CS 598CSC: Combinatorial Optimization	Lecture date: 01/19/2010
Instructor: Chandra Chekuri	Scribe: Alina Ene

1 Introduction and Motivation

Roughly speaking, an optimization problem has the following outline: given an instance of the problem, find the "best" *solution* among all solutions to the given instance. We will be mostly interested in discrete optimization problems where the instances and the solution set for each instance is from a discrete set. This is in contrast to continuous optimization where the input instance and the solution set for an instance can come from a continuous domain. Some of these distinctions are not as clear-cut as linear programming shows.

We assume familiarity with the computational complexity classes **P** , **NP** , **coNP** . In this class we are mainly interested in polynomial time solvable "combinatorial" optimization problems. Combinatorial optimization problems are a subset of discrete optimization problems although there is no formal definition for them. A typical problem in combinatorial optimization has a ground set E of objects and solutions correspond to some subsets of 2^E (the power set of E) and a typical goal is to find either a maximum or minimum weight solution for some given weights on E. For example, in the spanning tree problem, the ground set E is the set of edges of a graph $G = (V, E)$ and the solutions are subsets of E that correspond to spanning trees in G.

We will be interested in **NP** optimization problems – **NPO** problems for short. Formally, a problem Q is a subset of Σ^*, where Σ is a finite alphabet such as binary. Each string I in Q is an *instance* of Q. For a string x we use $|x|$ to denote its length. We say that Q is an **NPO** problem if the following hold:

(i) for each $x \in \Sigma^*$ there is a polynomial time algorithm that can check if $x \in Q$, i.e., if x is a valid instance of Q

(ii) for each instance I there is a set $sol(I) \subset \Sigma^*$ such that

  (a) $\forall s \in sol(I)$, $|s| = poly(|I|)$

  (b) there exists a poly-time algorithm that on input I, s correctly outputs whether $s \in sol(I)$ or not

(iii) there is a function $val : \Sigma^* \times \Sigma^* \to \mathbb{Z}$ s.t. $val(I, s)$ assigns an integer to each instance I and $s \in sol(I)$ and moreover val can be computed by a poly-time algorithm.

Given an **NPO** problem, we say it is a minimization problem if the goal is to compute, given $I \in Q$, $\arg\min_{s \in sol(I)} val(I, s)$. It is a maximization problem if the goal is to compute $\arg\max_{s \in sol(I)} val(I, s)$. A natural *decision* problem associated with an **NPO** problem (say, maximization) is: given I and integer k, is there $s \in sol(I)$ s.t. $val(s, I) \geq k$.

Many problems we encounter are **NPO** problems. Some of them can be solved in polynomial time. It is widely believed and conjectured that **P** $\neq$ **NP** , which would mean that there are **NPO** problems (in particular, those whose decision versions are **NP** -complete) that do not have polynomial time algorithms. Assuming **P** $\neq$ **NP** , some important and useful questions are: What problems are in **P** ? What characterizes problems in **P** ?

These are not easy questions. One insight over the years is that computation and algorithms are difficult to understand and we are far from being able to characterize the complexity of problems. However, in limited settings we seek to study broad classes of problems and understand some unifying themes. One particular class of problems where this has been possible is the class of constraint satisfaction problems in the Boolean domain. A result of Schaefer completely characterizes which problems are in P and which are **NP** -complete. In fact, there is a nice dichotomy. However, the non-Boolean domain is much more complicated even in this limited setting.

In the field of combinatorial optimization some unified and elegant treatment can be given via polyhedra and the ellipsoid method. The purpose of this course is to expose you to some of these ideas as well as outline some general problems that are known to be solvable in polynomial time. The three ingredients in our study are

(i) polynomial time algorithms

(ii) structural results, especially via min-max characterizations of optimal solutions

(iii) polyhedral combinatorics

We will illustrate these ingredients as we go along with examples and general results. In this introductory lecture, we discuss some known examples to highlight the view point we will take.

2 Network Flow

Given directed graph $D = (V, A)$, two distinct nodes $s, t \in V$, and arc capacities $c : A \rightarrow \mathbb{R}^+$, a flow is a function $f : A \rightarrow \mathbb{R}^+$ s.t.

(i) $\sum_{a\in\delta^-(v)} f(a) = \sum_{a\in\delta^+(v)} f(a)$ for all $v \in V - \{s,t\}$

(ii) $0 \leq f(a) \leq c(a)$ for all $a \in A$

The value of f is

$$val(f) = \sum_{a\in\delta^+(s)} f(a) - \sum_{a\in\delta^-(s)} f(a) = \sum_{a\in\delta^-(t)} f(a) - \sum_{a\in\delta^+(t)} f(a)$$

The optimization problem is to maximize the flow from s to t.
An s-t cut is a partition of V into (A, B) s.t. $s \in A$, $t \in B$, and the capacity of this cut is

$$c(A, B) = \sum_{a\in\delta^+(A)} c(a)$$

Clearly, for any s-t flow f and any s-t cut (A, B)

$$val(f) \leq c(A, B) \Rightarrow \max\ s\text{-}t \text{ flow} \leq \min\ s\text{-}t \text{ cut capacity}$$

The well-known maxflow-mincut theorem of Menger and Ford-Fulkerson states that

Theorem 1 *In any directed graph, the max s-t flow value is equal to the min s-t cut capacity.*

Ford and Fulkerson proved the above theorem algorithmically. Although their augmenting path algorithm is not a polynomial-time algorithm, it can be made to run in polynomial time with very slight modifications (for example, the Edmonds-Karp modification to use the shortest augmenting path). Moreover, the algorithm shows that there is an *integer* valued maximum flow whenever the capacities are integer valued. Thus we have

Theorem 2 *In any directed graph, the max s-t flow value is equal to the min s-t cut capacity. Moreover, if c is integer valued then there is an integer valued maximum flow.*

This is an example of a polynomial time (good) algorithm revealing structural properties of the problem in terms of a min-max result and integrality of flows. Conversely, suppose we knew the maxflow-mincut theorem but did not know any algorithmic result. Could we gain some insight? We claim that the answer is yes. Consider the decision problem: given G, s, t, is the s-t max flow value at least some given number k? It is easy to see that this problem is in **NP** since one can give a feasible flow f of value at least k as an **NP** certificate[1]. However, using the maxflow-mincut theorem we can also see that it is in **coNP** . To show that the flow value is smaller than k, all we need to exhibit is a cut of capacity smaller than k. Therefore the min-max result shows that the problem is in **NP** $\cap$ **coNP** . Most "natural" decision problems in **NP** $\cap$ **coNP** have eventually been shown to have polynomial time algorithms (there are a few well-known exceptions). Moreover, a problem in **NP** $\cap$ **coNP** being **NP** -complete or **coNP** -complete would imply that **NP** $=$ **coNP** , some thing that most believe is unlikely. Thus a min-max result implies that the decision version is in **NP** $\cap$ **coNP** , strong evidence for the existence of a poly-time algorithm. That does not imply that such an algorithm will come by easily. Examples include matchings in general graphs and linear programming.

Finally, let us consider network flow as a special case of a linear programming problem. We can write it as

$$
\begin{array}{rcll}
\max & & \displaystyle\sum_{a\in\delta^+(s)} f(a) - \sum_{a\in\delta^-(s)} f(a) & \\
s.t. & & & \\
\displaystyle\sum_{a\in\delta^+(v)} f(a) & - & \displaystyle\sum_{a\in\delta^-(v)} f(a) = 0 & \forall v \in V, v \neq s,t \\
f(a) & \leq & c(a) \qquad \forall a \in A & \\
f(a) & \geq & 0 \qquad \forall a \in A &
\end{array}
$$

From the polynomial time solvability of LP, one can conclude a poly-time algorithm for network flow. However, one can say much more. It is known that the matrix defining the above system of inequalities is *totally unimodular* (TUM). From this, we have that the vertices of the polytope are integral whenever the capacities are integral! Not only that, the dual also has integer vertices since the objective function has integer coefficients. In fact, one can derive the maxflow-mincut theorem from these facts about the polyhedra in question. In addition, one can derive more quite easily. For example, we could add lower bounds on the flow

$$\ell(a) \leq f(a) \leq c(a)$$

[1]We are ignoring a technicality here that the flow specification be of polynomial size in the input.

and one still obtains the fact that if there exists a flow that respects the lower/upper bounds and ℓ, c are integer valued, then there is a corresponding integer valued flow. Same for min-cost flow and several other variants such as circulations and transshipments.

3 Bipartite Matchings

Many of you have seen bipartite matchings reduced to flow. We can also treat them independently. Let $G = (X \cup Y, E)$ be a bipartite graph with bipartition given by X, Y. Recall that $M \subseteq E$ is a *matching* in a graph if no vertex in G is incident to more than one edge in M.

A vertex cover in $G = (V, E)$ is a subset of vertices $S \subseteq V$ such that for each edge $uv \in E$, u or v is in S. In other words, S covers all edges. We use $\nu(G)$ to indicate the cardinality of a maximum matching in G and $\tau(G)$ for the cardinality of a minimum vertex cover in G.

Proposition 3 *For every G, $\nu(G) \leq \tau(G)$.*

In bipartite graphs, one has the following theorem.

Theorem 4 *(König's Theorem) If G is bipartite then $\nu(G) = \tau(G)$.*

The above proves that the max matching and min vertex problems (decision versions) in bipartite graphs are both in **NP** $\cap$ **coNP** . (Note that the vertex cover problem is **NP** -hard in general graphs.) We therefore expect a polynomial time algorithm for $\nu(G)$ and $\tau(G)$ in bipartite graphs. As you know, one can reduce matching in bipartite graphs to maxflow and König's theorem follows from the maxflow-mincut theorem. One can also obtain a polynomial time augmenting path algorithm for matching in bipartite graphs (implicitly a maxflow algorithm) that proves König's theorem algorithmically.

We now look at the polyhedral aspect. We can write a simple LP as a relaxation for the maximum matching in a graph G.

$$\begin{array}{rcll} \max & & \sum_{e \in E} x(e) & \\ \sum_{e \in \delta(u)} x(e) & \leq & 1 & \forall u \in V \\ x(e) & \geq & 0 & \forall e \in E \end{array}$$

For bipartite graphs the above LP and its dual have integral solutions since the constraint matrix is TUM. One can easily derive König's theorem and a polynomial time algorithm from this. One also obtains a polynomial time algorithm for weighted matching (assignment problem).

4 General Graph Matchings

The constraint matrix of the basic LP for matchings given above is not integral for general graphs, as the following simple graph shows. Let $G = K_3$ be the complete graph on 3 vertices. The solution $x(e) = 1/2$ for each of the 3 edges in G is an optimum solution to the LP of value $3/2$ while the maximum matching in G has size 1.

The algorithmic study of general graph matchings and the polyhedral theory that was developed by Jack Edmonds in the 1960's, and his many foundational results are the start of the field of polyhedral combinatorics. Prior to the work of Edmonds, there was a min-max result for $\nu(G)$ due to Berge which is based on Tutte's necessary and sufficient condition for the existence of a perfect matching. To explain this, for a set $U \subseteq V$, let $o(G-U)$ be the number of odd cardinality components in the graph obtained from G by removing the vertices in U.

Tutte-Berge formula

$$\nu(G) = \min_{U \subseteq V} \frac{1}{2}(|V| + |U| - o(G-U))$$

We will prove the easy direction for now, i.e.,

$$\nu(G) \leq \frac{1}{2}(|V| + |U| - o(G-U)) \qquad \forall U \subseteq V$$

To see this, the number of unmatched vertices is at least $o(G-U) - |U|$, since each odd component in $G-U$ needs a vertex in U. Hence

$$\nu(G) \leq \frac{|V|}{2} - \frac{o(G-U) - |U|}{2} \leq \frac{1}{2}(|V| + |U| - o(G-U))$$

Corollary 5 *(Tutte's 1-factor theorem) G has a perfect matching iff* $\forall U \subseteq V \;\; o(G-U) \leq |U|$.

The formula shows that the decision version of matching is in **NP** $\cap$ **coNP** . Edmonds gave the first polynomial time algorithms for finding a maximum cardinality matching and also more difficult maximum weight matching problem. As a consequence of his algorithmic work, Edmonds showed the following results on matching polytopes.

Consider the following polytope:

$$\begin{array}{rcll} \sum_{e \in \delta(v)} x(e) & \leq & 1 & \forall v \in V \\ \sum_{e \in E[U]} x(e) & \leq & \lfloor \frac{|U|}{2} \rfloor & \forall U, |U| \text{ odd} \\ 0 & \leq & x(e) & \forall e \in E \end{array}$$

Edmonds showed that the vertices of the above polytope are exactly the characteristic vectors of the matchings of G. Observe that the polytope has an exponential number of constraints. One can ask whether this description of the matching polytope useful. Clearly if one takes the convex hull of the matchings of G, one obtains a polytope in $\mathbb{R}^E$; one can do this for any combinatorial optimization problem and in general one gets an exponential number of inequalities. Edmonds argued that the matching polytope is different since

(i) the inequalities are described implicitly in an uniform way

(ii) his algorithms gave a way to optimize over this polytope

At that time, no poly-time algorithm was known for solving LPs, although LP was known to be in **NP** $\cap$**coNP** . In 1978, Khachiyan used the ellipsoid algorithm to show that linear programming is in **P** . Very soon, Padberg-Rao, Karp-Papadimitriou, and Grötschel-Lövasz-Schrijver independently realized that the ellipsoid algorithm has some very important features, and can be used to show the polynomial-time equivalence of optimization and separation for polyhedra.

Separation Problem for Polyhedron Q
Given n (the dimension of Q), and an upper bound L on the size of the numbers defining the inequalities of Q, and a rational vector $x_0 \in \mathbb{R}^n$, output correctly either that Q $x \in Q$ or a hyperplane $ax = b$ s.t. $ax \leq b$ $\forall x \in Q$ and $ax_0 > b$.

Optimization Problem for Polyhedron Q
Given n (the dimension of Q), and an upper bound L on the size of the numbers defining the inequalities of Q, and a vector $c \in \mathbb{R}^n$, output correctly one of the following: (i) Q is empty (ii) $\max_{x \in Q} cx$ has no finite solution (iii) a vector x^* s.t. $cx^* = \max_{x \in Q} cx$.

Theorem 6 *(Grötschel-Lövasz-Schrijver) There is a polynomial time algorithm for the separation problem over Q iff there is a polynomial time algorithm for the optimization problem over Q.*

The above consequence of the ellipsoid method had/has a substantial theoretical impact on combinatorial optimization. In effect, it shows that an algorithm for a combinatorial optimization problem implies an understanding of the polytope associated with the underlying problem and vice-versa. For example, the weighted matching algorithm of Edmonds implies that one can separate over the matching polytope. Interesting, it took until 1982 for Padberg and Rao to find an *explicit* separation algorithm for the matching polytope although one is implied by the above theorem.

References

[1] J. Edmonds Paths, trees, and flowers. *Canadian J. of Mathematics*, 17, 449–467, 1965.

[2] J. Edmonds Maximum matching and a polyhedron with 0,1 vertices. *J. of Research National Bureau of Standards Section B*, 69, 73–77, 1965.

[3] M. Grötschel, L. Lövasz and A. Schrijver. *Goemetric Algorithms and Combinatorial Optimization.* Springer, 1988.

[4] L.G. Khachiyan. Polynomial algorithms in linear programming. (English translation of original Russian article). *Doklady Akademii Nauk SSSR* 244, 1093–1096, 1979.

[5] A. Schrijver. *Theory of Linear and Integer Programming (Paperback).* Wiley, 1998.

[6] A. Schrijver. *Combinatorial Optimization: Efficiency and Polyhedra.* Springer, 2003.

CS 598CSC: Combinatorial Optimization	Lecture date: January 21, 2009
Instructor: Chandra Chekuri	Scribe: Sungjin Im

1 Polyhedra and Linear Programming

In this lecture, we will cover some basic material on the structure of polyhedra and linear programming. There is too abundant material on this topic to be covered in a few classes, so pointers will be given for further reading. For algorithmic and computational purposes one needs to work with rational polyhedra. Many basic results, however, are valid for both real and rational polyhedra. Therefore, to expedite our exploration, we will not make a distinction unless necessary.

1.1 Basics

Definition 1. *Let $x_1, x_2, ..., x_m$ be points in $\mathbb{R}^n$. Let $x = \sum_{i=1}^m \lambda_i x_i$, where each $\lambda_i \in \mathbb{R}$, $1 \le i \le m$ is a scalar. Then, x is said to be a(n)*

1. *Linear combination (of x_i, $1 \le i \le m$) for arbitrary scalars λ_i.*
2. Affine *combination if $\sum_i \lambda_i = 1$.*
3. Conical *combination if $\lambda_i \ge 0$.*
4. *Convex combination if $\sum \lambda_i = 1$ and $\lambda_i \ge 0$ (affine and also canonical).*

In the following definitions and propositions, unless otherwise stated, it will be assumed that $x_1, x_2, ..., x_m$ are points in $\mathbb{R}^n$ and $\lambda_1, \lambda_2, ..., \lambda_m$ are scalars in $\mathbb{R}$.

Definition 2. *$x_1, x_2, ..., x_m$ are said to be linearly independent if $\sum_{i=1}^m \lambda_i x_i = 0 \Rightarrow \forall i \in [m]\ \lambda_i = 0$.*

Definition 3. *$x_1, x_2, ..., x_m$ are said to be affinely independent if the vectors $(x_i - x_1), i = 2, ..., m$ are linearly independent, or equivalently if $\sum_{i=1}^m \lambda_i x_i = 0$ and $\sum_{i=1}^m \lambda_i = 0 \Rightarrow \forall i \in [m]\ \lambda_i = 0$.*

The following proposition is easy to check and the proof is left as an exercise to the reader.

Proposition 4. *$x_1, x_2, ..., x_m$ are affinely independent if and only if the vectors $\binom{x_i}{1}$, $i = 1, 2, ..., m$, are linearly independent in $\mathbb{R}^{m+1}$.*

A set $X \subseteq \mathbb{R}^n$ is said to be a(n) subspace [affine set, cone set, convex set] if it is closed under linear [affine, conical, convex] combinations. Note that an affine set is a translation of a subspace. Given $X \subseteq \mathbb{R}^n$, we let $\texttt{Span}(X), \texttt{Aff}(X), \texttt{Cone}(X)$, and $\texttt{Convex}(X)$ denote the closures of X under linear, affine, conical, and convex combinations, respectively. To get an intuitive feel of the above definitions, see Figure 1.

Definition 5. *Given a convex set $X \subseteq \mathbb{R}^n$, the affine dimension of X is the maximum number of affinely independent points in X.*

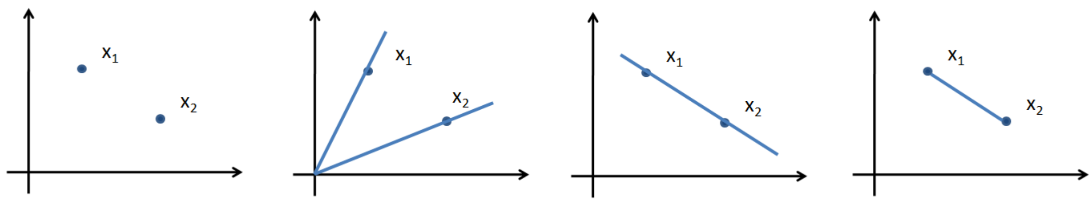


Figure 1: The subspace, cone set, affine set, and convex set of x_1, x_2 (from left to right). Note that the subspace is $\mathbb{R}^2$ and the cone set includes all points inside and on the two arrows.

1.2 Polyhedra, Polytopes, and Cones

Definition 6 (Hyperplane, Halfspace)**.** *A* hyperplane *in $\mathbb{R}^n$ is the set of all points $x \in \mathbb{R}^n$ that satisfy $a \cdot x = b$ for some $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. A* halfspace *is the set of all points x such that $a \cdot x \leq b$ for some $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.*

Definition 7 (Polyhedron)**.** *A* Polyhedron *in $\mathbb{R}^n$ is the intersection of finitely many halfspaces. It can be equivalently defined to be the set $\{x \mid Ax \leq b\}$ for a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^{m \times 1}$.*

Definition 8 (Polyhedral cone)**.** *A* polyhedral cone *is $\mathbb{R}^n$ the intersection of finitely many halfspaces that contain the origin, i.e. $\{x \mid Ax \leq 0\}$ for a matrix $A \in \mathbb{R}^{m \times n}$.*

Definition 9 (Polyotpe)**.** *A* polytope *is a* bounded *polyhedron.*

Note that a polyhedron is a convex and closed set. It would be illuminating to classify a polyhedron into the following four categories depending on how it looks.

