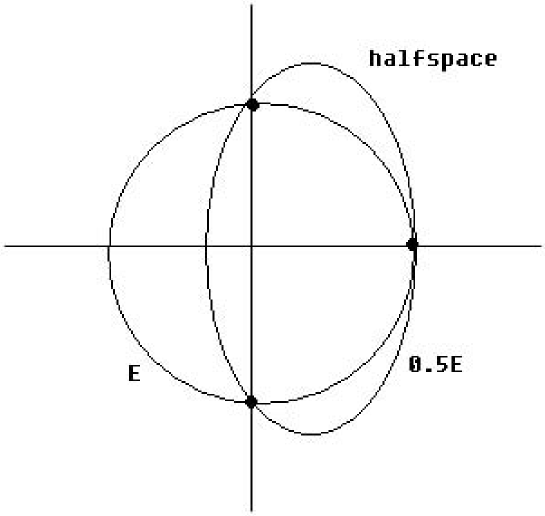


Figure 2: A half ellipsoid obtained from an ellipsoid and a hyperplane going through the center of the ellipsoid

2 Duality

Recall that for every LP there is a dual LP. For a primal LP of the form:

$$\begin{array}{rrcl} \text{minimize} & \sum_{i=1}^{n} c_i x_i & & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j & \geq & b_i \\ & x_j & \geq & 0 \end{array}$$

The dual has the form:

$$\begin{array}{rrcl} \text{maximize} & \sum_{i=1}^{n} b_i y_i & & \\ \text{subject to} & \sum_{i=1}^{m} a_{ij} y_i & \leq & c_j \\ & y_i & \geq & 0 \end{array}$$

By construction, every feasible solution to the dual program gives a lower bound on the optimum value of the primal. Also, every feasible solution to the primal program gives an upper bound on the optimal value of the dual. This is the notion of weak duality which can be proved formally:

Theorem 2.1 (Weak duality) *If $\overline{x}$ and $\overline{y}$ are feasible solutions for primal (min) and dual (max), then*

$$\sum c_i x_i \geq \sum b_j y_j$$

Proof:

$$\begin{aligned} \sum_{j=1}^{n} c_j x_j &\geq \sum_{j=1}^{n} (\sum_{i=1}^{m} a_{ij} y_i) x_j \\ &\geq \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j) y_i \\ &\geq \sum_{i=1}^{m} b_i y_i \end{aligned}$$

■

It turns out that if the optimal solutions of the primal and dual have the same value:

Theorem 2.2 (Strong duality) *If x^* and y^* are optimal feasible solutions of the primal and dual LP then:* $C^T x^* = b^T y^*$.

Theorem 2.3 (Fundermental Theorem of LP) *Any LP given in standard form either:*

(1) has an optimal solution with bounded value, or

(2) is feasible, or

(3) unbounded

This can be proved using the following lemma:

Lemma 2.4 (Ferka's Lemma) *The linear program $Ax = b$ with $x \geq 0$ has no solution if and only if there is y with $A^T y \geq 0$ and $b^T y < 0$.*

Using the above theorem, the relation between feasibility of an LP and its dual can be characterized as below:

$Primal \setminus Dual$	optimal	infeasible	unbounded
optimal	possible	impossible	impossible
infeasible	impossible	possible	possible
unbounded	impossible	possible	impossible

3 Totally Unimodular Matrices (TUM)

Definition 3.1 *A matrix A is TUM if every square submatrix of A has determinent in $\{-1, 0, 1\}$.*

Definition 3.2 *A polyhedron P is called integer if each vertex of P is integer.*

What is the importance of integral polyhedrons? If we can write an optimization problem as an LP and the polyhedron that it defines is integral then solving the LP and finding a vertex optimum solution gives an optimum integer solution of the problem (note that we use the fact that for every LP and every cost function, there is a vertex which is optimum for that function; also for every vertex of a given polytope there is a cost function for which that vertex is the unique optimum solution).