1. Empty set (when the system $Ax \leq b$ is infeasible.)
2. Polytope (when the polyhedron is bounded.)
3. Cone
4. (Combination of) Cone and Polytope

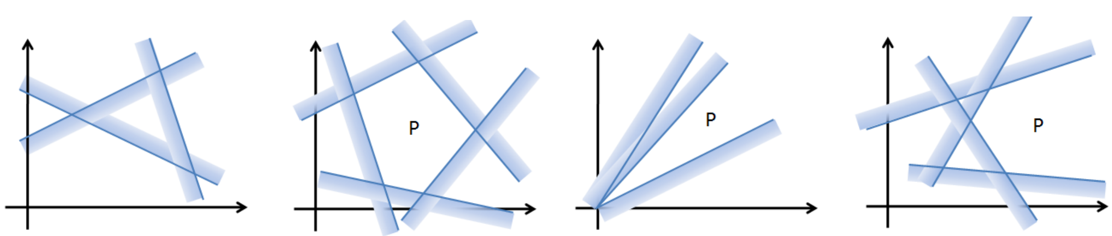
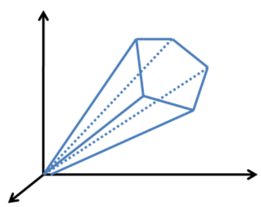


Figure 2: Examples of polyhedrons, left to right: Empty, Polytope, Cone, Combination of cone and polytope.

What "combination of cone and polytope" means will become clear soon in Theorem 12. For the examples, the reader is referred to Figure 2. In 2-D, a cone can have only two "extreme rays," while in 3-D there is no bound on the number of extreme rays it can have.



For the most of part, we will be largely concerned with polytopes, but we need to have a better understanding of polyhedra first. Although it is geometrically "obvious" that a polytope is the convex hull of its "vertices," the proof is quite non-trivial. We will state the following three theorems without proof.

Theorem 10. *A bounded polyhedron is the convex hull of a finite set of points.*

Theorem 11. *A polyhedral cone is generated by a finite set of vectors. That is, for any $A \in \mathbb{R}^{m \times n}$, there exists a finite set X such that $\{x = \sum_i \lambda_i x_i \mid x_i \in X, \lambda_i \geq 0\} = \{x \mid Ax \leq 0\}$.*

Theorem 12. *A polyhedron $\{x \mid Ax \leq b\}$ can be written as the Minkowski sum of a polytope Q and a cone C, i.e. $P = Q + C = \{x + y \mid x \in Q, y \in C\}$.*

One can (geometrically) obtain the Minkowski sum of a polytope Q and a cone C by sweeping the origin of the cone C over the polytope Q. If the polyhedron P is pointed (has at least one "vertex"), the decomposition is, in fact, modulo scaling factor unique. Further the cone C above is $\{x \mid Ax \leq 0\}$, or equivalently the set of unbounded directions in P. The cone C is called the characteristic cone or the recession cone of P.

Many facts about polyhedra and linear programming rely on (in addition to convexity) variants of Farkas' lemma that characterizes when a system of linear inequalities do not have solution. The simplest proof for one variant is via Fourier-Motzkin elimination that is independently interesting and related to the standard Gauss-Jordan elimination for solving system of linear equations.

1.2.1 Fourier-Motzkin Elimination

Let $P = \{x | Ax \leq b\} \subseteq \mathbb{R}^n$ be a polyhedron. For k in $[n]$,
we let $P^k = \{(x_1, .., x_{k-1}, x_{k+1}, ..., x_n) \mid (x_1, x_2, ..., x_n) \in P\}$ be the *projection* of P along the x_k-axis.

Theorem 13. *P^k is a polyhedron.*

Proof. We derive a set of inequalities that describe P^k. We do this by considering the inequalities in $Ax \leq b$ and eliminating the variables x_k as follows. Partition the inequalities in $Ax \leq b$ into three sets:

$$S^+ = \{i \in [m] \mid a_{ik} > 0\}, S^- = \{i \in [m] \mid a_{ik} < 0\}, \text{and } S^0 = \{i \in [m] \mid a_{ik} = 0\}.$$

Define a new set of inequalities consisting of S_0 and one new inequality for each pair (i, ℓ) in $S^+ \times S^-$:

$$a_{ik}(\sum_{j=1}^{n} a_{\ell j} x_j) - a_{\ell k}(\sum_{j=1}^{n} a_{ij} x_j) \leq a_{ik} b_\ell - a_{\ell k} b_i.$$

Note that the combined inequality does not have x_k. We now have a total of $|S^0| + |S^+||S^-|$ new inequalities. Let $P' = \{x' \in \mathbb{R}^{n-1} \mid A'x' \leq b'\}$ where $A'x' \leq b'$ is the new system of inequalities in variables $x_1, x_2, ..., x_{k-1}, x_{k+1}, ..., x_n$. We prove the theorem by showing that $P^k = P'$.

We first show the easier direction: $P^k \subseteq P'$. Consider any point $z \in P^k$. By definition of P^k, there exists $x \in P$ such that $Ax \leq b$ and x's projection along x_k-axis is z. It is easy to see that z satisfies the new system since the new one was obtained in a way oblivious to x_k, the real value of x's k_{th} coordinate.

We now show that $P' \subseteq P^k$. Without loss of generality, assume $k = 1$. Consider any $x' = (x_2, x_3, ..., x_n) \in P'$. We want to show that there exists $x_1 \in \mathbb{R}$ such that $Ax \leq b$, where $x = (x_1, x_2, ..., x_n)$. For simple notation, define $C_i = b_i - \sum_{j=2}^{n} a_{ij}x_j$ for $i \in [m]$. Note that $Ax \leq b$ can be rewritten as

$$a_{i1}x_1 \leq C_i, \forall i \in [m]. \tag{1}$$

Observe that x satisfies all inequalities consisting of S^0, since the new system as well includes those constraints. Thus we can refine our goal to show

$$\begin{array}{rl} \exists x_1 \text{ s.t.} & a_{i1}x_1 \leq C_i, \forall i \in S^+ \cup S^-. \\ \Leftrightarrow & \max_{\ell \in S^-} \frac{C_\ell}{a_{\ell 1}} \leq x_1 \leq \min_{i \in S^+} \frac{C_i}{a_{i1}}. \end{array}$$

It is easy to observe that this is equivalent to

$$\begin{array}{rrcll} & \frac{C_\ell}{a_{\ell 1}} & \leq & \frac{C_i}{a_{i1}} & , \forall (i, \ell) \in S^+ \times S^- \\ \Leftrightarrow & 0 & \leq & a_{i1}C_\ell - a_{\ell 1}C_i & , \forall (i, \ell) \in S^+ \times S^- \\ \Leftrightarrow & A'x' & \leq & b' & \end{array}$$

And we know that $A'x' \leq b'$ since $x' \in P'$, completing the proof. $\square$

From Fourier-Motzkin elimination we get an easy proof of one variant of Farkas' lemma.

Theorem 14 (Theorem of Alternatives). *Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. For the system $Ax \leq b$, exactly one of the following two alternatives hold:*

- *The system is feasible.*
- *There exists $y \in \mathbb{R}^m$ such that $y \geq 0$, $yA = 0$ and $yb < 0$.*

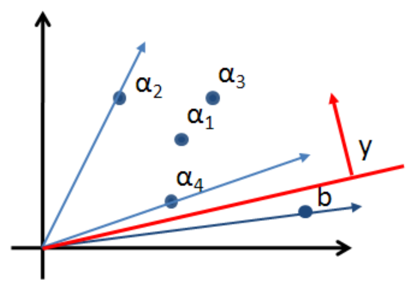
What the theorem says is that if the system of inequalities $Ax \leq b$ is infeasible then there is a proof (certificate) of this which can be obtained by taking non-linear combination of the inequalities (given by $y \geq 0$) to derive a contradiction of the following form: $0 = yA \leq yb < 0$.

Proof of [Theorem 14] Suppose that there exists a vector $y' \geq 0$ s.t. $y'A = 0$ and $y' \cdot b < 0$ and a vector x' such that $Ax' \leq b$. Then it easily follows that $0 \leq y'Ax' \leq y'b$, since $y' \geq 0$, which is a contradiction to the fact that $y'b < 0$.

Conversely, suppose $Ax \leq b$ is infeasible. Let $P = \{x \mid Ax \leq b\}$. We eliminate variables $x_1, x_2, ..., x_n$ (we can choose any arbitrary order) to obtain polyhedra $P = Q_0, Q_1, Q_2, ..., Q_{n-1}, Q_n$. Note that Q_{i+1} is non-empty iff Q_i is, and that Q_{n-1} has only one variable and Q_n has none. Note by the Fourier-Motzkin elimination procedure the inequalities describing Q_i are non-negative combination of the inequalities of P; this can be formally shown via induction. Thus, Q_n is empty iff we have derived an inequality of the form $0 \leq C$ for some $C < 0$ at some point in the process. That inequality gives the desired $y \geq 0$. $\square$

Two variant of Farkas' lemma that are useful can be derived from the theorem of alternatives.

Theorem 15. *$Ax = b$, $x \geq 0$ has no solution iff $\exists y \in \mathbb{R}^m$ s.t. $A^T y \geq 0$ and $b^T y < 0$.*



The above theorem has a nice geometric interpretation. Let $\alpha_1, \alpha_2, ..., \alpha_n$ be the columns of A viewed as vectors in $\mathbb{R}^m$. Then $Ax = b, x \geq 0$ has a solution if and only if b is in the cone generated by $\alpha_1, \alpha_2, ..., \alpha_n$; here the combination is given by $x \geq 0$. So b is either in the $\texttt{Cone}(\alpha_1, \alpha_2, ..., \alpha_n)$ or there is a hyperplane separating b from $\alpha_1, \alpha_2, ..., \alpha_n$.

In fact the theorem can be strengthened to show that the hyperplane can be chosen to be one that spans $t-1$ linearly independent vectors in $\alpha_1, \alpha_2, ..., \alpha_n$, where $t = \texttt{rank}(\alpha_1, \alpha_2, ..., \alpha_n, b)$.

Proof of [Theorem 15] We can rewrites $Ax = b, x \geq 0$ as

$$\begin{bmatrix} A \\ -A \\ -I \end{bmatrix} x \leq \begin{bmatrix} b \\ -b \\ 0 \end{bmatrix}$$

Hence by the Theorem of Alternatives, $Ax = b, x \geq 0$ is not feasible if and only if there exists a row vector $y' = \begin{bmatrix} u & v & w \end{bmatrix}$, where u, v are row vectors in $\mathbb{R}^m$ and w is a row vector in $\mathbb{R}^n$ such that

$$\begin{aligned} u, v, w &\geq 0 \\ uA - vA - w &= 0 \\ ub - vb &< 0 \end{aligned}$$

Let $y = u - v$. Note that $y \in \mathbb{R}^m$ is now not necessarily positive. From the second and third inequalities, we can easily obtain $A^T y \geq 0$ and $b^T y < 0$. □

Another variant of Farkas' lemma is as follows and the proof is left as an exercise.

Theorem 16. *$Ax \leq b, x \geq 0$ has a solution iff $yb \geq 0$ for each row vector $y \geq 0$ with $yA \geq 0$.*

Another interesting and useful theorem is Carathéodory's Theorem

Theorem 17 (Carathéodory)**.** *Let $x \in \texttt{Convexhull}(X)$ for a finite set X of points in $\mathbb{R}^n$. Then $x \in \texttt{Convexhull}(X')$ for some $X' \subseteq X$ such that vectors in X' are affinely independent. In particular, $|X'| \leq n+1$.*

A conic variant of Carathéodory's Theorem is as follows.

Theorem 18. *Let $x \in \texttt{Cone}(X)$ where $X = \{x_1, x_2, ..., x_m\}$, $x_i \in \mathbb{R}^n$. Then $x \in \texttt{Cone}(X')$ for some $X' \subseteq X$ where vectors in X' are linearly independent. In particular, $|X'| \leq n$.*

Proof. Since $x \in \texttt{Cone}(X)$, $x = \sum_i \lambda_i x_i$ for some $\lambda_i \geq 0$. Choose a combination with mininum support, i.e. the smallest number of non-zero λ_i values. Let $X' = \{\lambda_i x_i \mid \lambda_i > 0\}$ and $I = \{i \mid \lambda_i > 0\}$. If vectors in X' are linearly independent, we are done. Otherwise, $\exists \alpha_i, i \in I$ s.t. $\sum_{i \in I} \alpha_i \lambda_i x_i = 0$. By scaling we can assume that $\forall i \in I, \alpha_i \leq 1$, and $\exists j \in I$ s.t. $\alpha_j = 1$. Then,

$$x = \sum_{i \in I} \lambda_i x_i = \sum_{i \in I} \lambda_i x_i - \sum_{i \in I} \alpha_i \lambda_i x_i = \sum_{i \in I} \lambda_i (1 - \alpha_i) x_i = \sum_{i \in I \setminus \{j\}} \lambda'_i x_i \quad (\lambda'_i \geq 0).$$

This contradicts the fact that we chose the conical combination for x with the least support. $\square$

One can derive the affine version of Carathéodory's Theorem from the conical version, and the proof is left as an exercise.

1.3 Linear Programming

Linear programming is an optimization problem of the following form.

$$\begin{aligned} \max \quad & c \cdot x \quad \text{(Primal-LP)} \\ Ax & \leq b \end{aligned}$$

The above is one of the standard forms. In other words, we wish to maximize a linear objective function over a *polyhedron*. Given an LP, there are three possibilities:

1. The polyhedron is *infeasible*.
2. The objective function can be made arbitrarily large in which case we can say it is *unbounded*.
3. There is a finite optimum value in which case we say it is *bounded*.

Each linear program has its associated "dual" linear program. The LP we refer to by "dual" depends on the "starting" LP, which is called as the primal LP; in fact the dual of dual LP is exactly the same as the primal LP. Let us say that the following LP is the primal LP here.

$$\begin{aligned} \max \quad & c \cdot x \\ Ax & \leq b \end{aligned}$$

We can "derive" the dual by thinking about how we can obtain an upper bound on the optimal value for the primal LP. Given the system $Ax \leq b$, any inequality obtained by non-negative combination of the inequalities in $Ax \leq b$ is a valid inequality for the system. We can represent a non-negative combination by a $m \times 1$ row vector $y \geq 0$.

Thus $yAx \leq yb$ is a valid inequality for $y \geq 0$. Take any vector $y' \geq 0$ s.t. $y'A = c$. Then such a vector gives us an upperbound on the LP value since $y'Ax = cx \leq y'b$ is a valid inequality. Therefore one can obtain an upperbound by minimizing over all $y' \geq 0$ s.t. $y'A = c$. Therefore the objective function of the primal LP is upperbounded by the optimum value of

$$\begin{aligned} \min \quad & yb \quad \text{(Dual-LP)} \\ yA & = c \\ y & \geq 0 \end{aligned}$$

The above derivation of the Dual LP immediately implies the Weak Duality Theorem.

Theorem 19 (Weak Duality)**.** *If x' and y' are feasible solutions to Primal-LP and Dual-LP then $cx' \leq y'b$.*

Corollary 20. *If the primal-LP is unbounded then the Dual-LP is infeasible.*

Exercise: Prove that the dual of the Dual-LP is the Primal-LP.

The main result in the theory of linear programming is the following Strong Duality Theorem which is essentially a min-max result.

Theorem 21 (Strong Duality)**.** *If Primal-LP and Dual-LP have feasible solutions, then there exist feasible solutions x^* and y^* such that $cx^* = y^*b$.*

Proof. Note that by weak duality we have that $cx' \leq y'b$ for any feasible pair of x' and y'. Thus to show the existence of x^* and y^* it suffices to show that the system of inequalities below has a feasible solution whenever the two LPs are feasible.

$$\begin{aligned} cx &\geq yb \\ Ax &\leq b \\ yA &= c \\ y &\geq 0 \end{aligned}$$

We rewrite this as

$$\begin{aligned} Ax &\leq b \\ yA &\leq c \\ -yA &\leq -c \\ -y &\leq 0 \\ -cx + yb &\leq 0 \end{aligned}$$

and apply the Theorem of Alternatives. Note that we have inequalities in $n + m$ variables corresponding to the x and y variables. By expressing those variables as a vector $z = \begin{bmatrix} x \\ y^T \end{bmatrix}$, we have

$$\begin{bmatrix} A & 0 \\ 0 & A^T \\ 0 & -A^T \\ 0 & -I \\ -c & b^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} b \\ c^T \\ -c^T \\ 0 \\ 0 \end{bmatrix}$$

If the above system does not have a solution then there exists a vector $\begin{bmatrix} s & t & t' & u & v \end{bmatrix} \geq 0$, where s is a $m \times 1$ row vector, t, t' are $n \times 1$ row vectors, u is a $m \times 1$ row vector and v is a scalar such that

$$\begin{aligned} sA - v \cdot c &= 0 \\ tA^T - t'A^T - u + v \cdot b &= 0 \\ sb + t \cdot c - t \cdot c' &< 0 \end{aligned}$$

We replace $t - t'$ by w and note that now w is not necessarily positive. Hence we obtain that if the strong duality does not hold then there exist vectors $s, u \in \mathbb{R}^m, w \in \mathbb{R}^n, v \in \mathbb{R}$ such that

$$\begin{aligned} s, u, v &\geq 0 \\ sA - v \cdot c &= 0 \\ wA^T - u + v \cdot b &= 0 \\ sb + wc^T &< 0 \end{aligned}$$

We consider two cases.
Case 1: $v = 0$. (note that v is a scalar.)
In this case, the above system simplifies to

$$\begin{aligned} s &\geq 0 \\ sA &= 0 \\ wA^T &= 0 \\ sb + wc^T &< 0 \end{aligned}$$

Since we have $y^*A = c$ and $sA = 0$, $y^* + \alpha s$ is a feasible solution for the dual for any scalar $\alpha \geq 0$. Similarly knowing that $Ax^* \leq b$ and $Aw^T = 0$ (from $wA^T = 0$), it follows that $x^* - \alpha w^T$ is feasible for the primal for any scalar $\alpha \geq 0$. Applying the Weak Duality Theorem, we have that $\forall \alpha \geq 0$,

$$\begin{aligned} & c(x^* - \alpha w^T) &\leq\ & (y^* + \alpha s) \cdot b \\ \Rightarrow\ & cx^* - y^*b &\leq\ & \alpha(s \cdot b + c \cdot w^T) \end{aligned}$$

However, the LHS is fixed while the RHS can be made arbitrarily small because $s \cdot b + c \cdot w^T < 0$ and α can be chosen arbitrarily large.
Case 2: $v > 0$.
Let $s' = \frac{1}{v}(s)$, $w' = \frac{1}{v}(w)$, and $u' = \frac{1}{v}(u)$. Then, we have

$$\begin{aligned} s', u' &\geq 0 \\ s'A &= c \\ w'A^T - u' &= -b \Rightarrow -A(w')^T = b - u' \leq b \;\; \text{[Since } u' \geq 0.] \\ s'b + w'c^T &< 0 \end{aligned}$$

From the inequalities above, we observe that s' is dual feasible and $-w'$ is primal feasible. Thus by the Weak Duality, we have $-w' \cdot c^T \leq s'b$, contradicting that $s'b + w \cdot c^T < 0$.

Finally, we make a remark on where the contradiction really comes from for each case. For the first case where $v = 0$, note that the inequality $-cx + yb \leq 0$, which forces the optimal values for the primal and the dual meet each other, was never used. In other words, we got a contradiction because there do not exist feasible solutions satisfying both the primal and the dual. On the other hand, for the second case $v > 0$, we had feasible solutions for both LPs, but obtained a contradiction essentially from the assumption that the two optimal values are different. $\square$

Complementary Slackness is a very useful consequence of Strong Duality.

Theorem 22 (Complementary Slackness)**.** *Let x^*, y^* be feasible solutions to the primal and dual LPs. Then x^*, y^* are optimal solutions if and only if $\forall i \in [m]$, either $y_i^* = 0$ or $a_i x^* = b_i$.*

Proof. Let $\alpha_1, \alpha_2, ...\alpha_m$ be the row vectors of A. Suppose that the given condition is satisfied. Then we have $y^*Ax^* = \sum_{i=1}^m y_i^*(\alpha_i x^*) = \sum_{i=1}^m y_i^* b_i = y^*b$. Also we know that $cx^* = y^*Ax^*$ since y^* is a feasible solution for the dual. Thus we have $cx^* = y^*b$, and by the Weak Duality, we conclude that x^*, y^* are optimal.

Conversely, suppose that x^* and y^* both are optimal. Then by the Strong Duality and because x^*, y^* are feasible, we have that $y^*b = cx^* = y^*Ax^* \leq y^*b$. Thus we obtain the equality $y^*Ax^* = y^*b$, that is $\sum_{i=1}^m y_i^*(\alpha_i x^*) = \sum_{i=1}^m y_i^* b_i$. This equality forces the desired condition, since $\alpha_i x^* \leq b_i, y_i^* \geq 0$ because x^* and y^* are feasible solutions. $\square$

References

[1] L. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, Chapters 7, 8, 1998.

[2] J. Lee. *A First Course in Combinatorial Optimization*. Cambridge University Press, 2004.

[3] M. Goemans. Lecture notes on linear programming and polyhedral combinatorics. http://www-math.mit.edu/ goemans/18433S09/polyhedral.pdf, 2009.

3. Linear Programming and Polyhedral Combinatorics

Summary of what was seen in the introductory lectures on linear programming and polyhedral combinatorics.

Definition 3.1 *A halfspace in* $\mathbb{R}^n$ *is a set of the form* $\{x \in \mathbb{R}^n : a^T x \leq b\}$ *for some vector* $a \in \mathbb{R}^n$ *and* $b \in \mathbb{R}$*.*

Definition 3.2 *A polyhedron is the intersection of finitely many halfspaces:* $P = \{x \in \mathbb{R}^n : Ax \leq b\}$*.*

Definition 3.3 *A polytope is a bounded polyhedron.*

Definition 3.4 *If* P *is a polyhedron in* $\mathbb{R}^n$*, the projection* $P_k \subseteq \mathbb{R}^{n-1}$ *of* P *is defined as* $\{y = (x_1, x_2, \cdots, x_{k-1}, x_{k+1}, \cdots, x_n) : x \in P \text{ for some } x_k\}$*.*

This is a special case of a projection onto a linear space (here, we consider only coordinate projection). By repeatedly projecting, we can eliminate any subset of coordinates.

We claim that P_k is also a polyhedron and this can be proved by giving an explicit description of P_k in terms of linear inequalities. For this purpose, one uses *Fourier-Motzkin elimination.* Let $P = \{x : Ax \leq b\}$ and let

- $S_+ = \{i : a_{ik} > 0\}$,
- $S_- = \{i : a_{ik} < 0\}$,
- $S_0 = \{i : a_{ik} = 0\}$.

Clearly, any element in P_k must satisfy the inequality $a_i^T x \leq b_i$ for all $i \in S_0$ (these inequalities do not involve x_k). Similarly, we can take a linear combination of an inequality in S_+ and one in S_- to eliminate the coefficient of x_k. This shows that the inequalities:

$$a_{ik}\left(\sum_j a_{lj}x_j\right) - a_{lk}\left(\sum_j a_{ij}x_j\right) \leq a_{ik}b_l - a_{lk}b_i \tag{1}$$

for $i \in S_+$ and $l \in S_-$ are satisfied by all elements of P_k. Conversely, for any vector $(x_1, x_2, \cdots, x_{k-1}, x_{k+1}, \cdots, x_n)$ satisfying (1) for all $i \in S_+$ and $l \in S_-$ and also

$$a_i^T x \leq b_i \text{ for all } i \in S_0 \tag{2}$$

we can find a value of x_k such that the resulting x belongs to P (by looking at the bounds on x_k that each constraint imposes, and showing that the largest lower bound is smaller than the smallest upper bound). This shows that P_k is described by (1) and (2), and therefore is a polyhedron.

Definition 3.5 *Given points $a^{(1)}, a^{(2)}, \cdots, a^{(k)} \in \mathbb{R}^n$,*

- *a linear combination is $\sum_i \lambda_i a^{(i)}$ where $\lambda_i \in \mathbb{R}$ for all i,*
- *an affine combination is $\sum_i \lambda_i a^{(i)}$ where $\lambda_i \in \mathbb{R}$ and $\sum_i \lambda_i = 1$,*
- *a conical combination is $\sum_i \lambda_i a^{(i)}$ where $\lambda_i \geq 0$ for all i,*
- *a convex combination is $\sum_i \lambda_i a^{(i)}$ where $\lambda_i \geq 0$ for all i and $\sum_i \lambda_i = 1$.*

The set of all linear combinations of elements of S is called the linear hull of S and denoted by $lin(S)$. Similarly, by replacing *linear* by *affine, conical or convex*, we define the affine hull, $aff(S)$, the conic hull, $cone(S)$ and the convex hull, $conv(S)$. We can give an equivalent definition of a polytope.

Definition 3.6 *A polytope is the convex hull of a finite set of points.*

The fact that Definition 3.6 implies Definition 3.3 can be shown by using Fourier-Motzkin elimination repeatedly on

$$x - \sum_k \lambda_k a^{(k)} = 0$$
$$\sum_k \lambda_k = 1$$
$$\lambda_k \geq 0$$

to eliminate all variables λ_k and keep only the variables x. The converse will be discussed later in these notes.

3.1 Solvability of System of Inequalities

In linear algebra, we saw that, for $A \in R^{m\times n}$, $b \in R^m$, $Ax = b$ has no solution $x \in \mathbb{R}^n$ if and only if there exists a $y \in R^m$ with $A^T y = 0$ and $b^T y \neq 0$ (in 18.06 notation/terminology, this is equivalent to saying that the column space $C(A)$ is orthogonal to the left null space $N(A^T)$).

One can state a similar *Theorem of the Alternatives* for systems of linear inequalities.

Theorem 3.1 (Theorem of the Alternatives) *$Ax \leq b$ has no solution $x \in \mathbb{R}^n$ if and only if there exists $y \in \mathbb{R}^m$ such that $y \geq 0$, $A^T y = 0$ and $b^T y < 0$.*

One can easily show that both systems indeed cannot have a solution since otherwise $0 > b^T y = y^T b \geq y^T Ax = 0^T x = 0$. For the other direction, one takes the insolvable system $Ax \leq b$ and use Fourier-Motzkin elimination repeatedly to eliminate all variables and thus obtain an inequality like $0^T x \leq c$ where $c < 0$. In the process one has derived a vector y with the desired properties (as Fourier-Motzkin only performs nonnegative combinations of linear inequalities).

Another version of the above theorem is Farkas' lemma:

Lemma 3.2 *$Ax = b$, $x \geq 0$ has no solution if and only if there exists y with $A^T y \geq 0$ and $b^T y < 0$.*

Exercise 3-1. Prove Farkas' lemma from the Theorem of the Alternatives.

3.2 Linear Programming Basics

A linear program (LP) is the problem of minimizing or maximizing a linear function over a polyhedron:

$$
\begin{array}{lrl}
 & \text{Max} & c^T x \\
 & \text{subject to:} & \\
(P) & & Ax \leq b,
\end{array}
$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and the variables x are in $\mathbb{R}^n$. Any x satisfying $Ax \leq b$ is said to be *feasible*. If no x satisfies $Ax \leq b$, we say that the linear program is *infeasible*, and its optimum value is $-\infty$ (as we are maximizing over an empty set). If the objective function value of the linear program can be made arbitrarily large, we say that the linear program is *unbounded* and its optimum value is $+\infty$; otherwise it is *bounded*. If it is neither infeasible, not unbounded, then its optimum value is finite.

Other equivalent forms involve equalities as well, or nonnegative constraints $x \geq 0$. One version that is often considered when discussing algorithms for linear programming (especially the simplex algorithm) is $\min\{c^T x : Ax = b, x \geq 0\}$.

Another linear program, *dual* to (P), plays a crucial role:

$$
\begin{array}{lrl}
 & \text{Min} & b^T y \\
 & \text{subject to:} & \\
(D) & & A^T y = c \\
 & & y \geq 0.
\end{array}
$$

(D) is the dual and (P) is the *primal*. The terminology for the dual is similar. If (D) has no feasible solution, it is said to be *infeasible* and its optimum value is $+\infty$ (as we are minimizing over an empty set). If (D) is unbounded (i.e. its value can be made arbitrarily negative) then its optimum value is $-\infty$.

The primal and dual spaces should not be confused. If A is $m \times n$ then we have n primal variables and m dual variables.

Weak duality is clear: For any feasible solutions x and y to (P) and (D), we have that $c^T x \leq b^T y$. Indeed, $c^T x = y^T Ax \leq b^T y$. The dual was precisely built to get an upper bound on the value of any primal solution. For example, to get the inequality $y^T Ax \leq b^T y$, we need that $y \geq 0$ since we know that $Ax \leq b$. In particular, weak duality implies that if the primal is unbounded then the dual must be infeasible.

Strong duality is the most important result in linear programming; it says that we can prove the optimality of a primal solution x by exhibiting an optimum dual solution y.

Theorem 3.3 (Strong Duality) *Assume that* (P) *and* (D) *are feasible, and let* z^* *be the optimum value of the primal and* w^* *the optimum value of the dual. Then* $z^* = w^*$.

The proof of strong duality is obtained by writing a big system of inequalities in x and y which says that (i) x is primal feasible, (ii) y is dual feasible and (iii) $c^T x \geq b^T y$. Then use the Theorem of the Alternatives to show that the infeasibility of this system of inequalities would contradict the feasibility of either (P) or (D).

Proof: Let x^* be a feasible solution to the primal, and y^* be a feasible solution to the dual. The proof is by contradiction. Because of weak duality, this means that there are no solution $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ such that

$$\begin{cases} Ax & & \leq b \\ & A^T y & = c \\ & Iy & \leq 0 \\ -c^T x & +b^T y & \leq 0 \end{cases}$$

By a variant of the Theorem of the Alternatives or Farkas' lemma (for the case when we have a combination of inequalities and equalities), we derive that there must exist $s \in \mathbb{R}^m$, $t \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $v \in \mathbb{R}$ such that:

$$\begin{aligned} & s \geq 0 \\ & u \geq 0 \\ & v \geq 0 \\ & A^T s - vc = 0 \\ & At - u + vb = 0 \\ & b^T s + c^T t < 0. \end{aligned}$$

We distinguish two cases.

Case 1: $v = 0$**.** Then s satisfies $s \geq 0$ and $A^T s = 0$. This means that, for any $\alpha \geq 0$, $y^* + \alpha s$ is feasible for the dual. Similarly, $At = u \geq 0$ and therefore, for any $\alpha \geq 0$, we have that $x^* - \alpha t$ is primal feasible. By weak duality, this means that, for any $\alpha \geq 0$, we have

$$c^T(x^* - \alpha t) \leq b^T(y^* + \alpha s)$$

or

$$c^T x^* - b^T y^* \leq \alpha(b^T s + c^T t).$$

The right-hand-side tend to $-\infty$ as α tends to ∞, and this is a contradiction as the left-hand-side is fixed.

Case 2: $v > 0$. By dividing throughout by v (and renaming all the variables), we get that there exists $s \geq 0$, $u \geq 0$ with

$$\begin{aligned} A^T s &= c \\ At - u &= -b \\ b^T s + c^T t &< 0. \end{aligned}$$

This means that s is dual feasible and $-t$ is primal feasible, and therefore by weak duality $c^T(-t) \leq b^T s$ contradicting $b^T s + c^T t < 0$. $\triangle$

Exercise 3-2. Show that the dual of the dual is the primal.

Exercise 3-3. Show that we only need either the primal or the dual to be feasible for strong duality to hold. More precisely, if the primal is feasible but the dual is infeasible, prove that the primal will be unbounded, implying that $z^* = w^* = +\infty$.

Looking at $c^T x = y^T A x \leq b^T y$, we observe that to get equality between $c^T x$ and $b^T y$, we need *complementary slackness*:

Theorem 3.4 (Complementary Slackness) *If x is feasible in (P) and y is feasible in (D) then x is optimum in (P) and y is optimum in (D) if and only if for all i either $y_i = 0$ or $\sum_j a_{ij} x_j = b_i$ (or both).*

Linear programs can be solved using the simplex method; this is not going to be explained in these notes. No variant of the simplex method is known to provably run in polynomial time, but there are other polynomial-time algorithms for linear programming, namely the ellipsoid algorithm and the class of interior-point algorithms.

3.3 Faces of Polyhedra

Definition 3.7 *$\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ are linearly independent if $\sum_i \lambda_i a^{(i)} = 0$ implies that $\lambda_i = 0$ for all $i \in K$.*

Definition 3.8 *$\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ are affinely independent if $\sum_i \lambda_i a^{(i)} = 0$ and $\sum_i \lambda_i = 0$ together imply that $\lambda_i = 0$ for all $i \in K$.*

Observe that $\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ are *affinely* independent if and only if

$$\left\{ \begin{bmatrix} a^{(i)} \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1} : i \in K \right\}$$

are *linearly* independent.

Definition 3.9 *The dimension, $dim(P)$, of a polyhedron P is the maximum number of affinely independent points in P minus 1.*

The dimension can be -1 (if P is empty), 0 (when P consists of a single point), 1 (when P is a line segment), and up to n when P affinely spans $\mathbb{R}^n$. In the latter case, we say that P is *full-dimensional*. The dimension of a cube in $\mathbb{R}^3$ is 3, and so is the dimension of $\mathbb{R}^3$ itself (which is a trivial polyhedron).

Definition 3.10 *$\alpha^T x \leq \beta$ is a valid inequality for P if $\alpha^T x \leq \beta$ for all $x \in P$.*

Observe that for an inequality to be valid for $conv(S)$ we only need to make sure that it is satisfied by all elements of S, as this will imply that the inequality is also satisfied by points in $conv(S) \setminus S$. This observation will be important when dealing with convex hulls of combinatorial objects such as matchings or spanning trees.

Definition 3.11 *A face of a polyhedron P is $\{x \in P : \alpha^T x = \beta\}$ where $\alpha^T x \leq \beta$ is some valid inequality of P.*

By definition, all faces are polyhedra. The empty face (of dimension -1) is *trivial*, and so is the entire polyhedron P (which corresponds to the valid inequality $0^T x \leq 0$). Non-trivial are those whose dimension is between 0 and $dim(P) - 1$. Faces of dimension 0 are called *extreme points* or *vertices*, faces of dimension 1 are called *edges*, and faces of dimension $dim(P) - 1$ are called *facets*. Sometimes, one uses *ridges* for faces of dimension $dim(P) - 2$.

Exercise 3-4. List all 28 faces of the cube $P = \{x \in \mathbb{R}^3 : 0 \leq x_i \leq 1 \text{ for } i = 1, 2, 3\}$.

Although there are infinitely many valid inequalities, there are only finitely many faces.

Theorem 3.5 *Let $A \in \mathbb{R}^{m \times n}$. Then any non-empty face of $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ corresponds to the set of solutions to*

$$\sum_j a_{ij} x_j = b_i \textit{ for all } i \in I$$

$$\sum_j a_{ij} x_j \leq b_i \textit{ for all } i \notin I,$$

for some set $I \subseteq \{1, \cdots, m\}$. Therefore, the number of non-empty faces of P is at most 2^m.

Proof: Consider any valid inequality $\alpha^T x \leq \beta$. Suppose the corresponding face F is non-empty. Thus F are all optimum solutions to

$$\begin{array}{lll} & \text{Max} & \alpha^T x \\ & \text{subject to:} & \\ (P) & & Ax \leq b. \end{array}$$

Choose an optimum solution y^* to the dual LP. By complementary slackness, the face F is defined by those elements x of P such that $a_i^T x = b_i$ for $i \in I = \{i : y_i^* > 0\}$. Thus F is defined by

$$\sum_j a_{ij} x_j = b_i \text{ for all } i \in I$$

$$\sum_j a_{ij} x_j \leq b_i \text{ for all } i \notin I.$$

As there are 2^m possibilities for F, there are at most 2^m non-empty faces. △

The number of faces given in Theorem 3.5 is tight for polyhedra (see exercise below), but can be considerably improved for polytopes in the so-called *upper bound theorem.*

Exercise 3-5. Let $P = \{x \in \mathbb{R}^n : x_i \geq 0 \text{ for } i = 1, \cdots, n\}$. Show that P has $2^n + 1$ faces. How many faces of dimension k does P have?

For extreme points (faces of dimension 0), the characterization is even stronger (we do not need the inequalities):

Theorem 3.6 *Let x^* be an extreme point for $P = \{x : Ax \leq b\}$. Then there exists I such that x^* is the unique solution to*

$$\sum_j a_{ij} x_j = b_i \text{ for all } i \in I.$$

Proof: Given an extreme point x^*, define I by $I = \{i : \sum_j a_{ij} x_j^* = b_i\}$. This means that for $i \notin I$, we have $\sum_j a_{ij} x_j^* < b_i$.

From Theorem 3.5, we know that x^* is uniquely defined by

$$\sum_j a_{ij} x_j = b_i \text{ for all } i \in I \tag{3}$$

$$\sum_j a_{ij} x_j \leq b_i \text{ for all } i \notin I. \tag{4}$$

Now suppose there exists another solution $\hat{x}$ *when* we consider only the equalities for $i \in I$. Then because of $\sum_j a_{ij} x_j^* < b_i$, we get that $(1-\epsilon)x^* + \epsilon\hat{x}$ also satisfies (3) and (4) for ϵ sufficiently small. A contradiction (as the face was supposed to contain a single point). △

If P is given as $\{x : Ax = b, x \geq 0\}$ (as is often the case), the theorem still applies (as we still have a system of inequalities). In this case, the theorem says that every extreme point x^* can be obtained by setting some of the variables to 0, and solving for the unique solution to the resulting system of equalities. Without loss of generality, we can remove from $Ax = b$ equalities that are redundant; this means that we can assume that A has full row rank ($rank(A) = m$ for $A \in \mathbb{R}^{m \times n}$). Letting N denote the indices of the *non-basic* variables that we set of 0 and B denote the remaining indices (of the so-called *basic* variables), we can partition x^* into x^*_B and x^*_N (corresponding to these two sets of variables) and rewrite $Ax = b$ as $A_B x_B + A_N x_N = b$, where A_B and A_N are the restrictions of A to the indices in B and N respectively. The theorem says that x^* is the unique solution to $A_B x_B + A_N x_N = 0$ and $x_N = 0$, which means $x^*_N = 0$ and $A_B x^*_B = b$. This latter system must have a unique solution, which means that A_B must have full column rank ($rank(A_B) = |B|$). As A itself has rank m, we have that $|B| \leq m$ and we can augment B to include indices of N such that the resulting B satisfies (i) $|B| = m$ and (ii) A_B is a $m \times m$ invertible matrix (and thus there is still a unique solution to $A_B x_B = b$). In linear programming terminology, a *basic feasible solution* or *bfs* of $\{x : Ax = b, x \geq 0\}$ is obtained by choosing a set $|B| = m$ of indices with A_B invertible and letting $x_B = A_B^{-1} b$ and $x_N = 0$ where N are the indices not in B. All extreme points are bfs and vice versa (although two different bases B may lead to the same extreme point, as there might be many ways of extending A_B into a $m \times m$ invertible matrix in the discussion above).

One consequence of Theorem 3.5 is:

Corollary 3.7 *The maximal (inclusion-wise) non-trivial faces of a non-empty polyhedron P are the facets.*

For the vertices, one needs one additional condition:

Corollary 3.8 *If $rank(A) = n$ (full column rank) then the minimal (inclusion-wise) non-trivial faces of a non-empty polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ are the vertices.*

Exercise 3-7 shows that the rank condition is necessary.

This means that, if a linear program $\max\{c^T x : x \in P\}$ with $P = \{x : Ax \leq b\}$ is feasible, bounded and $rank(A) = n$, then there exists an optimal solution which is a vertex of P (indeed, the set of all optimal solutions define a face — the optimal face — and if this face is not itself a vertex of P, it must contain vertices of P).

We now prove Corollary 3.8.

Proof: Let F be a minimal (inclusion-wise) non-trivial face of F. This means that we have a set I such that

$$F = \{x : \begin{array}{ll} a_i^T x = b_i & i \in I \\ a_j^T x \leq b_j & j \notin I\} \end{array}$$

and adding any element to I makes this set empty. Consider two cases. Either $P = \{x \in \mathbb{R}^n : a_i^T x = b_i \text{ for } i \in I\}$ or not. In the first case, it means that for every $j \notin I$ we have $a_j \in lin(\{a_i : i \in I\})$ and therefore since $rank(A) = n$ we have that the system $a_i^T x = b_i$ for $i \in I$ has a unique solution and thus F is a vertex.

On the other hand, if $P \neq \{x \in \mathbb{R}^n : a_i^T x = b_i \text{ for } i \in I\}$ then let $j \notin I$ such that there exists $\tilde{x}$ with

$$\begin{array}{ll} a_i^T \tilde{x} = b_i & i \in I \\ a_j^T \tilde{x} > b_j. & \end{array}$$

Since F is not trivial, there exists $\hat{x} \in F$, and in particular, $\hat{x}$ satisfies

$$\begin{array}{ll} a_i^T \hat{x} = b_i & i \in I \\ a_j^T \hat{x} \leq b_j. & \end{array}$$

Thus for a suitable convex combination x' of $\hat{x}$ and $\tilde{x}$, we have $a_i^T x' = b_i$ for $i \in I \cup \{j\}$, contrdicting the maximality of I. $\triangle$

We now go back to the equivalence between Definitions 3.3 and 3.6 and claim that we can show that Definition 3.3 implies Definition 3.6.

Theorem 3.9 *If $P = \{x : Ax \leq b\}$ is bounded then $P = conv(X)$ where X is the set of extreme points of P.*

This is a nice exercise using the Theorem of the Alternatives.