Theorem 3.3 *If A is TUM then for every integral vector b, polyhedron $P = \{x | Ax \leq b, x \geq 0\}$ is integeral.*

Proof: We prove the theorem for $P' = \{x | Ax + Ix = b\}$ since for P we can use slack variables and obtain the polytope P' in standard form. It can be easily shown (exercise) that P is integral if and only if P' is integral. Take any bfs of P' and basis B corresponding to this basic solution. B consists of some columns of A and some of I. Since I has exactly one 1 in each column, if we expand the determinent of A_B along these columns, it becomes (in absolute value) equal to the determinent of some square submatrix of A. Since A is

TUM and because A_B is a collection of linearly independent columns (so its det $\neq 0$), its determinent must be in $\{-1, +1\}$. The basic solution is

$$x = A_B^{-1} \cdot b = \frac{A_b^{adj}}{det(A_B)} \cdot b$$

where A_B^{adj} is the adjoint matrix of A_B and consists of subdeterminents of A_B. Thus both A_B^{adj} and b are integral and $det(B) \in \{-1, +1\}$ which implies that x is integeral. ■

4 Matching and Perfect Matching Polytope

Let $G = (V, E)$ be a weighted graph. The perfect matching polytope $P(G)$ is defined as as the convex hull of incident vectors of perfect matchings of G. More precisely, for each perfect matching M, we define

$$\chi^M \in \mathbb{R}^{|E|} : \chi_e^M = \begin{cases} 1 & e \in M \\ 0 & \text{o.w.} \end{cases}$$

Then $P(G) = conv(\chi^M)$. $P(G)$ is clearly a polyhedron and hence can be defined by a set of linear inequalities. Suppose we consider the following LP to describe this polytope:

$$\begin{aligned} \sum_{e \ni v} x_e &= 1 \quad \forall v \qquad (1) \\ x_e &\geq 0 \end{aligned}$$

Clearly, each point of $P(G)$ satisfies this LP. The question is: does this LP correspond to $P(G)$? The answer is no as these equalities are in general not sufficient since for any odd cycle, e.g. C_3 there is a fractional solution with $x_e = \frac{1}{2}$ everywhere which satisfies this LP but does not belong to $P(G)$. It turns out that for bipartite graphs G, the polytope of this LP is exactly $P(G)$.

Theorem 4.1 *If G is bipartite, any vertex (bfs) of LP(1) corresponds to a perfect matching of G and so is an integer solution.*

Proof: We show any fractional solution is not a vertex, i.e. can be written as convex combination of perfect matchings of G. Take any fractional solution x^*. Without loss of generality, we assume x^* is totally fractional, i.e. $0 < x_e^* < 1$ (we can do so by dropping the edges with $x_e^* = 0$ and also for every edge with $x_e^* = 1$ we consider the project of x^* on the smaller graph obtained by removing the end-points of x_e^*). So every vertex of the graph is incident to at least two edges in x^*. Consider an edge e and one end-point of e, say v. Since $\sum_{e \ni v} = 1$ there is some other edge e' incident to v with $x_{e'}^* > 0$. We can move on e' to the other vertex. We can continue in this manner until we find a cycle. Note that since G is bipartite each cycle has even length. Let C be such a cycle. We can partition the edges of C into two matchings M and H where M contains the odd (indexed) edges and N contains the even (indexed) edges of C (see Figure3).