Proof: Since $X \subseteq P$, we have $conv(X) \subseteq P$. Assume, by contradiction, that we do not have equality. Then there must exist $\tilde{x} \in P \setminus conv(X)$. The fact that $\tilde{x} \notin conv(X)$ means that there is no solution to:

$$\begin{cases} \sum_{v \in X} \lambda_v v = \tilde{x} \\ \sum_{v \in X} \lambda_v = 1 \\ \lambda_v \geq 0 \qquad\qquad v \in X. \end{cases}$$

By the Theorem of the alternatives, this implies that $\exists c \in \mathbb{R}^n, t \in \mathbb{R}$:

$$\begin{cases} t + \sum_{j=1}^n c_j v_j \geq 0 \quad \forall v \in X \\ t + \sum_{j=1}^n c_j \tilde{x}_j < 0. \end{cases}$$

Since P is bounded, $\min\{c^T x : x \in P\}$ is finite (say equal to z^*), and the face induced by $c^T x \geq z^*$ is non-empty but does not contain any vertex (as all vertices are dominated by $\tilde{x}$ by the above inequalities). This is a contradiction with Corollary 3.8. $\triangle$

When describing a polyhedron P in terms of linear inequalities, the only inequalities that are needed are the ones that define facets of P. This is stated in the next few theorems. We say that an inequality in the system $Ax \leq b$ is *redundant* if the corresponding polyhedron is unchanged by removing the inequality. For $P = \{x : Ax \leq b\}$, we let $I_=$ denote the indices i such that $a_i^T x = b_i$ for all $x \in P$, and $I_<$ the remaining ones (i.e. those for which there exists $x \in P$ with $a_i^T x < b_i$).

This theorem shows that facets are sufficient:

Theorem 3.10 *If the face associated with $a_i^T x \leq b_i$ for $i \in I_<$ is not a facet then the inequality is redundant.*

And this one shows that facets are necessary:

Theorem 3.11 *If F is a facet of P then there must exists $i \in I_<$ such that the face induced by $a_i^T x \leq b_i$ is precisely F.*

In a *minimal* description of P, we must have a set of *linearly independent equalities* together with precisely one inequality for each facet of P.

Exercises

Exercise 3-6. Prove Corollary 3.7.

Exercise 3-7. Show that if $rank(A) < n$ then $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ has no vertices.

Exercise 3-8. Suppose $P = \{x \in \mathbb{R}^n : Ax \leq b, Cx \leq d\}$. Show that the set of vertices of $Q = \{x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$ is a subset of the set of vertices of P.

(In particular, this means that if the vertices of P all belong to $\{0,1\}^n$, then so do the vertices of Q.)

Exercise 3-9. Given two extreme points a and b of a polyhedron P, we say that they are *adjacent* if the line segment between them forms an edge (i.e. a face of dimension 1) of the polyhedron P. This can be rephrased by saying that a and b are adjacent on P if and only if there exists a cost function c such that a and b are the only two extreme points of P minimizing $c^T x$ over P.

Consider the polyhedron (polytope) P defined as the convex hull of all perfect matchings in a (not necessarily bipartite) graph G. Give a necessary and sufficient condition for two matchings M_1 and M_2 to be adjacent on this polyhedron (hint: think about $M_1 \triangle M_2 = (M_1 \setminus M_2) \cup (M_2 \setminus M_1)$) and prove that your condition is necessary and sufficient.)

Exercise 3-10. Show that two vertices u and v of a polyhedron P are adjacent if and only there is a unique way to express their midpoint ($\frac{1}{2}(u+v)$) as a convex combination of vertices of P.

3.4 Polyhedral Combinatorics

In one sentence, polyhedral combinatorics deals with the study of polyhedra or polytopes associated with discrete sets arising from combinatorial optimization problems (such as matchings for example). If we have a discrete set X (say the incidence vectors of matchings in a graph, or the set of incidence vectors of spanning trees of a graph, or the set of incidence vectors of *stable* sets[1] in a graph), we can consider $conv(X)$ and attempt to describe it in terms of linear inequalities. This is useful in order to apply the machinery of linear programming. However, in some (most) cases, it is actually hard to describe the set of all inequalities defining $conv(X)$; this occurs whenever optimizing over X is hard and this statement can be made precise in the setting of computational complexity. For matchings, or spanning trees, or several other structures (for which the corresponding optimization problem is polynomially solvable), we will be able to describe their convex hull in terms of linear inequalities.

Given a set X and a proposed system of inequalities $P = \{x : Ax \leq b\}$, it is usually easy to check whether $conv(X) \subseteq P$. Indeed, for this, we only need to check that every member of X satisfies every inequality in the description of P. The reverse inclusion is more difficult.

[1]A set S of vertices in a graph $G = (V, E)$ is stable if there are no edges between any two vertices of S.

Here are 3 general techniques to prove that $P \subseteq conv(X)$ (if it is true!) (once we know that $conv(X) \subseteq P$).

1. **Algorithmically.** This involves linear programming duality. This is what we did in the notes about the assignment problem (minimum weight matchings in bipartite graphs). In general, consider any cost function c and consider the combinatorial optimization problem of maximizing $c^T x$ over $x \in X$. We know that:

$$\begin{aligned} \max\{c^T x : x \in X\} &= \max\{c^T x : x \in conv(X)\} \\ &\leq \max\{c^T x : Ax \leq b\} \\ &= \min\{b^T y : A^T y = c, y \geq 0\}, \end{aligned}$$

   the last equality coming from strong duality. If we can exhibit a solution $x \in X$ (say a perfect matching in the assignment problem) and a dual feasible solution y (values u_i, v_j in the assignment problem) such that $c^T x = b^T y$ we will have shown that we have equality throughout, and if this is true for any cost function, this implies that $P = conv(X)$.

   This is usually the most involved approach but also the one that works most often.

2. **Focusing on extreme points.** Show first that $P = \{x : Ax \leq b\}$ is bounded (thus a polytope) and then study its extreme points. If we can show that every extreme point of P is in X then we would be done since $P = conv(ext(P)) \subseteq conv(X)$, where $ext(P)$ denotes the extreme points of P (see Theorem 3.9). The assumption that P is bounded is needed to show that indeed $P = conv(ext(P))$ (not true if P is unbounded).

   In the case of the convex hull of bipartite matchings, this can be done easily and this leads to the notion of Totally Unimodular Matrices (TUM), see the next section.

3. **Focusing on the facets of $conv(X)$.** This leads usually to the shortest and cleanest proofs. Suppose that our proposed P is of the form $\{x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$. We have already argued that $conv(X) \subseteq P$ and we want to show that $P \subseteq conv(X)$.

   First we need to show that we are not missing any equality. This can be done for example by showing that $dim(conv(X)) = dim(P)$ (i.e. showing that if there are $n - d$ linearly independent rows in C we can find $d + 1$ affinely independent points in X).

   Then we need to show that we are not missing a valid inequality that induces a *facet* of $conv(X)$. Consider any valid inequality $\alpha^T x \leq \beta$ for $conv(X)$ with $\alpha \neq 0$. We can assume that α is any vector in $\mathbb{R}^n \setminus \{0\}$ and that $\beta = \max\{\alpha^T x : x \in conv(X)\}$. The face of $conv(X)$ this inequality defines is $F = conv(\{x \in X : \alpha^T x = \beta\})$. Assume that this is a non-trivial face; this will happen precisely when α is not in the row space of C. We need to make sure that if F is facet then we have in our description of P an inequality representing it. What we will show is that if F is non-trivial then we can find an inequality $a_i^T x \leq b_i$ in our description of P such that $F \subseteq \{x : a_i^T x = b_i\}$, or simply that every optimum solution to $\max\{\alpha^T x : x \in X\}$ satisfies $a_i^T x = b_i$. This means that if F was a facet, by maximality, we have a representative of F in our description.

This is a very simple and powerful technique, and this is best illustrated on an example.

Example. Let $X = \{(\sigma(1), \sigma(2), \cdots, \sigma(n)) : \sigma \text{ is a permutation of } \{1, 2, \cdots, n\}\}$. We claim that

$$conv(X) = \{x \in \mathbb{R}^n : \quad \begin{array}{ll} \sum_{i=1}^n x_i = \binom{n+1}{2} & \\ \sum_{i \in S} x_i \geq \binom{|S|+1}{2} & S \subset \{1, \cdots, n\}\}. \end{array}$$

This is known as the *permutahedron*.

Here $conv(X)$ is not full-dimensional; we only need to show that we are not missing any facets and any equality in the description of $conv(P)$. For the equalities, this can be seen easily as it is easy to exhibit n affinely independent permutations in X. For the facets, suppose that $\alpha^T x \leq \beta$ defines a non-trivial facet F of $conv(X)$. Consider maximizing $\alpha^T x$ over all permutations x. Let $S = \arg\min\{\alpha_i\}$; by our assumption that F is non-trivial we have that $S \neq \{1, 2, \cdots, n\}$ (otherwise, we would have the equality $\sum_{i=1}^n x_i = \binom{n+1}{2}$). Moreover, it is easy to see (by an exchange argument) that any permutationσ whose incidence vector x maximizes $\alpha^T x$ will need to satisfy $\sigma(i) \in \{1, 2, \cdots, |S|\}$ for $i \in S$, in other words, it will satisfy the inequality $\sum_{i \in S} x_i \geq \binom{|S|+1}{2}$ at equality. Hence, F is contained in the face corresponding to an inequality in our description, and hence our description contains inequalities for all facets. This is what we needed to prove. That's it!

Exercises

Exercise 3-11. Consider the set $X = \{(\sigma(1), \sigma(2), \cdots, \sigma(n)) : \sigma \text{ is a permutation of } \{1, 2 \cdots, n\}\}$. Show that $\dim(conv(X)) = n - 1$. (To show that $\dim(conv(X)) \geq n - 1$, exhibit n affinely independent permutations σ (and prove that they are affinely independent).)

Exercise 3-12. A *stable set* S (sometimes, it is called also an independent set) in a graph $G = (V, E)$ is a set of vertices such that there are no edges between any two vertices in S. If we let P denote the convex hull of all (incidence vectors of) stable sets of $G = (V, E)$, it is clear that $x_i + x_j \leq 1$ for any edge $(i, j) \in E$ is a valid inequality for P.

1.Give a graph G for which P is *not* equal to

$$\{x \in \mathbb{R}^{|V|} : \quad \begin{array}{ll} x_i + x_j \leq 1 & \text{for all } (i,j) \in E \\ x_i \geq 0 & \text{for all } i \in V\} \end{array}$$

2.Show that if the graph G is bipartite then P equals

$$\{x \in \mathbb{R}^{|V|} : \quad \begin{array}{ll} x_i + x_j \leq 1 & \text{for all } (i,j) \in E \\ x_i \geq 0 & \text{for all } i \in V\}. \end{array}$$

Exercise 3-13. Let $e_k \in \mathbb{R}^n$ ($k = 0, \ldots, n-1$) be a vector with the first k entries being 1, and the following $n-k$ entries being -1. Let $S = \{e_0, e_1, \ldots, e_{n-1}, -e_0, -e_1, \ldots, -e_{n-1}\}$, i.e. S consists of all vectors consisting of $+1$ followed by -1 or vice versa. In this problem set, you will study *conv*(S).

1. Consider any vector $a \in \{-1, 0, 1\}^n$ such that (i) $\sum_{i=1}^n a_i = 1$ and (ii) for all $j = 1, \ldots, n-1$, we have $0 \leq \sum_{i=1}^j a_i \leq 1$. (For example, for $n = 5$, the vector $(1, 0, -1, 1, 0)$ satisfies these conditions.) Show that $\sum_{i=1}^n a_i x_i \leq 1$ and $\sum_{i=1}^n a_i x_i \geq -1$ are valid inequalities for *conv*(S).
2. How many such inequalities are there?
3. Show that any such inequality defines a facet of *conv*(S).

   (This can be done in several ways. Here is one approach, but you are welcome to use any other one as well. First show that either e_k or $-e_k$ satisfies this inequality at equality, for any k. Then show that the resulting set of vectors on the hyperplane are affinely independent (or uniquely identifies it).)
4. Show that the above inequalities define the entire convex hull of S.

   (Again this can be done in several ways. One possibility is to consider the 3rd technique described above.)

3.5 Total unimodularity

Definition 3.12 *A matrix A is totally unimodular if every square submatrix of A has determinant $-1, 0$ or $+1$.*

The importance of total unimodularity stems from the following theorem. This theorem gives a subclass of integer programs which are easily solved. A polyhedron P is said to be *integral* if all its vertices or extreme points are integral (belong to $\mathbb{Z}^n$).

Theorem 3.12 *Let A be a totally unimodular matrix. Then, for any integral right-hand-side b, the polyhedron*

$$P = \{x : Ax \leq b, x \geq 0\}$$

is integral.

Before we prove this result, two remarks can be made. First, the proof below will in fact show that the same result holds for the polyhedrons $\{x : Ax \geq b, x \geq 0\}$ or $\{x : Ax = b, x \geq 0\}$. In the latter case, though, a slightly weaker condition than totally unimodularity is sufficient to prove the result. Secondly, in the above theorem, one can prove the converse as well: If $P = \{x : Ax \leq b, x \geq 0\}$ is integral for all integral b then A must be totally unimodular (this is not true though, if we consider for example $\{x : Ax = b, x \geq 0\}$).

Proof: Adding slacks, we get the polyhedron $Q = \{(x, s) : Ax + Is = b, x \geq 0, s \geq 0\}$. One can easily show (see exercise below) that P is integral iff Q is integral.

Consider now any bfs of Q. The basis B consists of some columns of A as well as some columns of the identity matrix I. Since the columns of I have only one nonzero entry per column, namely a one, we can expand the determinant of A_B along these entries and derive that, in absolute values, the determinant of A_B is equal to the determinant of some square submatrix of A. By definition of totally unimodularity, this implies that the determinant of A_B must belong to $\{-1, 0, 1\}$. By definition of a basis, it cannot be equal to 0. Hence, it must be equal to ± 1.

We now prove that the bfs must be integral. The non-basic variables, by definition, must have value zero. The vector of basic variables, on the other hand, is equal to $A_B^{-1}b$. From linear algebra, A_B^{-1} can be expressed as

$$\frac{1}{\det A_B} A_B^{adj}$$

where A_B^{adj} is the adjoint (or adjugate) matrix of A_B and consists of subdeterminants of A_B. Hence, both b and A_B^{adj} are integral which implies that $A_B^{-1}b$ is integral since $|\det A_B| = 1$. This proves the integrality of the bfs. △

Exercise 3-14. Let $P = \{x : Ax \leq b, x \geq 0\}$ and let $Q = \{(x, s) : Ax + Is = b, x \geq 0, s \geq 0\}$. Show that x is an extreme point of P iff $(x, b - Ax)$ is an extreme point of Q. Conclude that whenever A and b have only integral entries, P is integral iff Q is integral.

In the case of the bipartite matching problem, the constraint matrix A has a very special structure and we show below that it is totally unimodular. This alongs with Theorem 3.12 proves Theorem 1.6 from the notes on the bipartite matching problem. First, let us restate the setting. Suppose that the bipartition of our bipartite graph is (U, V) (to avoid any confusion with the matrix A or the basis B). Consider

$$\begin{aligned}
P \quad = \quad \{x : \quad & \sum_j x_{ij} = 1 & & i \in U \\
& \sum_i x_{ij} = 1 & & j \in V \\
& x_{ij} \geq 0 & & i \in U, j \in V\} \\
= \quad \{x : \quad & Ax = b, x \geq 0\}. & &
\end{aligned}$$

Theorem 3.13 *The matrix A is totally unimodular.*

The way we defined the matrix A corresponds to a *complete* bipartite graph. If we were to consider any bipartite graph then we would simply consider a submatrix of A, which is also totally unimodular by definition.

Proof: Consider any square submatrix T of A. We consider three cases. First, if T has a column or a row with all entries equal to zero then the determinant is zero. Secondly, if there exists a column or a row of T with only one $+1$ then by expanding the determinant along that +1, we can consider a smaller sized matrix T. The last case is when T has at least two nonzero entries per column (and per row). Given the special structure of A, there must in fact be *exactly* 2 nonzero entries per column. By adding up the rows of T corresponding to the vertices of U and adding up the rows of T corresponding to the vertices of V, one therefore obtains the same vector which proves that the rows of T are linearly dependent, implying that its determinant is zero. This proves the totally unimodularity of A. $\triangle$

We conclude with a technical remark. One should first remove one of the rows of A before applying Theorem 3.12 since, as such, it does not have full row rank and this fact was implicitly used in the definition of a bfs. However, deleting a row of A still preserves its totally unimodularity.

Exercise 3-15. If A is totally unimodular then A^T is totally unimodular.

Exercise 3-16. Use total unimodularity to prove König's theorem.

The following theorem gives a necessary and sufficient condition for a matrix to be totally unimodular.

Theorem 3.14 *Let A be a $m \times n$ matrix with entries in $\{-1, 0, 1\}$. Then A is TUM if and only if for all subsets $R \subseteq \{1, 2, \cdots, n\}$ of rows, there exists a partition of R into R_1 and R_2 such that for all $j \in \{1, 2, \cdots, m\}$:*

$$\sum_{i \in R_1} a_{ij} - \sum_{i \in R_2} a_{ij} \in \{0, 1, -1\}.$$

We will prove only the *if* direction.

Proof: Assume that, for every R, the desired partition exists. We need to prove that the determinant of any $k \times k$ submatrix of A is in $\{-1, 0, 1\}$, and this must be true for any k. Let us prove it by induction on k. It is trivially true for $k = 1$. Assume it is true for $k - 1$ and we will prove it for k.

Let B be a $k \times k$ submatrix of A, and we can assume that B is invertible (otherwise the determinant is 0 and there is nothing to prove). The inverse B^{-1} can be written as $\frac{1}{\det(B)} B^*$, where all entries of B^* correspond to $(k-1) \times (k-1)$ submatrices of A. By our inductive hypothesis, all entries of B^* are in $\{-1, 0, 1\}$. Let b_1^* be the first row of B and e_1 be the k-dimensional row vector $[1\ 0\ 0 \cdots 0]$, thus $b_1^* = e_1 B^*$. By the relationship between B and B^*, we have that

$$b_1^* B = e_1 B^* B = \det(B) e_1 B^{-1} B = \det(B) e_1. \tag{5}$$

Let $R = \{i : b_{1i}^* \in \{-1, 1\}\}$. By assumption, we know that there exists a partition of R into R_1 and R_2 such that for all j:

$$\sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij} \in \{-1, 0, 1\}. \tag{6}$$

From (5), we have that

$$\sum_{i\in R} b^*_{1i} b_{ij} = \begin{cases} \det(B) & j = 1 \\ 0 & j \neq 1 \end{cases} \tag{7}$$

Since the left-hand-sides of equations (6) and (7) differ by a multiple of 2 for each j (since $b^*_{1i} \in \{-1, 1\}$), this implies that

$$\sum_{i\in R_1} b_{ij} - \sum_{i\in R_2} b_{ij} \begin{cases} = 0 & j \neq 1 \\ \in \{-1, 1\} & j = 1 \end{cases} \tag{8}$$

The fact that we could not get 0 for $j = 1$ follows from the fact that otherwise B would be singular (we would get exactly the 0 vector by adding and subtracting rows of B). If we define $y \in \mathbb{R}^k$ by

$$y_i = \begin{cases} 1 & i \in R_1 \\ -1 & i \in R_2 \\ 0 & otherwise \end{cases}$$

we get that $yB = \pm e_1$. Thus

$$y = \pm e_1 B^{-1} = \pm \frac{1}{\det B} e_1 B^* = \pm \frac{1}{\det B} b^*_1,$$

which implies that $\det B$ must be either 1 or -1. △

Exercise 3-17. Suppose we have n activities to choose from. Activity i starts at time t_i and ends at time u_i (or more precisely just before u_i); if chosen, activity i gives us a profit of p_i units. Our goal is to choose a subset of the activities which do not overlap (nevertheless, we can choose an activity that ends at t and one that starts at the same time t) and such that the total profit (i.e. sum of profits) of the selected activities is maximum.

1. Defining x_i as a variable that represents whether activity i is selected ($x_i = 1$) or not ($x_i = 0$), write an integer program of the form $\max\{p^T x : Ax \leq b, x \in \{0, 1\}^n\}$ that would solve this problem.
2. Show that the matrix A is totally unimodular, implying that one can solve this problem by solving the linear program $\max\{p^T x : Ax \leq b, 0 \leq x_i \leq 1 \text{ for every } i\}$.

1 More Background on Polyhedra

This material is mostly from [3].

1.1 Implicit Equalities and Redundant Constraints

Throughout this lecture we will use **affhull** to denote the affine hull, **linspace** to be the linear space, **charcone** to denote the characteristic cone and **convexhull** to be the convex hull. Recall that $P = \{x \mid Ax \leq b\}$ is a *polyhedron* in $\mathbb{R}^n$ where A is a $m \times n$ matrix and b is a $m \times 1$ matrix. An inequality $a_i x \leq b_i$ in $Ax \leq b$ is an *implicit equality* if $a_i x = b_i \ \forall x \in P$. Let $I \subseteq \{1, 2, \ldots, m\}$ be the index set of all implicit equalities in $Ax \leq b$. Then we can partition A into $A^= x \leq b^=$ and $A^+ x \leq b^+$. Here $A^=$ consists of the rows of A with indices in I and A^+ are the remaining rows of A. Therefore, $P = \{x \mid A^= x = b^=, A^+ x \leq b^+\}$. In other words, P lies in an affine subspace defined by $A^= x = b^=$.

Exercise 1 *Prove that there is a point $x' \in P$ such that $A^= x' = b^=$ and $A^+ x' < b^+$.*

Definition 1 *The dimension, $\mathbf{dim}(P)$, of a polyhedron P is the maximum number of affinely independent points in P minus 1.*

Notice that by definition of dimension, if $P \subseteq \mathbb{R}^n$ then $\mathbf{dim}(P) \leq n$, if $P = \emptyset$ then $\mathbf{dim}(P) = -1$, and $\mathbf{dim}(P) = 0$ if and only if P consists of a single point. If $\mathbf{dim}(P) = n$ then we say that P is *full-dimensional*.

Exercise 2 *Show that $\mathbf{dim}(P) = n - \mathbf{rank}(A^=)$.*

The previous exercise implies that P is full-dimensional if and only if there are no implicit inequalities in $Ax \leq b$.

Definition 2 $\mathbf{affhull}(P) = \{x \mid A^= x = b^=\}$

Definition 3 $\mathbf{linspace}(P) = \{x \mid Ax = 0\} = \mathbf{charcone}(P) \cap -\mathbf{charcone}(P)$. *In other words, $\mathbf{linspace}(P)$ is the set of all directions c such that there is a line parallel to c fully contained in P.*

Definition 4 *A polyhedron P is* pointed *if and only if $\mathbf{linspace}(P) = \{0\}$, that is $\mathbf{linspace}(P)$ has dimension 0.*

A constraint row in $Ax \leq b$ is *redundant* if removing it does not change the polyhedron. The system $Ax \leq b$ is *irredundant* if no constraint is redundant.

1.2 Faces of Polyhedra

Definition 5 *An inequality $\alpha x \leq \beta$, where $\alpha \neq 0$, is a* valid *inequality for a polyhedron $P = \{x \mid Ax \leq b\}$ if $\alpha x \leq \beta$ $\forall x \in P$. The inequality is a* supporting hyperplane *if it is valid and has a non-empty intersection with P*

Definition 6 *A* face *of a polyhedron P is the intersection of P with $\{x \mid \alpha x = \beta\}$ where $\alpha x \leq \beta$ is a valid inequality for P.*

We are interested in non-empty faces. Notice that a face of a polyhedron is also a polyhedron. A face of P is an *extreme point* or a *vertex* if it has dimension 0. It is a *facet* if the dimension of the face is $\mathbf{dim}(P) - 1$. The face is an *edge* if it has dimension 1.

Another way to define a face is to say that F is a face of P if $F = \{x \in P \mid A'x = b'\}$ where $A'x \leq b'$ is a subset of the inequalities of $Ax \leq b$. In other words, $F = \{x \in P \mid a_i x = b_i, i \in I\}$ where $I \subseteq \{1, 2, \ldots, m\}$ is a subset of the rows of A.

Now we will show that these two definitions are equivalent.

Theorem 7 *Let $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^n$. Let $P = \{x \mid Ax \leq b\}$ be a polyhedron. Let F be a face defined by a valid inequality $\alpha x \leq \beta$. Then $\exists I \subseteq \{1.2. \ldots, m\}$ such that $F = \{x \in P \mid a_i x = b_i, i \in I\}$.*

Proof: Let $F = \{x \mid x \in P, \alpha x = \beta\}$ where $\alpha x \leq \beta$ is a supporting hyperplane. Then, the following claim is easy to see.

Claim 8 *F is the set of all optimal solutions to the LP*

$$\max \alpha x$$
$$Ax \leq b.$$

The above LP has an optimal value β. This implies that the dual LP is feasible and has an optimum solution y^*. Let $I = \{i \mid y_i^* > 0\}$. Let X be the set of all optimal solutions to the primal. For any $x' \in X$, by complimentary slackness for x' and y^*, we have that $y_i^* > 0$ implies $a_i x' = b_i$. Therefore X is a subset of the solutions to the following system of inequalities:

$$\begin{aligned} a_i x &= b_i \qquad i \in I \\ a_i x &\leq b_i \qquad i \notin I \end{aligned}$$

Again, by complementary slackness any x' that satisfies the above is optimal (via y^*) for the primal LP and. Therefore $F = X = \{x \in P \mid a_i x_i = b_i, i \in I\}$. $\square$

Now we consider the converse.

Theorem 9 *Let $P = \{x \mid Ax \leq b\}$ where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Let $I \subseteq \{1, \ldots, m\}$ and $F = \{x \in P \mid a_i x = b_i, i \in I\}$. If F is non-empty, then there is a valid inequality $\alpha x \leq \beta$ such that $F = P \cap \{x \mid \alpha x = \beta\}$.*

Proof:[Idea] Let $\alpha = \sum_{i \in I} a_i$ be a row vector and $\beta = \max\{\alpha x \mid Ax \leq b\}$. We claim that $F = \{x \mid x \in P, \alpha x = \beta\}$ which implies that F is the intersection of P with the supporting hyperplane $\alpha x \leq \beta$. $\square$

Corollary 10

1. *The number of faces of $P = \{x \mid Ax \leq b\}$ where A is a $m \times n$ matrix is at most 2^m.*

2. *Each face is a polyhedron.*

3. *If F is a face of P and $F' \subseteq F$ then F' is a face of P if and only if F' is a face of F.*

4. *The intersecton of two faces is either a face or is empty.*

1.3 Facets

Definition 11 *A* facet *of P is an inclusion-wise maximal face distinct from P. Equivalently, a face F of P is a facet if and only if $\mathbf{dim}(F) = \mathbf{dim}(P) - 1$.*

We have the following theorem about facets.

Theorem 12 *Let $P = \{x \mid Ax \leq b\} = \{x \mid A^{=}x = b^{=}, A^{+}x \leq b^{+}\}$. If no inequality of $A^{+}x \leq b^{+}$ is redundant in $Ax \leq b$, then there is a one to one correspondence between the facets of P and the inequalities in $A^{+}x \leq b^{+}$. That is, F is a facet of P if and only if $F = \{x \in P \mid a_i x = b_i\}$ for some inequality $a_i x \leq b_i$ from $A^{+}x \leq b^{+}$.*

Proof: Let F be a facet of P. Then $F = \{x \in P \mid A'x = b'\}$ where $A'x \leq b'$ is a subsystem of $A^{+}x \leq b^{+}$. Take some inequality $\alpha x \leq \beta$ in $A'x \leq b'$. Then $F' = \{x \in P \mid \alpha x = \beta\}$ is a face of P and $F \subseteq F'$. Moreover, $F' \neq P$ since no inequality in $A^{+} \leq b^{+}$ is an implicit equality.

Let $F = \{x \in P \mid \alpha x = \beta\}$ for some inequality $\alpha x \leq \beta$ from $A^{+} \leq b^{+}$. We claim that $\mathbf{dim}(F) = \mathbf{dim}(P) - 1$ which implies that F is a facet. To prove the claim it is sufficient to show that there is a point $x_0 \in P$ such that $A^{=}x_0 = b^{=}, \alpha x_0 = \beta$ and $A'x_0 < b'$ where $A' \leq b'$ is the inequalities in $A^{+}x \leq b^{+}$ with $\alpha x \leq \beta$ omitted. From Exercise 1, there is a point x_1 such that $\alpha x_1 = \beta$ and $A^{=}x_1 = b^{=}$ and $A^{+}x_1 < b^{+}$. Moreover since $\alpha x \leq \beta$ is irredundat in $Ax \leq b$, there is a point x_2 such that $A^{=}x_2 = b^{=}$ and $A'x_2 \leq b'$ and $\alpha x_2 > \beta$. A convex combination of x_1 and x_2 implies the existence of the desired x_0. $\square$

Corollary 13 *Each face of P is the intersection of some of the facets of P.*

Corollary 14 *A polyhedrom P has no facet if and only if P is an affine subspace.*

Exercise 3 *Prove the above two corollaries using Theorem 12.*

1.4 Minimal Faces and Vertices

A face is inclusion-wise minimal if it does not contain any other face. From Corollary 14 and the fact that a face of a polyhedron is a polyhedron the next proposition follows.

Proposition 15 *A face F of P is minimal if and only if F is an affine subspace.*

Theorem 16 *A set F is minimal face of P if and only if $\emptyset \neq F$, $F \subseteq P$ and $F = \{x \mid A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$.*

Proof: Suppose F is a face and $F = \{x \mid A'x = b'\}$ then by Proposition 15, it is minimal. For the converse direction suppose F is a minimal face of P. Since F is a face, $F = \{x \mid A''x \leq b'', A'x = b'\}$ where $A''x \leq b''$ and $A'x \leq b'$ are two subsystems of $Ax \leq b$. We can assume that $A''x \leq b''$ is as small as possible and therefore, irredundant. From Theorem 12, if $A''x \leq b''$ has any inequality then F has a facet which implies that F is not minimal. Therefore, $F = \{x \mid A'x = b'\}$. $\square$

Exercise 4 *Prove that all minimal faces of a polyhedron $\{x \mid Ax \leq b\}$ are of the form $\{x \mid A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$ where $rank(A') = rank(A)$. Conclude that all minimal faces are translates of* **linspace**(P) *and have the same dimension.*

A *vertex* or an *extreme point* of P is a (minimal) face of dimension 0. That is, a single point. A polyhedron is *pointed* if and only if it has a vertex. Note that since all minimal faces have the same dimension, if P has a vertex than all minimal faces are vertices. Since a minimal face F of P is defined by $A'x = b'$ for some subsystem $A'x \leq b'$ of $Ax \leq b$, if a vertex of P is the *unique solution* to $A'x = b'$ then $\mathbf{rank}(A') = n$. We can then assume that A' has n rows. Vertices are also called *basic feasible solutions.*

Corollary 17 *A polyhedron $\{x \mid Ax \leq b\}$ has a vertex only if A has rank n. The polyhedron $\{x \mid Ax \leq b\}$ is pointed if it is not empty.*

1.5 Decomposition of Polyhedra

Recall that we had earlier stated that,

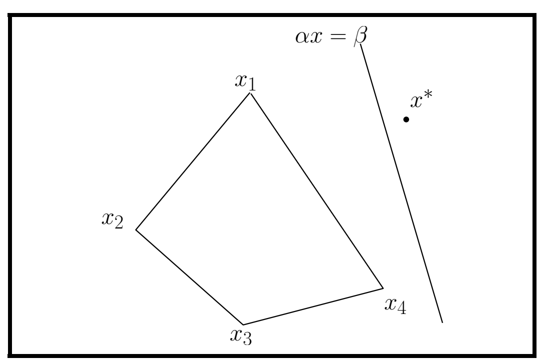
Theorem 18 *Any polyhedron P can be written as $Q + C$ where Q is a convex hull of a finites set of vectors and $C = \{x \mid Ax \leq 0\}$ is the* **charcone** *of P.*

We can give more details of the decomposition now. Given P, let $F_1, F_2, \ldots, F_h$ be its minimal faces. Choose $x_i \in F_i$ arbitrarily. Then $P = \mathbf{convexhull}(x_1, x_2, \ldots, x_h) + C$. In particular, if P is pointed then $x_1, x_2, \ldots, x_h$ are vertices of P and hence $P = \mathbf{convexhull}(\text{vertices}(P)) + C$.

We will prove the above for polytopes.

Theorem 19 *A polytope (bounded polyhedron) is the convex hull of its vertices*

Proof: First observe that a bounded polyhedron is necessarily pointed. Let $X = \{x_1, x_2, \ldots, x_h\}$ be the vertices of P. Clearly $\mathbf{convexhull}(X) \subseteq P$. We prove the converse. Suppose $x^* \in P$ does no belong to $\mathbf{convexhull}(X)$.



Claim 20 *There exists a hyperplane $\alpha x = \beta$ such that $\alpha x_i < \beta \ \forall x_i \in X$ and $\alpha x^* \geq \beta$.*

Proof:[Claim] One can prove this by using Farkas lemma (see [2] for example) or appeal to the general theorem that if two convex sets do not intersect then there is a separating hyperplane for them; in particular if one of the sets is bounded then there is a *strict* separating hyperplane (see [1], Section 2.4). A sketch of this for the restricted case we have is as follows. Let $y \in \textbf{convexhull}(X)$ minimize the distance from x^* to $\textbf{convexhull}(X)$. The claim is that a hyperplane that is normal to the line segment joining x^* and y and passing through x^* is the desired hyperplane. Otherwise $\textbf{convexhull}(X)$ intersects this hyperplane, and let y' be a point in the intersection. Since $\textbf{convexhull}(X)$ is convex, the line segment joining y' and y is contained in $\textbf{convexhull}(X)$. Now consider the right angled triangle formed by y, x^*, y'. From elementary geometry, it follows that there is a point closer to x^* than y on the line segment joining y and y', contradicting the choice of y. □

Now consider $\max\{\alpha x \mid x \in P\}$. The set of optimal solutions to this LP is a face of F. By Claim 20, $X \cap F = \emptyset$. Since F is a face of P, it has a vertex of P since P is pointed. This contradicts that X is the set of all vertices of P. □

One consequence of the decomposition theorem is the following.

Theorem 21 *If $P = \{x \mid Ax \leq b\}$ is pointed then for any $c \neq 0$ the LP*

$$\max cx$$
$$Ax \leq b$$

is either unbounded, or there is a vertex x^* such that x^* is an optimal solution.

The proof of the previous theorem is left as an exercise.

2 Complexity of Linear Programming

Recall that LP is an optimization problem of the following form.

$$\max \alpha x$$
$$Ax \leq b$$

As a computational problem we assume that the inputs c, A, b are rational. Thus the input consists of $n + m \times n + n$ rational numbers. Given an instance I we use $\textbf{size}(I)$ to denote the number of bits in the binary representation of I. We use it loosely for other quantities such as numbers, matrices, etc. We have that $\textbf{size}(I)$ for an LP instance is,

$$\textbf{size}(c) + \textbf{size}(A) + \textbf{size}(b) \leq (m \times n + 2n)\textbf{size}(L)$$

where L is the largest number in c, A, b.

Lemma 22 *Given a $n \times n$ rational matrix* $\textbf{size}(\det(A)) = \textbf{poly}(\textbf{size}(A))$.

Proof: For simplicity assume that A has integer entries, otherwise one can multiply each entry by the lcm of the denominators of the rational entries. We have

$$\det(A) = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)}$$

where S_n is the set of all permutations on $\{1, \ldots, n\}$. Hence

$$\begin{aligned} |\det(A)| &\leq \sum_{\sigma \in S_n} \prod_{i=1}^{n} |A_{i,\sigma(i)}| \\ &\leq n! \times L^n \end{aligned}$$

where $L = \max |A_{i,j}|$; here $|x|$ for a number x is its absolute value. Therefore the number of bits required to represent $\det(A)$ is $O(n \log L + n \log n)$ which is polynomial in n and $\log L$, and hence $\mathbf{poly}(\mathbf{size}(A))$. $\square$

Corollary 23 *If A has an inverse, then* $\mathbf{size}(A^{-1}) = \mathbf{poly}(\mathbf{size}(A))$.

Corollary 24 *If $Ax = b$ has a feasible solution then there exists a solution x^* such that* $\mathbf{size}(x^*) = \mathbf{poly}(\mathbf{size}(A, b))$.

Proof: Suppose $Ax = b$ has a feasible solution. By basic linear algbegra, there is a square submatrix U of A with full rank and a sub-vector b' such that $U^{-1}b'$ padded by 0's for the other variables is a feasible solution for the original system. We then apply the previous corollary to U and b'. $\square$

Gaussian elimination can be adapted using the above to show the following — see [3].

Theorem 25 *There is a polynomial time algorith, that given a linear system $Ax = b$, either correctly outputs that it has no feasible solution or outputs a feasible solution. Moreover, the algorithm determines whether A has a unique feasible solution.*

Now we consider the case when $Ax \leq b$ has a feasible solution.

Theorem 26 *If a linear system $Ax \leq b$ has a feasible solution then there exists a solution x^* such that* $\mathbf{size}(x^*) = \mathbf{poly}(\mathbf{size}(A, b))$.

Proof: Consider a minimal face F of $P = \{x \mid Ax \leq b\}$. We have seen that $F = \{x \mid A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$. By Theorem 25, $A'x = b'$ has a solution of size $\mathbf{poly}(\mathbf{size}(A', b'))$. $\square$

Corollary 27 *The problem of deciding whether $\{x \mid Ax \leq b\}$ is non-empty is in* **NP**.

Corollary 28 *The problem of deciding whether $\{x \mid Ax \leq b\}$ is* empty *is in* **NP**. *Equivalently the problem of deciding non-emptiness is in* **coNP**.

Proof: By Farkas' lemma, if $Ax \leq b$ is empty only if $\exists y \geq 0$ such that $yA = 0$ and $yb = -1$. Therefore there is a certificate y the problem of deciding whether $\{x \mid Ax \leq b\}$ is empty. Further, by Lemma 22 this certificate has polynomial size. $\square$

Thus we have seen that deciding whether $Ax \leq b$ is non-empty is in $\mathbf{NP} \cap \mathbf{coNP}$.

Now consider the optimization problem.

$$\begin{gathered} \max \alpha x \\ Ax \leq b \end{gathered}$$

A natural decision problem associated with the above problem is to decide if the optimum value is at least some given rational number α.

Exercise 5 *Prove that the above decision problem is in* $\mathbf{NP} \cap \mathbf{coNP}$.

Another useful fact is the following.

Theorem 29 *If the optimum value of the LP* $\max cx$ *such that* $Ax \leq b$ *is finite then the optimum value has size polynomial in the input size.*

Proof:[sketch] If the optimum value is finite then by strong duality then it is achieved by a solution (x', y') that satisfies the following system:

$$cx = yb$$

$$Ax \leq b$$

$$yA = c$$

$$y \geq 0.$$

From Theorem 26, there is a solution (x^*, y^*) to the above system with $\mathbf{size}(x^*, y^*)$ polynomial in $\mathbf{size}(A, b, c)$. Hence the optimum value which is cx^* has size polynomial in $\mathbf{size}(A, b, c)$. $\square$

Exercise 6 *Show that the decision problem of deciding whether* $\max cx$ *where* $Ax \leq b$ *is unbounded is in* $\mathbf{NP} \cap \mathbf{coNP}$.

The optimization problem for

$$\max \alpha x$$
$$Ax \leq b$$

requires an algorithm that correctly outputs one of the following

1. $Ax \leq b$ is infeasible
2. the optimal value is unbounded
3. a solution x^* such that cx^* is the optimum value

A related search problem is given $Ax \leq b$ either output that $Ax \leq b$ is infeasible or a solution x^* such that $Ax^* \leq b$.

Exercise 7 *Prove that the above two search problems are polynomial time equivalent.*

3 Polynomial-time Algorithms for LP

Khachiyan's ellipsoid algorithm in 1978 was the first polynomial-time algorithm for LP. Although an impractical algorithm, it had (and contiues to have) a major theoretical impact. The algorithm shows that one does not need the full system $Ax \leq b$ in advance. If one examines carefully the size of a proof of feasibility of a system of inequalities $Ax \leq b$, one notices that there is a solution x^* such that x^* is a solution to $A'x \leq b'$ for some subsystem $A'x \leq b'$ where rank of A is at most n. This implies that A' can be chosen to have at most n rows. Therefore, if the system has a solution then there is one whose size is polynomial in n and the size of the largest entry in A. We may discuss more details of the ellipsoid method in a later lecture.

Subsequently, Karmarkar in 1984 gave another polynomial-time algorithm using an interior point method. This is much more useful in practice, especially for certain large linear programs, and can beat the simplex method which is the dominant method in practice although it is not a polynomial time algorithm in the worst case.

References

[1] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004. Available electronically at `http://www.stanford.edu/~boyd/cvxbook/`.

[2] M. Goemans. Lecture notes on linear programming and polyhedral combinatorics. `http://www-math.mit.edu/~goemans/18433S09/polyhedral.pdf`, 2009.

[3] A. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, Chapters 7, 8, 1998.

1 Integer Programming, Integer Polyhedra, and Totally Unimodular Matrices

Many discrete optimization problems are naturally modeled as an integer (linear) programming (ILP) problem. An ILP problem is of the form

$$\begin{aligned} &\max cx \\ &Ax \leq b \\ &x \text{ is an integer vector.} \end{aligned} \tag{1}$$

It is easy to show that ILP is NP-hard via a reduction from say SAT. The decision version of ILP is the following: Given rational matrix A and rational vector b, does $Ax \leq b$ have an integral solution x?