We can now find two other fractional solutions in the following way. For a sufficiently small $\varepsilon > 0$, in one solution we subtract ε from all edges of M and add ε to all edges of N and we do the opposite in in the other solutions; then the two solutions are $M - \varepsilon \cup N + \varepsilon$ and $M + \varepsilon \cup N - \varepsilon$, note that these are two feasible solutions still and x^* is convex combination of these two. Therefore, x^* is not a bfs, a contradiction. ■

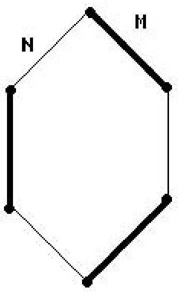


Figure 3: An even cycle C obtained from a fractional solution

This implies that the following LP for weighted perfect matching for bipartite graphs has always an integer solution:

$$\begin{array}{rcll} \max \sum_e w_e x_e & & & \\ \sum_{e \ni v} x_e & = & 1 & \forall v \\ x_e & \geq & 0 & \end{array}$$

We had proved this fact earlier using the primal-dual method that finds both an optimum matching and an optimum vertex cover. The above is yet another proof of the same fact that this LP is integral.

There is a close connection between TUM matrices and the incident matrix of bipartite graphs (as proved in the following theorem). Let A be the $V \times E$ incident matrix of a graph which has a 1 in entry $A[v, e]$ if edge e is incident to vertex v. Note that A has exactly two 1's in each column.

Theorem 4.2 *G is bipartite iff A is TUM.*

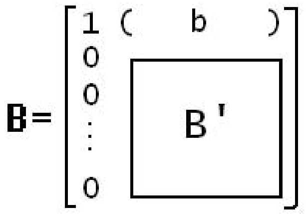
Corollary 4.3 *$P = \{x | Ax \leq 1\}$ is integeral for bipartite graphs G.*

Now we prove the thoerem:

Proof: Suppose A is TUM and G has an odd cycle C, say $|C| = t = 2k + 1$. Let A_c be the submatrix of A corresponding to C; so A_c is a $t \times t$ matrix with exactly two 1's in each row and each column. Since t is odd, it can be easily verified that $det(A_c) = \pm 2$, thus A is not TUM.

Conversely, suppose G is bipartite. We prove that A is TUM. Let B be any $t \times t$ submatrix of A. We prove by indcution that $det(B) \in \{-1, 0, 1\}$. The case of $t = 1$ is trivial. So assume $t > 1$.

- case 1: If B has a column with only 0's then clearly $det(B) = 0$ and we are done.
- case 2: if B had a column with exactly one 1 then it has the following form for some vectore b and submatrix B':



Then using induction on B', $det(B') \in \{-1, 0, 1\}$. Expanding $det(B)$ along this column gives $det(B) \in \{-1, 0, 1\}$.

- case 3: every column has exactly two 1's; since G is bipartite we can write B as $B = \begin{pmatrix} B' \\ B'' \end{pmatrix}$ such that each solumn of B' and each column of B'' has exactly one 1. So sum of rows of B' is vector $(1, 1, \dots, 1)$ and so is the sum of rows of B''. Therefore, rows of B are not linearly independent which implies $det(B) = 0$.

■

Implication of this theorem:
Suppose that $G = (V, E)$ is a bipartite graph and A is its vertex-edge incident matrix. We can derive König's theorem (which says the maximum matching size is equal to the minimum vertex cover) using the above theorem. We know that the maximum matching and minimum vertex cover LP formulations are the following: $P_M = min\{x|Ax \leq 1\}$ and $P_{vc} = max\{y|y^T A \geq 1\}$. By LP duality, the optimum solutions to these LP's are equal. On the other hand, by the previous theorem, A is TUM and therefore polytope P_M is integeral, i.e. each vertex is integer. Thus so is the optimum solution which implies the maximum matching has the same size as minimum vertex cover.

This min-max relation holds for the weighted matching too as long as the weights are integer:

$$\begin{array}{lrcl} \max & \sum w^T.x & & \\ & Ax & \leq & 1 \\ & x & \geq & 0 \end{array}$$

Theorem 4.4 *An integer matrix $A_{m \times n}$ with $a_{ij} \in \{-1, 0, 1\}$ is TUM if no more than two non-zero entries appear in each column and the rows can be partitioned into R_1, R_2 s.t.:*

1. *If a column has two entries of the same sign, their rows are in different parts*
2. *If a column has two entries of different signs, then their rows are in the same part*

Proof: Let B be a $k \times k$ submatrix of A. We prove A is TUM by induction on k. The case of $k = 1$ is trivial.