Theorem 1 *Decision version of ILP is in NP, and hence it is NP-Complete.*

The above theorem requires technical work to show that if there is an integer vector in $Ax \leq b$ then there is one whose size is polynomial in $size(A, b)$.

A special case of interest is when the number of variables, n, is a fixed constant but the number of constraints, m, is part of the input. The following theorem is known.

Theorem 2 (Lenstra's Algorithm) *For each fixed n, there is a polynomial time algorithm for ILP in n variables.*

1.1 Integer Polyhedra

Given a rational polyhedron $P = \{x | Ax \leq b\}$, we use P_I to denote the convex hull of all the integer vectors in P; this is called the integer hull of P.

It is easy to see that if P is a polytope then P_I is also a polytope. Somewhat more involved is the following.

Theorem 3 *For any rational polyhedron P, P_I is also a polyhedron.*

Definition 4 *A rational polyhedron P is an integer polyhedron if and only if $P = P_I$.*

Theorem 5 *The following are equivalent:*

(i) $P = P_I$ *i.e., P is integer polyhedron.*

(ii) Every face of P has an integer vector.

(iii) Every minimal face of P has an integer vector.

(iv) $\max\{cx|x \in P\}$ *is attained by an integer vector when the optimum value is finite.*

Proof: (i)$\Longrightarrow$(ii): Let F be a face, then $F = P \cap H$, where H is a supporting hyperplane, and let $x \in F$. From $P = P_I$, x is a convex combination of integral points in P, which must belong to H and thus to F.
(ii)$\Longrightarrow$(iii): it is direct from (ii).
(iii)$\Longrightarrow$(iv): Let $\delta = \max\{cx : x \in P\} < +\infty$, then $F = \{x \in P : cx = \delta\}$ is a face of P, which has an integer vector from (iii).
(iv)$\Longrightarrow$(i): Suppose there is a vector $y \in P \setminus P_I$. Then there is an inequality $\alpha x \leq \beta$ valid for P_I while $\alpha y > \beta$ (a hyperplane separating y and P_I). It follows that $\max\{\alpha x \mid x \in P_I\} \leq \beta$ while $\max\{\alpha x \mid x \in P\} > \beta$ since $y \in P \setminus P_I$. Then (iv) is violated for $c = \alpha$. $\square$

Another useful theorem that characterizes integral polyhedra, in full generality due to Edmons and Giles [1977], is the following.

Theorem 6 *A rational polyhedron P is integral if and only if* $\max\{cx \mid Ax \leq b\}$ *is an integer for each integral vector c for which the maximum is finite.*

1.2 Totally Unimodular Matrices

Totally Unimodular Matrices give rise to integer polyhedra with several fundamental applications in combinatorial optimization.

Definition 7 *A matrix A is totally unimodular (TUM) if the determinant of each square submatrix of A is in $\{0, 1, -1\}$. In particular, each entry of A is in $\{0, 1, -1\}$.*

Proposition 8 *If A is TUM and U is a non-singular square submatrix of A, then U^{-1} is an integral matrix.*

Proof: $U^{-1} = \dfrac{U^*}{det(U)}$ where U^* is the adjoint matrix of U. From the definition of total unimodularity, U^* only contains entries in $\{0, +1, -1\}$ and $det(U) = 1$ or -1. Therefore, U is an integral matrix. $\square$

Theorem 9 *If A is TUM then for all integral b, the polyhedron $P = \{x|Ax \leq b\}$ is an integer polyhedron.*

Proof: Consider any minimal face F of P. $F = \{x|A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$. Let A' have $m' \leq n$ rows. Then $A' = [U \quad V]$, where U is a $m' \times m'$ matrix of full row and column rank (after potentially rearranging rows and columns). U is a submatrix of A and hence $det(U) \in \{-1, +1\}$. Therefore $A'x = b'$ has an integer solution $\begin{pmatrix} U^{-1}b' \\ 0 \end{pmatrix}$. Thus every minimal face has an integer vector and hence P is an integer polyhedron. $\square$

We claim several important corollaries.

Corollary 10 *If A is TUM then for all integral vector a, b, c, d, the polyhedron $\{x|a \leq x \leq b, c \leq Ax \leq d\}$ is integral.*

Proof: If A is TUM, so is the matrix $\begin{bmatrix} I \\ -I \\ A \\ -A \end{bmatrix}$. This can be easily proven by expanding the submatrix along the row associated with the identity matrix. $\square$

Proposition 11 *A is TUM $\iff$ A^T is TUM.*

Corollary 12 *If A is TUM and b, c are integral vectors, then $\max\{cx | Ax \leq b, x \geq 0\} = \min\{yb | yA \leq c, y \geq 0\}$ are attained by integral vectors x^* and y^*, if they are finite.*

Proof: The polyhedron $\{y | y \geq 0, yA \leq c\}$ is integral since A^T is TUM and also $\begin{bmatrix} A^T \\ -I \end{bmatrix}$. $\square$

There are many characterizations of TUM matrices. We give a few useful ones below. See [1] (Chapter 19) for a proof.

Theorem 13 *Let A be a matrix with entries in $\{0, +1, -1\}$. Then the followings are equivalent.*

(i) A is TUM.

(ii) For all integral vector b, $\{x | Ax \leq b, x \geq 0\}$ has only integral vertices.

(iii) For all integral vectors a, b, c, d, $\{x | a \leq x \leq b, c \leq Ax \leq d\}$ has only integral vertices.

(iv) Each collection of column S of A can be split into two sets S_1 and S_2 such that the sum of columns in S_1 minus the sum of columns in S_2 is a vector with entries in $\{0, +1, -1\}$.

(v) Each nonsingular submatrix of A has a row with an odd number of nonzero components.

(vi) No square submatrix of A has determinant $+2$ or -2.

$(i) \iff (ii)$ is the Hoffman-Kruskal's theorem. $(ii) \implies (iii)$ follows from the fact that A is TUM $\implies \begin{bmatrix} I \\ -I \\ A \\ -A \end{bmatrix}$ is TUM. $(i) \iff (iv)$ is Ghouila-Houri's theorem.

Several important matrices that arise in combinatorial optimization are TUM.

Example 1: Bipartite Graphs. Let $G = (V, E)$ an undirected graph. Let M be the $\{0, 1\}$ edge-vetex incidence matrix defined as follows. M has $|E|$ rows, one for each edge and $|V|$ columns, one for each vertex. $M_{e,v} = 1$ if e is incident to v otherwise it is 0. The claim is that M is TUM iff G is bipartite.

To see bipartiteness is needed, consider the matrix $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ for a triangle which is an odd cycle. Its determinant is 2.

Exercise 1 *Show that edge-vertex adjacency matrix of any odd cycle has determinant 2.*

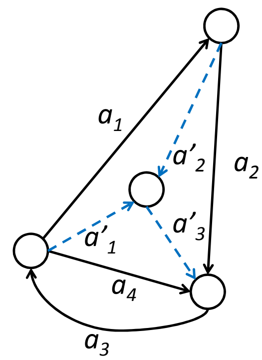


Figure 1: Network matrix is defined by a directed tree (dotted edges) and a directed graph on the same vertex set.

Example 2: Directed Graphs. Let $D = (V, A)$ be a directed graph. Let M be an $|E| \times |V|$ arc-vertex adjacency matrix defined as

$$M_{a,v} = \begin{cases} 0, \text{ if } a \text{ is not incident to } v \\ +1, \text{ if } a \text{ enters } v \\ -1, \text{ if } a \text{ leaves } v \end{cases} \tag{2}$$

M is TUM. This was first observed by Poincaré [1900].

Example 3: Consecutive 1's: A is a consecutive 1's matrix if it is a matrix with entries in $\{0, 1\}$ such that in each row the 1's are in a consecutive block. This naturally arises as an incidence matrix of a collection of intervals and a set of points on the real line.

The above three claims of matrices are special cases of network matrices (due to Tutte).

Definition 14 *A network matrix is defined from a directed graph $D = (V, A)$ and a directed tree $T = (V, A')$ on the same vertex set V. The matrix M is $|A'| \times |A|$ matrix such that for $a = (u, v) \in A$ and $a' \in A'$*

$$M_{a,a'} = \begin{cases} 0, \ \textit{if the unique path from } u \to v \textit{ in } T \textit{ does not contain } a' \\ +1, \ \textit{if the unique path from } u \to v \textit{ in } T \textit{ passes through } a' \textit{ in forward direction} \\ -1, \ \textit{if the unique path} u \to v \textit{ in } T \textit{ passes through } a' \textit{ in backward direction} \end{cases}$$

The network matrix corresponding to the directed graph and the tree in Figure 1 is given below. The dotted edge is T, and the solid edge is D.

$$M = \begin{array}{c} \\ a'_1 \\ a'_2 \\ a'_3 \end{array} \begin{array}{c} \begin{array}{cccc} a_1 & a_2 & a_3 & a_4 \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 1 \\ -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 1 \end{bmatrix} \end{array}$$

Theorem 15 *(Tutte) Every network matrix is TUM.*

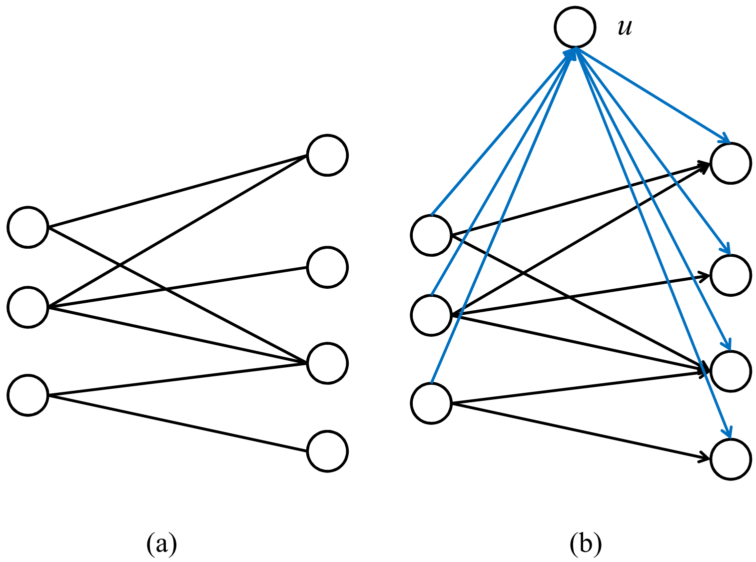


Figure 2:

We will prove this later. First we show that the previous examples can be cast as special cases of network matrices.

Bipartite graphs. Say $G = \{X \cup Y, E\}$ as in Figure 2(a). One can see that edge-vertex adjacency matrix of G as the network matrix induced by a directed graph $G = (X \cup Y \cup \{u\}, A)$ where u is a new vertex and A is the set of arcs defined by orientating the edges of G from X to Y. $T = (X \cup Y \cup \{u\}, A')$ where $A' = \{(v, u)|v \in X\} \cup \{(u, v)|v \in Y\}$ as in Figure 2(b).

Directed graphs. Suppose $D = (V, A)$ is a directed graph. Consider the network matrix induced by $D = (V \cup \{u\}, A)$ and $T = (V \cup \{u\}, A')$ where u is a new vertex and where $A' = \{(v, u)|v \in V\}$.

Consecutive 1's matrix. Let A be a consecutive 1's matrix with m rows and n columns. Assume for simplicity that each row has at least one 1 and let ℓ_i and r_i be the left most and right most columns of the consecutive block of 1's in row i. Let $V = \{1, 2, \ldots, n\}$. Consider $T = (V, A')$ where $A' = \{(i, i+1) \mid 1 \leq i < n\}$ and $D = (V, A)$ where $A = \{(\ell_i, r_i) \mid 1 \leq i \leq n\}$. It is easy to see that A is the network matrix defined by T and A.

Now we prove that every network matrix is TUM. We need a preliminary lemma.

Lemma 16 *Every submatrix M' of a network matrix M is also a network matrix.*

Proof: If M is a network matrix, defined by $D = (V, A)$ and $T = (V, A')$, then removing a column in M corresponds to removing an arc $a \in A$. Removing a row corresponds to identifying/contracting the end points of an arc a' in T. $\square$

Proposition 17 *A is TUM $\Longleftrightarrow$ A' obtained by multiplying any row or column by -1 is TUM.*

Corollary 18 *If M is a network matrix, M is TUM $\Longleftrightarrow$ M' is TUM where M' is obtained by reversing an arc of either T or D.*

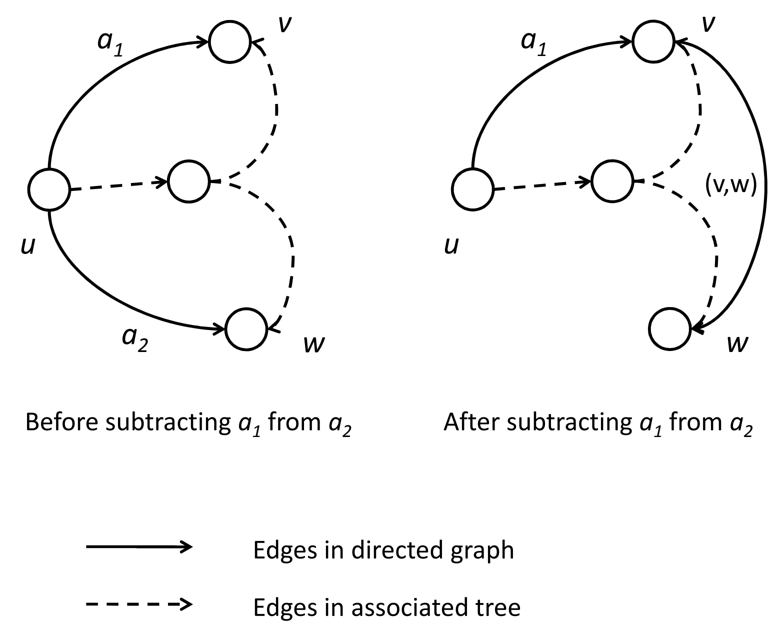


Figure 3:

Proof of Theorem 15. By Lemma 16, it suffices to show that any square network matrix C has determinant in $\{0, +1, -1\}$. Let C be a $k \times k$ network matrix defined by $D = (V, A)$ and $T = (V, A')$. We prove by induction on k that $det(C) \in \{0, 1, -1\}$. Base case with $k = 1$ is trivial since entries of C are in $\{0, 1, -1\}$.

Let $a' \in A'$ be an arc incident to a leaf u in T. By reorienting the arcs of T, we will assume that a' leaves u and moreover all arcs A incident to u leave u (see Corollary 18).

Let $a_1, a_2, \cdots, a_h$ be arcs in A leaving u (If no arcs are incident to u then $det(C) = 0$). Assume without loss of generality that a' is the first row of C and that $a_1, a_2, \cdots, a_h$ are the first h columns of C.

Claim 19 *Let C' be obtained by subtracting column a_1 from column a_2. C' is the network matrix for $T = (V, A')$ and $D = (V, A - a_2 + (v, w))$ where $a_1 = (u, v)$ and $a_2 = (u, w)$.*

We leave the proof of the above as an exercise — see Figure 3.

Let C'' be the matrix obtained by subtracting column of a_1 from each of $a_2, \cdots, a_h$. From the above claim, it is also a network matrix. Moreover, $det(C'') = det(C)$ since determinant is preserved by these operations. Now C'' has 1 in the first row in column one (corresponding to a_1) and 0's in all other columns. Therefore, $det(C'') \in \{0, +1, -1\}$ by expanding along the first row and using induction for the submatrix of C'' consisting of columns 2 to k and rows 2 to k. □

Some natural questions on TUM matrices are the following.

(i) Are there TUM matrices that are not a network matrix (or its transpose)?

(ii) Given a matrix A, can one check efficiently whether it is a TUM matrix?

The answer to (i) is negative as shown by the following two matrices given by Hoffman[1960]

$$\begin{bmatrix} 1 & -1 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 1 \end{bmatrix}$$

and Bixby[1977].

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Amazingly, in some sense, these are the only two exceptions.

Seymour, in a deep and difficult technical theorem, showed via matroid theory methods that any TUM matrix can be obtained by "gluing" together network matrices and the above two matrices via some standard operations that preserve total unimodularity. His descomposition theorem also led to a polynomial time algorithm for checking if a given matrix is TUM. There was an earlier polynomial time algorithm to check if a given matrix is a network matrix. See [1] (Chapters 20 and 21) for details.

References

[1] A. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, 1998.

CS 598CSC: Combinatorial Optimization	Lecture date: Feb 2, 2010
Instructor: Chandra Chekuri	Scribe: Siva Theja Maguluri

Material taken mostly from [1] (Chapter 19).

1 Integer Decomposition Property

A polyhedron P has the *integer decomposition property* if $\forall$ integers $k \geq 1$ and $x \in P$, kx is integral implies $kx = x_1 + x_2 + \ldots + x_k$ for integral vectors $x_1, \ldots, x_k$ in P. Baum and Trotter showed the following:

Theorem 1 (Baum and Trotter) *A matrix A is TUM iff $P = \{x \mid x \geq 0, Ax \leq b\}$ has the integer decomposition property for all integral verctors b.*

Proof: We show one direction, the one useful for applications. Suppose A is TUM, consider $P = \{x \mid x \geq 0, Ax \leq b\}$. Let $y = kx^*$ be an integral vector where $x^* \in P$. We prove by induction on k that $y = x_1 + x_2 + \ldots .x_k$ for integral vectors $x_1, x_2, \ldots, x_k$ in P.

Base case for $k = 1$ is trivial.

For $k \geq 2$, consider the polyhedron $P' = \{x \mid 0 \leq x \leq y; Ay - kb + b \leq Ax \leq b\}$. P' is an integral polyhedron since A is TUM and $Ay - kb + b$ and b are integral. The vector $x^* \in P'$ and hence P' is not empty. Hence there is an integral vector $x_1 \in P'$. Moreover $y' = y - x_1$ is integral and $y' \geq 0$, $Ay' \leq (k-1)b$.

By induction $y' = x_2 + \ldots + x_{k-1}$ where $x_2, \ldots, x_{k-1}$ are integral vectors in P. $y = x_1 + \ldots + x_k$ is the desired combination for y. $\Box$

Remark 2 *A polyhedron P may have the integer decomposition property even if the constraint matrix A is not TUM. The point about TUM matrices is that the property holds for all integral right hand side vectors b.*

2 Applications of TUM Matrices

We saw that network matrices are TUM and that some matrices arising from graphs are network matrices. TUM matrices give rise to integral polyhedra, and in particular, simultaneously to the primal and dual in the following when A is TUM and c, b are integral vectors.

$$\max\{cx \mid x \geq 0, Ax \leq b\} = \min\{yb \mid y \geq 0, yA \geq c\}$$

We can derive some min-max results and algorithms as a consequence.

2.1 Bipartite Graph Matchings

Let $G = (V, E)$ be a bipartite graph with $V = V_1 \oplus V_2$ as the bipartition. We can write an integer program for the maximum cardinality matching problem as

$$\begin{array}{rcl} \max & & \displaystyle\sum_{e \in E} x(e) \\ x(\delta(u)) & \leq & 1 \ \forall u \in V \\ x(e) & \geq & 0 \ \forall e \in E \\ x & \in & \mathbb{Z} \end{array}$$

We observe that this is a ILP problem $\max\{1 \cdot x \mid Mx \leq 1, x \geq 0, x \in \mathbb{Z}\}$ where M is the edge-vertex incidence matrix of G. Since M is TUM, we can drop the integrality constraint and solve the *linear program* $\max\{1 \cdot x \mid Mx \leq 1, x \geq 0\}$ since $\{x \mid Mx \leq 1, x \geq 0\}$ is an integral polyhedron. The dual of the above LP is

$$\begin{array}{rcll} \min \displaystyle\sum_{u \in V} y(u) & & & \\ y(u) + y(v) & \geq & 1 & uv \in E \\ y & \geq & 0 & \end{array}$$

in other words $\min\{y \cdot 1 | yM \geq 1, y \geq 0\}$ which is is also an integral polyhedron since M^T is TUM. We note that this is the min-cardinality vertex cover problem

Note that the primal LP is a polytope and hence has a finite optimum solution. By duality, and integrality of the polyhedra, we get that both primal and dual have integral optimum solutions x^* and y^* such that $1 \cdot x^* = y^* \cdot 1$. We get as an immediate corollary König's Theorem.

Theorem 3 *In a bipartite graph the cardinality of a maximum matching is equal to the cardinality of a minimum vertex cover.*

Also, by poly-time solvability of linear programming, there is a polynomial time algorithm for maximum matching and minimum vertex cover in bipartite graphs, and also their weighted versions. Note that we have much more efficient combinatorial algorithms for these problems. We also obtain that $\{x \mid Mx \leq 1, x \geq 0\}$ is the convex hull of the characteristic vectors of the matchings in G and that $\{x \mid Mx = 1, x \geq 0\}$ is the convex hull of the perfect matchings of G.

One easy consequence is the following theorem

Theorem 4 (Birkhoff - Von Neumann) *Let A be a $n \times n$ doubly stochastic matrix. Then A can be written as a convex combination of permutation matrices.*

A doubly stochastic matrix is a square non-negative matrix in which each row and column sum is 1. A permutation matrix is a square $\{0, 1\}$ matrix that has a single 1 in each row and column. Each permulation matrix corresponds to a permutation σ in S_n, the set of all permutations on an n-element set.

Exercise 5 *Prove the above theorem using the perfect matching polytope description for bipartite graphs. How many permutations matrices do you need in the convex combination?*

Note that $\max\{wx \mid Mx \leq 1, x \geq 0\} = \min\{y \cdot 1 \mid yM \geq w, y \geq 0\}$ has integer primal and dual solutions x^* and $y^* \cdot 1$ if w is integral. For the primal x^* corresponds to a maximum w-weight matching. In the dual, via complementary slackness, we have

$$y^*(u) + y^*(v) = w^*(v)$$

for all $x^*(uv) > 0$ Interpreting $y^*(u)$ as a weight on u, one obtains a generalization of König's theorem, known as the Egervary theorem.

Another theorem on bipartite graphs is the Hall's marriage theorem.

Theorem 6 (Hall) *Let $G = (V, E)$ be a bipartite graph with X, Y as the vertex sets of the bipartition. Then there is a matching that saturates X iff $|N(S)| \geq |S| \forall S \subseteq X$ where $N(S)$ is the set of neighbors of S.*

Exercise 7 *Derive Hall's theorem from König's Theorem.*

A generalization of the above also holds.

Theorem 8 *Let $G = (X \cup Y, E)$ be a bipartite graph. Let $R \subseteq U$. Then there is a matching that covers R iff there exists a matching M that covers $R \cap X$ and a matching that covers $R \cap Y$. Therefore, a matching covers R iff $|N(S)| \geq |S| \forall S \subseteq R \cap X$ and $\forall S \subseteq R \cap Y$.*

Exercise 9 *Prove above theorem.*

b-matchings: b-matchings generalize matchings. Given an integral vector $b : V \rightarrow \mathbb{Z}^+$, a b-matching is a set of edges such that the number of edges incident to a vertex v is at most $b(v)$. From the fact that the matrix M is TUM, one can obtain various properties of b-matchings by observing that the polyhedron

$$\begin{aligned} Mx &\leq b \\ x &\geq 0 \end{aligned}$$

is integral for integral b.

2.2 Single Commodity Flows and Cuts

We can derive various useful and known facts about single commodity flows and cuts using the fact that the directed graph arc-vertex incidence matrix is TUM.

Consider the s-t maximum-flow problem in a directed graph $D = (V, A)$ with capacities $c : A \rightarrow \mathbb{R}^+$. We can express the maximum flow problem as an LP with variables $x(a)$ for flow on arc a.

$$\begin{array}{rcll} \max \displaystyle\sum_{a \in \delta^+(s)} x(a) - \sum_{a \in \delta^-(s)} x(a) & & & \\ \displaystyle\sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) & = & 0 & \forall v \in V - \{s, t\} \\ x(a) & \leq & c(a) & \forall a \in A \\ x(a) & \geq & 0 & \forall a \in A \end{array}$$

Note that the polyhedron defined by the above is of the form $\{x \mid M'x = 0, 0 \leq x \leq c\}$ where M' is the arc-vertex incidence matrix of D with the columns corresponding to s, t removed. M' is a submatrix of M, the arc-vertex incidence matrix of D which is TUM, and hence M' is also TUM. Therefore, the polyhedron above is integral for integral c. One immediate corollary is that for integral capacities, there is a maximum flow that is integral. We now derive the maxflow-mincut theorem as a consequence of the total unimodularity of M.

The dual to the maximum-flow LP above has two sets of variables. $y(a), a \in A$ for the capacity constraints and $z(v), v \in V - \{s,t\}$ for the flow conservation constraints. We let $w(a)$ be the weight vector of the primal. Note that

$$w(a) = \begin{cases} 1 & \text{if } a = (s,v) \text{ for some } v \in V \\ -1 & \text{if } a = (v,s) \text{ for some } v \in V \\ 0 & \text{otherwise} \end{cases}$$

For simplicity assume that there is no arc (s,t) or (t,s). Then the dual is:

$$\begin{array}{rcll} \min \sum_{a \in A} c(a)y(a) & & & \\ z(u) - z(v) + y(u,v) & \geq & 0 & (u,v) \in A \quad \{u,v\} \cap \{s,t\} = \emptyset \\ z(v) + y(s,v) & \geq & 1 & \forall (s,v) \in A \\ z(v) + y(s,v) & \geq & -1 & \forall (v,s) \in A \\ z(v) + y(v,t) & \geq & 0 & \forall (v,t) \in A \\ -z(v) + y(t,v) & \geq & 0 & \forall (t,v) \in A \\ y & \geq & 0 & \end{array}$$

Note that z are unconstrained variables. In matrix form, the primal is $\max\{wx \mid M'x = 0, 0 \leq x \leq c\}$ and the dual is $\min\{yc \mid y \geq 0; \exists z : y + zM' \geq w^T\}$. Since w is integral and M' is TUM, dual is an integral polyhedron. Primal is bounded polyhedron and hence primal and dual have optimal solution x^* and (y^*, z^*) such that $wx^* = y^*c$ and y^*, z^* is integral.

We can extend z to have variables $z(s)$ and $z(t)$ with $z(s) = -1$ and $z(t) = 0$. Then the dual has a cleaner form, $\max\{yc \mid y \geq 0, \exists z : y + zM \geq 0\}$. Note that M here is the full arc-vertex incidence matrix of D. Thus we have x^* and integral (y^*, z^*) such that $wx^* = y^*c$ and $y^* + z^*M \geq 0$.

Let $U = \{v \in V | z^*(v) < 0\}$. Note that $s \in U$ and $t \notin U$ and hence $\delta^+(U)$ is a s-t cut.

Claim 10 $c(\delta^+(U)) \leq y^*c = \sum_{a \in A} y^*(a)c(a)$

Proof: Take any arc $(u,v) \in \delta^+(U)$. We have $z^*(u) - z^*(v) + y^*(u,v) \geq 0$ for each (u,v). Since $u \in U$ and $v \notin U$, $z^*(u) < 0$ and $z^*(v) \geq 0$. Since z^* is integral, we have

$$\begin{array}{rcl} y^*(u,v) & \geq & 1 \\ \Longrightarrow c(\delta^+(U)) & \leq & \sum_{a \in \delta^+(U)} c(a)y^*(a) \\ & \leq & \sum_{a \in A} c(a)y^*(a) \text{ since } y^* \geq 0 \end{array}$$

$\square$

Therefore, U is a s-t cut of capacity at most $y^*c = wx^*$ but wx^* is the value of a maximum flow. Since the capacity of any cut upper bounds the maximum flow, we have that there exists a cut of capacity equal to that of the maximum flow. We therefore, get the following theorem,

Theorem 11 *In any directed graph $G = (V, A)$ with non-negative arc capacities, $c : E \rightarrow \mathbb{Q}^+$, the s-t maximum-flow value is equal to the s-t minimum cut capacity. Moreover, if $c : E \rightarrow F^+$, then there is an integral maximum flow.*

Interpretation of the dual values: A natural interpretation of the dual is the following. The dual values, $y(a)$ indicate whether a is cut or not. The value $z(v)$ is the shortest path distance from s to v with $y(a)$ values as the length on the arcs. We want to separate s from t. So, we have (implicitly) $z(s) = -1$ and $z(t) = 0$. The constraints $z(u) - z(v) + y(u, v) \geq 0$ enforce that the z values are indeed shortest path distances. The objective function $\sum_{a \in A} c(a)y(a)$ is the capacity of the cut subject to separating s from t.

Circulations and lower and upper bounds on arcs: More general applications of flows are obtained by considering both lower and upper bounds on the flow on arcs. In these settings, circulations are more convenient and natural.

Definition 12 *For a directed graph $D = (V, A)$, a circulation is a function $f : A \rightarrow \mathbb{R}^+$ such that $\sum_{a \in \delta^-(v)} f(a) = \sum_{a \in \delta^+(v)} f(a) \forall v \in V$*

Given non-negative lower and upper bounds on the arcs, $l : A \rightarrow \mathbb{R}^+$ and $u : A \rightarrow \mathbb{R}^+$, we are interested in circulations that satisfy the bounds on the arcs. In other words, the feasibility of the following:

$$\begin{gathered} l(a) \leq x(a) \leq u(a) \\ x \text{ is a circulation} \end{gathered}$$

The above polyhedron is same as $\{x \mid Mx = 0, l \leq x \leq u\}$ where M is the arc-vertex incidence graph of D, which is TUM. Therefore, if l, u are integral then the polyhedron is integral. Checking if there is a feasible circulation in a graph with given l and u is at least as hard as solving the maximum flow problem.

Exercise 13 *Given D, $s, t \in V$ and a flow value F, show that checking if there is an $s - t$ flow of value F can be efficently reduced to checking if a given directed graph has a circulation respecting lower and upper bounds.*

The converse is also true however; one can reduce circulation problems to regular maximum-flow problems, though it takes a bit of work.

Min-cost circulation is the problem: $\min\{cx \mid l \leq x \leq u, Mx = 0\}$. We therefore obtain that

Theorem 14 *The min-cost circulation problem with lower and upper bounds can be solved in (strongly) polynomial time. Moreover, if l, u are integral then there exists an integral optimum solution.*

The analogue of max flow-min cut theorem in the circulation setting is Hoffman's circulation theorem.

Theorem 15 *Given $D = (V, A)$ and $l : A \to \mathbb{R}^+$ and $u : A \to \mathbb{R}^+$, there is a feasible circulation $x : A \to \mathbb{R}^+$ iff*

1. *$l(a) \leq c(a)$ $\forall a \in A$ and*
2. *$\forall U \subseteq V, l(\delta^-(U)) \leq c(\delta^+(U))$.*

Moreover, if l, u are integral then there is an integral circulation.

Exercise 16 *Prove Hoffman's theorem using TUM property of M and duality.*

b-Transshipments: One obtains slightly more general objects called transshipments as follows:

Definition 17 *Let $D = (V, E)$ be a directed graph and $b : A \to \mathbb{R}$. A b-transshipment is a function $f : A \to \mathbb{R}^+$ such that $\forall u \in V$, $f(\delta^-(u)) - f(\delta^+(u)) = b(u)$ i.e, the excess inflow at u is equal to $b(u)$.*

We think of nodes u with $b(u) < 0$ as supply nodes and $b(u) > 0$ as demand nodes. Note that $b = 0$ captures circulations. Once can generalize Hoffman's circulation theorem.

Theorem 18 *Given $D = (V, A)$, $b : V \to \mathbb{R}^+$ and $l : A \to \mathbb{R}^+$and $u : A \to \mathbb{R}^+$,there exists a b-transshipment respecting l, u iff*

1. *$l(a) \leq u(a)$ $\forall a \in A$ and*
2. *$\sum_{v \in V} b(v) = 0$ and*
3. *$\forall S \subseteq V$, $u(\delta^+(S)) \geq l(\delta^-(S)) + b(S)$.*

Moreover, if b, l, u are integral, there is an integral b-transshipment.

Exercise 19 *Derive the above theorem from Hoffman's circulation theorem.*

2.3 Interval graphs

A graph $G = (V, E)$ on n nodes is an interval graph if there exist a collection $\mathcal{I}$ of n closed intervals on the real line and a bijection $f : V \to \mathcal{I}$ such that $uv \in E$ iff $f(u)$ and $f(v)$ intersect. Given an interval graph, an interesting problem is to find a maximum weight independent set in G where $w : A \to \mathbb{R}^+$ is a weight function. This is same as asking for the maximum weight non-overlapping set of intervals in a collection of intervals.

We can write an LP for it. Let $\mathcal{I} = \{I_1, \ldots, I_n\}$

$$
\begin{aligned}
\max \sum_{i=1}^{n} w_i x_i & & & \\
\sum_{I_i : p \in I_i} x_i &\leq 1 & & \forall p \text{ in } \mathbb{R} \\
x_i &\geq 0 & & 1 \leq i \leq n
\end{aligned}
$$

Note that the constraints can be written only for a finite set of points which correspond to the end points of the intervals. These are the natural "clique" constraints: each maximal clique in G corresponds to a point p and all the intervals containing p. and clearly an independent set cannot pick more than one node from a clique.

The LP above is $\max\{wx \mid x \geq 0, Mx \leq 1\}$ where M is a consecutive ones matrix, and hence TUM. Therefore, the polyhedron is integral. We therefore have a polynomial time algorithm for the max-weight independent set problem in interval graphs. This problem can be easily solved efficienctly via dynamic programming. However, we observe that we can also solve $\max\{wx|x \geq 0, Mx \leq b\}$for any integer b and this is not easy to see via other methods.

To illustrate the use of integer decomposition properties of polyhedra, we derive a simple and well known fact.

Proposition 20 *Suppose we have a collection of intervals $\mathcal{I}$ such that $\forall p \in \mathbb{R}$ the maximum number of intervals containing p is at most k. Then $\mathcal{I}$ can be partitioned into $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$ such that each $\mathcal{I}_k$ is a collection of non-overlapping intervals. In other words, if G is an interval graph then $\omega(G) = \chi(G)$ where $\omega(G)$ is the clique-number of G and $\chi(G)$ is the chromatic number of G.*

One can prove the above easily via a greedy algorithm. We can also derive this by considering the independent set polytope $\{x \mid x \geq 0, Mx \leq 1\}$. We note that $x^* = \frac{1}{k}.\overline{1}$ is feasible for this polytope if no point p is contained in more than k intervals. Since P has the integer decomposition property, $y = kx^* = \overline{1}$ can be written as $x_1 + \ldots + x_k$ where x_i is integral $\forall 1 \leq i \leq k$ and $x_i \in P$. This gives the desired decomposition. The advantage of the polyhedral approach that one obtains a more general theorem by using an arbitrary integral b in the polytope $\{x \mid x \geq 0, Mx \leq b\}$ and this has applications; see for example [2].

References

[1] A. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, 1998.

[2] P. Winkler and L. Zhang. Wavelength Assignment and Generalized Interval Graph Coloring. ACM-SIAM SODA, 2003.

Gomory-Hu Trees

(The work in this section closely follows [3])

Let $G = (V, E)$ be an undirected graph with non-negative edge capacities defined by $c : E \rightarrow \mathbb{R}$. We would like to be able to compute the *global* minimum cut on the graph (i.e., the minimum over all min-cuts between pairs of vertices s and t). Clearly, this can be done by computing the minimum cut for all $\binom{n}{2}$ pairs of vertices, but this can take a lot of time. Gomory and Hu showed that the number of distinct cuts in the graph is at most $n - 1$, and furthermore that there is an efficient tree structure that can be maintained to compute this set of distinct cuts [1] (note that there is also a very nice randomized algorithm due to Karger and Stein that can compute the global minimum cut in near-linear time with high probability [2]).

An important note is that Gomory-Hu trees work because the cut function is both submodular and symmetric. We will see later that *any* submodular, symmetric function will induce a Gomory-Hu tree.

Definition 1. *Given a graph $G = (V, E)$, we define $\alpha_G(u, v)$ to be the value of a minimum u, v cut in G. Furthermore, for some set of vertices U, we define $\delta(U)$ to be the set of edges with one endpoint in U.*

Definition 2. *Let G, c, and α_G be defined as above. Then, a tree $T = (V(G), E_T)$ is a* **Gomory-Hu tree** *if for all $st \in E_T$, $\delta(W)$ is a minimum s, t cut in G, where W is one component of $T - st$.*

The natural question is whether such a tree even exists; we will return to this question shortly. However, if we are given such a tree for an arbitrary graph G, we know that this tree obeys some very nice properties. In particular, we can label the edges of the tree with the values of the minimum cuts, as the following theorem shows (an example of this can be seen in figure 1):

Theorem 1. *Let T be a Gomory-Hu tree for a graph $G = (V, E)$. Then, for all $u, v \in V$, let st be the edge on the unique path in T from u to v such that $\alpha_G(s, t)$ is minimized. Then,*

$$\alpha_G(u, v) = \alpha_G(s, t)$$

and the cut $\delta(W)$ induced by $T - st$ is a u, v minimum cut in G. Thus $\alpha_G(s, t) = \alpha_T(s, t)$ for each $s, t \in V$ where the capacity of an edge st in T is equal to $\alpha_G(s, t)$.

Proof. We first note that α_G obeys a triangle inequality. That is, $\alpha_G(a, b) \geq \min(\alpha_G(a, c), \alpha_G(b, c))$ for any undirected graph G and vertices a, b, c (to see this, note that c has to be on one side or the other of any a, b cut).

Consider the path from u to v in T. We note that if $uv = st$, then $\alpha_G(u, v) = \alpha_G(s, t)$. Otherwise, let $w \neq v$ be the neighbor of u on the u-v path in T. By the triangle inequality mentioned above, $\alpha_G(u, v) \geq \min(\alpha_G(u, w), \alpha_G(w, v))$. If $uw = st$, then $\alpha_G(u, v) \geq \alpha_G(s, t)$; otherwise, by induction on the path length, we have that $\alpha_G(u, v) \geq \alpha_G(w, v) \geq \alpha_G(s, t)$.

However, by the definition of Gomory-Hu trees, we have that $\alpha_G(u, v) \leq \alpha_G(s, t)$, since the cut induced by $T - st$ is a valid cut for u, v. Thus, we have $\alpha_G(u, v) = \alpha_G(s, t)$ and the cut induced by $T - st$ is a u, v minimum cut in G. ■

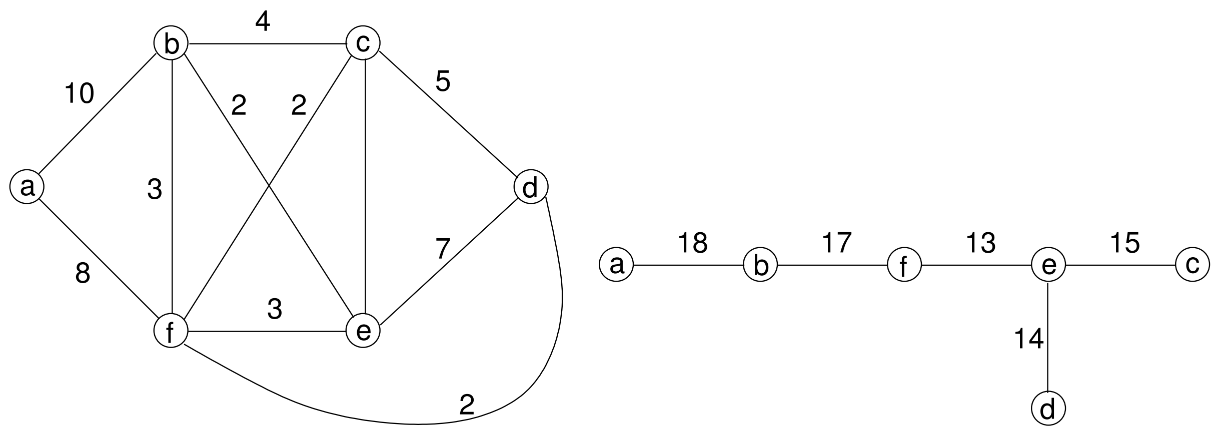


Figure 1: A graph G with its corresponding Gomory-Hu tree [4].

Remark 2. *Gomory-Hu trees can be (and are often) defined by asking for the property described in Theorem 1. However, the proof shows that the basic requirement in Definition 2 implies the other property.*

The above theorem shows that we can represent compactly all of the minimum cuts in an undirected graph. Several non-trivial facts about undirected graphs fall out of the definition and the above result. The only remaining question is "Does such a tree exist? And if so, how does one compute it efficiently?" We will answer both questions by giving a constructive proof of Gomory-Hu trees for any undirected graph G. However, first we must discuss some properties of submodular functions.

Definition 3. *Given a finite set E, $f : 2^E \to \mathbb{R}$ is* **submodular** *if for all $A, B \in 2^E$, $f(A)+f(B) \geq f(A \cup B) + f(A \cap B)$.*

An alternate definition based on the idea of "decreasing marginal value" is the following:

Definition 4. *Given E and f as above, f is submodular if $f(A+e) - f(A) \geq f(B+e) - f(B)$ for all $A \subseteq B$ and $e \in E$.*

To see the equivalence of these definitions, let $f_A(e) = f(A+e) - f(A)$, and similarly for $f_B(e)$. Take any $A, B \subseteq E$ and $e \in E$ such that $A \subseteq B$, and let f be submodular according to definition 3. Then $f(A+e) + f(B) \geq f((A+e) \cup B) + f((A+e) \cap B) = f(B+e) + f(A)$. Rearranging shows that $f_A(e) \geq f_B(e)$. Showing that definition 4 implies definition 3 is slightly more complicated, but can be done (**Exercise**).