- If B has a column with all zero entries then clearly $det(B) = 0$.
- If B has a column with exactly one non-zero entry then we can expand $det(B)$ around that column and use induction for the rest of the matrix.
- if every column has two non-zero entry then conditions 1 and 2 above imply that:

$$\sum_{i \in R_1} a_{ij} = \sum_{i \in R_2} a_{ij}$$

  So there is a linear combination of rows which adds to zero, thus B is singular and thus has $det(B) = 0$.

■

Corollary 4.5 *Any LP whose constraint matrix A is either :*

1. *the node-edge incident matrix of an undirected bipartite graph, or*
2. *the node-arc incident matrix of a directed bipartite graph*

has only integeral optimal solutions. This includes LP for shortest path, max-flow, and matching.

Here we give yet another proof of showing integrality of perfect matching polytope of Bipartite graphs.

Theorem 4.6 *Polytope $P_M(G) = \{x | \sum_{e \in v} x_e = 1, x \geq 0\}$ is integeral for bipartite graphs G.*

Proof: Suppose $G = (A \cup B, E)$ is a bipartite graph with $|A| = |B| = n$. Take any bfs x^* and without loss of generality, assume that x^* is totally fractional. Since $\sum_{e \ni v} x_e = 1$, and since all edges are fractional, each node has degree at least 2. Since

$$|E| = \frac{1}{2} \sum d(v) \geq \frac{1}{2} \cdot 2 \cdot 2n = 2n$$

there are at least $2n$ edges with non-zero value. There are $2n$ variables so in bfs there are $2n$ tight constraints, but these tight constraints are not linearly independent because:

$$\sum_{i \in A} x^*_{ij} = \sum_{i \in B} x^*_{ij}.$$

so there are at most $2n - 1$ linearly independent tight constraints, but we have $2n$ variables. Thus a bfs cannot be fractional. ■

5 General Matching Polytope

Recall that our first attempt at describing the perfect matching polytope for general graphs $P_{perfect} = convx(X)$ where X is the set of perfect matchings of G was:

$$\begin{aligned} x(\delta(v)) &= 1 \quad \forall v \\ x_e &\geq 0 \end{aligned} \tag{2}$$

The polytope defined by LP(2) is integral for bipartite graphs (as we proved) but there are fractional solutions for odd cycles that are feasible for this LP and do not belong to $P_{perfect}$. So this LP is not strong enough for the perfect matching polytope of general graphs.

Note that for any odd size set $U \subseteq V$, any perfect matching M can have at most $\frac{|U|-1}{2}$ edges in U, and thus must have at least one edge crossing U. So we can strenghten the above LP by adding constraints of these type:

$$\begin{aligned} x(\delta(v)) &= 1 \quad \forall v \\ x(E(U)) &\leq \frac{|U|-1}{2} \quad \forall \text{ odd set } U \subseteq V \\ x_e &\geq 0 \end{aligned} \tag{3}$$

Note that constraints of the second type are equivalent to saying that $x(\delta(U)) \geq 1$. It is easy to see that such constraints are violated by fractional solutions that assigns $1/2$ to each edge of an odd cycle (and was feasible for LP(2). Next lecture we will prove that this LP is actually describing the polytope of perfect matchings of G by showing that:

Theorem 5.1 *LP(3) is integeral i.e. every vertex corresponds to a perfect matching of G.*

References

1. Combinatorial Optimization, Schrijver, (Volume 1) Springer-Verlag, 2003.

2. Lecture notes by M.X. Goemans for Combinatorial Optimization, 2007 http://www-math.mit.edu/ goemans/18433S07/18433.html

3. Lecture notes by Lap Chi Lau for Combinatorial Optimization and Approximation Algorithms, 2008. http://www.cse.cuhk.edu.hk/ chi/csc5160/index.html