There are three types of submodular functions that will be of interest:

1. Arbitrary submodular functions

2. Non-negative (range is $[0, \infty)$). Two subclasses of non-negative submodular functions are monotone ($f(A) \leq f(B)$ whenever $A \subseteq B$) and non-monotone.

3. Symmetric submodular functions where $f(A) = f(E \setminus A)$ for all $A \subseteq E$.

As an example of a submodular function, consider a graph $G = (V, E)$ with capacity function $c : E \to \mathbb{R}^+$. Then $f : 2^V\ to \mathbb{R}^+$ defined by $f(A) = c(\delta(A))$ (i.e., the capacity of a cut induced by a set A) is submodular.

To see this, notice that $f(A) + f(B) = a + b + 2c + d + e + 2f$, for any arbitrary A and B, and a, b, c, d, e, f are as shown in figure 2. Here, a (for example) represents the total capacity of edges with one endpoint in A and the other in $V \setminus (A \cup B)$. Also notice that $f(A \cup B) + f(A \cap B) = a + b + 2c + d + e$, and since all values are positive, we see that $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$, satisfying definition 3.

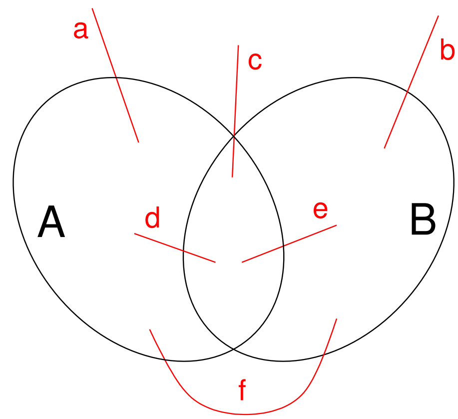


Figure 2: Given a graph G and two sets $A, B \subseteq V$, this diagram shows all of the possible classes of edges of interest in G. In particular, there could be edges with both endpoints in $V \setminus (A \cup B)$, A, or B that are not shown here.

Exercise 1. *Show that cut function on the vertices of a directed graph is submodular.*

Another nice property about this function f is that it is **posi-modular**, meaning that $f(A) + f(B) \geq f(A - B) + f(B - A)$. In fact, posi-modularity follows for any symmetric submodular function:

$$
\begin{aligned}
f(A) + f(B) = f(V - A) + f(B) &\geq f((V - A) \cap B) + f((V - A) \cup B) \\
&= f(B - A) + f(V - (A - B)) \\
&= f(B - A) + f(A - B)
\end{aligned}
$$

We use symmetry in the first and last lines above. In fact, it turns out that the above two properties of the cut function are the *only* two properties necessary for the proof of existence of Gomory-Hu trees. As mentioned before, this will give us a Gomory-Hu tree for *any* non-negative symmetric submodular function. We now prove the following lemma, which will be instrumental in constructing Gomory-Hu trees:

Key Lemma. *Let $\delta(W)$ be an s, t minimum cut in a graph G with respect to a capacity function c. Then for any $u, v \in W, u \neq v$, there is a u, v minimum cut $\delta(X)$ where $X \subseteq W$.*

Proof. Let $\delta(X)$ be any u, v minimum cut that crosses W. Suppose without loss of generality that $s \in W, s \in X$, and $u \in X$. If one of these are not the case, we can invert the roles of s and t or X and $V \setminus X$. Then there are two cases to consider:

Case 1: $t \notin X$ (see figure 3). Then, since c is submodular,

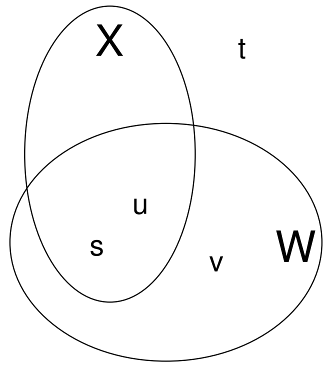


Figure 3: $\delta(W)$ is a minimum s,t cut. $\delta(X)$ is a minimum u,v cut that crosses W. This diagram shows the situation in Case 1; a similar picture can be drawn for Case 2

$$c(\delta(X)) + c(\delta(W)) \geq c(\delta(X \cap W)) + c(\delta(X \cup W)) \tag{1}$$

But notice that $\delta(X \cap W)$ is a u,v cut, so since $\delta(X)$ is a minimum cut, we have $c(\delta(X \cap W)) \geq c(\delta(X))$. Also, $X \cup W$ is a s,t cut, so $c(\delta(X \cup W)) \geq c(\delta(W))$. Thus, equality holds in equation (1), and $X \cap W$ is a minimum u,v cut.

Case 2: $t \in X$. Since c is posi-modular, we have that

$$c(\delta(X)) + c(\delta(W)) \geq c(\delta(W \setminus X)) + c(\delta(X \setminus W)) \tag{2}$$

However, $\delta(W \setminus X)$ is a u,v cut, so $c(\delta(W \setminus X)) \geq c(\delta(X))$. Similarly, $\delta(X \setminus W)$ is an s,t cut, so $c(\delta(X \setminus W)) \geq c(\delta(W))$. Therefore, equality holds in equation (2), and $W \setminus X$ is a u,v minimum cut.

■

The above argument shows that minimum cuts can be **uncrossed**, a technique that is useful in many settings. In order to construct a Gomory-Hu tree for a graph, we need to consider a slightly generalized definition:

Definition 5. *Let $G = (V, E), R \subseteq V$. Then a* **Gomory-Hu tree for R in G** *is a pair consisting of $T = (R, E_T)$ and a partition $(C_r \mid r \in R)$ of V associated with each $r \in R$ such that*

1. *For all $r \in R$, $r \in C_r$*
2. *For all $st \in E_T$, $T - st$ induces a minimum cut in G between s and t defined by*

$$\delta(U) = \bigcup_{r \in X} C_r$$

where X is the vertex set of a component of $T - st$.

Notice that a Gomory-Hu tree for G is simply a generalized Gomory-Hu tree with $R = V$.

Algorithm 1 GomoryHuAlg(G, R)

if $|R| = 1$ **then**
  return $T = (\{r\}, \varnothing), C_r = V$
else
  Let $r_1, r_2 \in R$, and let $\delta(W)$ be an r_1, r_2 minimum cut

  ⟨⟨*Create two subinstances of the problem*⟩⟩
  $G_1 = G$ with $V \setminus W$ shrunk to a single vertex, v_1; $R_1 = R \cap W$
  $G_2 = G$ with W shrunk to a single vertex, v_2; $R_2 = R \setminus W$

  ⟨⟨*Now we recurse*⟩⟩
  $T_1, (C_r^1 \mid r \in R_1) =$ GomoryHuAlg(G_1, R_1)
  $T_2, (C_r^2 \mid r \in R_2) =$ GomoryHuAlg(G_2, R_2)

  ⟨⟨*Note that r', r'' are not necessarily r_1, r_2!*⟩⟩
  Let r' be the vertex such that $v_1 \in C_{r'}^1$
  Let r'' be the vertex such that $v_2 \in C_{r''}^2$

  ⟨⟨*See figure 4*⟩⟩
  $T = (R_1 \cup R_2, E_{T_1} \cup E_{T_2} \cup \{rr'\})$
  $(C_r \mid r \in R) =$ ComputePartitions$(R_1, R_2, C_r^1, C_r^2, r', r'')$
  return T, C_r
end if

Algorithm 2 ComputePartitions$(R_1, R_2, C_r^1, c_r^2, r', r'')$

⟨⟨*We use the returned partitions, except we remove v_1 and v_2 from $C_{r'}$ and $C_{r''}$, respectively*⟩⟩
For $r \in R_1$, $r \neq r'$, $C_r = C_r^1$
For $r \in R_1$, $r \neq r''$, $C_r = C_r^2$
$C_{r'} = C_{r'}^1 - \{v_1\}, C_{r''} = C_{r''}^2 - \{v_2\}$
return $(C_r \mid r \in R)$

Intuitively, we associate with each vertex v in the tree a "bucket" that contains all of the vertices that have to appear on the same side as v in some minimum cut. This allows us to define the algorithm GomoryHuAlg.

Theorem 3. GomoryHuAlg *returns a valid Gomory-Hu tree for a set R.*

Proof. We need to show that any $st \in E_T$ satisfies the "key property" of Gomory-Hu trees. That is, we need to show that $T - st$ induces a minimum cut in G between s and t. The base case is trivial. Then, suppose that $st \in T_1$ or $st \in T_2$. By the Key Lemma, we can ignore all of the vertices outside of T_1 or T_2, because they have no effect on the minimum cut, and by our induction hypothesis, we know that T_1 and T_2 are correct.

Thus, the only edge we need to care about is the edge we added from r' to r''. First, consider the simple case when $\alpha_G(r_1, r_2)$ is minimum over all pairs of vertices in R. In this case, we see that in particular, $\alpha_G(r_1, r_2) \leq \alpha_G(r', r'')$, so we are done.

However, in general this may not always be the case. Let $\delta(W)$ be a minimum cut between r_1 and r_2, and suppose that there is a smaller r', r'' minimum cut $\delta(X)$ than what W induces; that is $c(\delta(X)) < (\delta(W))$. Assume without loss of generality that $r_1, r' \in W$. Notice that if $r_1 \in X$, we

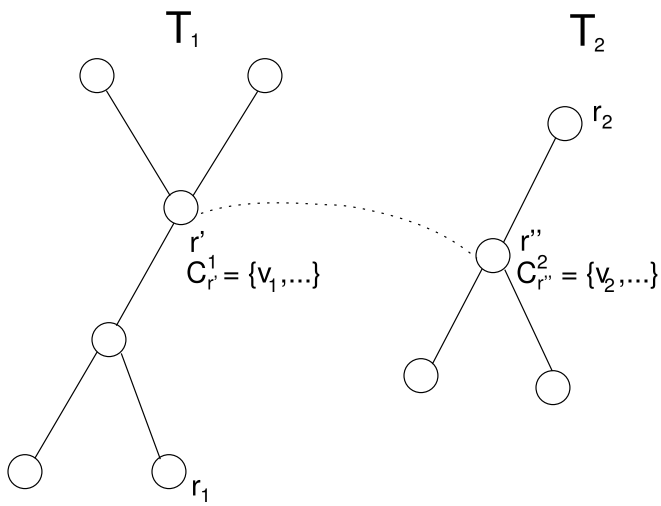


Figure 4: T_1 and T_2 have been recursively computed by GomoryHuAlg. Then we find r' and r'' such that v_1 (the shrunken vertex corresponding to $V \setminus W$ in T_1) is in the partition of r', and similarly for r'' and v_2. Then, to compute T, we connect r' and r'', and recompute the partitions for the whole tree according to ComputePartitions.

have a smaller r_1, r_2 cut than $\delta(W)$, and similarly if $r_2 \in X$. So, it is clear that X separates r_1 and r'. By our key lemma, we can then uncross and assume that $X \subseteq W$.

Now, consider the path from r' to r_1 in T_1. There exists an edge uv on this path such that the weight of uv in T_1, $w_1(uv)$, is at most $c(\delta(X))$. Because T_1 is a Gomory-Hu tree, uv induces an r_1, r_2 cut in G of capacity $w_1(uv)$ (since $v_1 \in C^1_{r'}$). But this contradicts the fact that W is a r_1, r_2 minimum cut. Threfore, e can pick r_1 and r_2 arbitrarily from R_1 and R_2, and GomoryHuAlg is correct. ■

This immediately implies the following corollary:

Corollary 4. *A Gomory-Hu tree for $R \subseteq V$ in G can be computed in the time needed to compute $|R| - 1$ minimum-cuts in graphs of size at most that of G.*

Finally, we present the following alternative proof of the last step of theorem 3 (that is, showing that we can choose r_1 and r_2 arbitrarily in GomoryHuAlg). As before, let $\delta(W)$ be an r_1, r_2 minimum cut, and assume that $r_1 \in W, r_2 \in V \setminus W$. Assume for simplicity that $r_1 \neq r'$ and $r_2 \neq r''$ (the other cases are similar). We claim that $\alpha_{G_1}(r_1, r') = \alpha_G(r_1, r') \geq \alpha_G(r_1, r_2)$. To see this, note that if $\alpha_{G_1}(r_1, r') < \alpha_G(r_1, r_2)$, there is an edge $uv \in E_{T_1}$ on the path from r_1 to r' that has weight less than $\alpha_G(r_1, r_2)$, which gives a smaller r_1, r_2 cut in G than W (since $v_1 \in C^1_{r'}$). For similar reasons, we see that $\alpha_G(r_2, r'') \geq \alpha_G(r_1, r_2)$.

Thus, by the triangle inequality we have

$$\alpha_G(r', r'') \geq \min(\alpha_G(r', r_1), \alpha_G(r'', r_2), \alpha_G(r_1, r_2) \geq \alpha_G(r_1, r_2)$$

which completes the proof.

Gomory-Hu trees allow one to easily show some facts that are otherwise hard to prove directly. Some examples are the following.

Exercise 2. *For any undirected graph there is a pair of nodes s, t and an s-t minimum cut consisting of a singleton node (either s or t). Such a pair is called a* pendant *pair.*

Exercise 3. *Let G be a graph such that $deg(v) \geq k$ for all $v \in V$. Show that there is some pair s, t such that $\alpha_G(s, t) \geq k$.*

Notice that the proof of the correctness of the algorithm relied only on the key lemma which in turn used only the symmetry and submodularity of the cut function. One can directly extend the proof to show the following theorem.

Theorem 5. *Let V be a ground set, and let $f : 2^V \to \mathbb{R}^+$ be a symmetric submodular function. Given s, t in V, define the minimum cut between s and t as*

$$\alpha_f(s,t) = \min_{W \subseteq V, |W \cap \{s,t\}|=1} f(W)$$

Then, there is a Gomory-Hu tree that represents α_f. That is, there is a tree $T = (V, E_T)$ and a capacity function $c : E_T \to \mathbb{R}^+$ such that $\alpha_f(s,t) = \alpha_T(s,t)$ for all s, t in V, and moreover, the minimum cut in T induces a minimum cut according to f for each s, t.

Exercise 4. *Let $G = (V, \xi)$ be a hypergraph. That is, each hyper-edge $S \in \xi$ is a subset of V. Define $f : 2^V \to \mathbb{R}^+$ as $f(W) = |\delta(W)|$, where $S \in \xi$ is in $\delta(W)$ iff $S \cap W$ and $S \setminus W$ are non-empty. Show that f is a symmetric, submodular function.*

References

[1] R. E. Gomory, T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, 1961.

[2] D. R. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, vol. 47, 2000.

[3] A. Schrijver. Combinatorial Optimization. Springer-Verlag Berlin Heidelberg, 2003. Chapter 15.4.

[4] V. Vazirani. Approximation Algorithms. Springer, 2004.

CS 598CSC: Combinatorial Optimization	Lecture date: Feb 9, 2010
Instructor: Chandra Chekuri	Scribe: Matthew Yancey

1 Matchings in Non-Bipartite Graphs

We discuss matching in general undirected graphs. Given a graph G, $\nu(G)$ denotes the size of the largest matching in G. We follow [1] (Chapter 24).

1.1 Tutte-Berge Formula for $\nu(G)$

Tutte (1947) proved the following basic result on perfect matchings.

Theorem 1 (Tutte) *A graph $G = (V, E)$ has a prefect matching iff $G - U$ has at most $|U|$ odd components for each $U \subseteq V$.*

Berge (1958) generalized Tutte's theorem to obtain a min-max formula for $\nu(G)$ which is now called the Tutte-Berge formula.

Theorem 2 (Tutte-Berge Formula) *For any graph $G = (V, E)$,*

$$\nu(G) = \frac{|V|}{2} - \max_{U \subseteq V} \frac{o(G-U) - |U|}{2}$$

where $o(G - U)$ is the number of components of $G - U$ with an odd number of vertices.

Proof: We have already seen the easy direction that for any U, $\nu(G) \le \frac{|V|}{2} - \frac{o(G-U)-|U|}{2}$ by noticing that $o(G - U) - |U|$ is the number of nodes from the odd components in $G - U$ that must remain unmatched.

Therefore, it is sufficient to show that $\nu(G) = \frac{|V|}{2} - \max_{U \subseteq V} \frac{o(G-U)-|U|}{2}$. Any reference to left-hand side (LHS) or right-hand side (RHS) will be in reference to this inequality. Proof via induction on $|V|$. Base case of $|V| = 0$ is trivial.

Case 1: There exists $v \in V$ such that v is in every maximum matching. Let $G' = (V', E') = G - v$, then $\nu(G') = \nu(G) - 1$ and by induction, there is $U' \subseteq V'$ such that the RHS of the formula is equal to $\nu(G') = \nu(G) - 1$. It is easy to verify that $U = U' \cup \{v\}$ satisfies equality in the formula for G.

Case 2: For every $v \in G$, there is a maximum matching that misses it. By Claim 3 below, $\nu(G) = \frac{|V|-1}{2}$ and that there is an odd number of vertices in the entire graph. If we take $U = \emptyset$, then the theorem holds. □

Claim 3 *Let $G = (V, E)$ be a graph such that for each $v \in V$ there is a maximum matching in G that misses v. Then, $\nu(G) = \frac{|V|-1}{2}$. In particular, $|V|$ is odd.*

Proof: G is necessarily connected. By way of contradiction, assume there exists two vertices $u \neq v$ and a maximum matching M that avoids them. Among all such choices, choose M, u, v such that $dist(u, v)$ is minimized. If $dist(u, v) = 1$ then M can be grown by adding uv to it. Therefore there exists a vertex t, $u \neq t \neq v$, such that t is on a shortest path from u to v. Also, by minimality of distance between u and v we know that $t \in M$.

By the assumption, there is at least one maximum matching that misses t. We are going to choose a maximum matching N that maximizes $N \cap M$ while missing t. N must cover u, or else N, u, t would have been a better choice above. Similarly, N covers v. Now $|M| = |N|$ and we have found one vertex $t \in M - N$ and two $u, v \in N - M$, so there must be another vertex $x \in M - N$ that is different from all of the above. Let $xy \in M$. N is maximal, so xy can't be added to it. Thus, we must have that $y \in N$ and that means $y \neq t$. Let $yz \in N$. Then we have that $z \in N - M$ because $xy \in M$ and $z \neq x$.

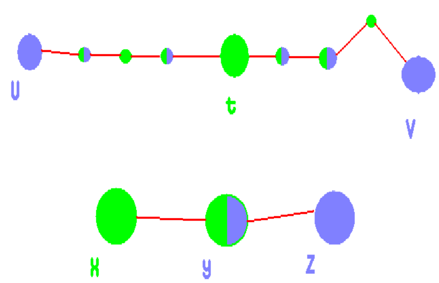


Figure 1: Green vertices are in M. Blue vertices are in N.

Consider the matching $N' = N - yz + xy$. We have that $|N'| = |N|$ and N' avoids t and $|N' \cap M| > |N \cap M|$. This is a contradiction. $\square$

2 Algorithm for Maximum Cardinality Matching

We now describe a polynomial time algorithm for finding a maximum cardinality matching in a graph, due to Edmonds. Faster algorithms are now known but the fundamental insight is easier to see in the original algorithm. Given a matching M in a graph G, we say that a node v is M-exposed if it is not covered by an edge of M.

Definition 4 *A path P in G is M-alternating if every other edge is in M. It can have odd or even length. A path P is M-augmenting if it is M-alternating and both ends are M-exposed.*

Lemma 5 *M is a maximum matching in G if and only if there is no M-augmenting path.*

Proof: If there is an M-augmenting path, then we could easily use it to grow M and it would not be a maximum matching.

In the other direction, assume that M is a matching that is not maximum by way of contradiction. Then there is a maximum matching N, and $|N| > |M|$. Let H be a subgraph of G induced by the edge set $M \Delta N = (M - N) \cup (N - M)$ (the symmetric difference). Note that the maximum degree of a node in H is at most 2 since a node can be incident to at most one edge from $N - M$ and one edge from $M - N$. Therefore, H is a disjoint collection of paths and cycles. Furthermore, all paths are M-alternating (and N-alternating too). All cycles must be of even length, since they alternate edges from M and N too. At least one of the paths must have more N edges than M edges because $|N| > |M|$ and we deleted the same number of edges from N as M. That path is an M-augmenting path. $\square$

The above lemma suggests a greedy algorithm for finding a maximum matching in a graph G. Start with a (possibly empty) matching and iteratively augment it by finding an augmenting path, if one exists. Thus the heart of the matter is to find an *efficient* algorithm that given G and matching M, either finds an M-augmenting path or reports that there is none.

Bipartite Graphs: We quickly sketch why the problem of finding M-augmenting paths is relatively easy in bipartite graphs. Let $G = (V, E)$ with A, B forming the vertex bipartition. Let M be a matching in G. Let X be the M-exposed vertices in A and let Y be the M-exposed vertices in B. Obtain a directed graph $D = (V, E')$ by orienting the edges of G as follows: orient edges in M from B to A and orient edges in $E \setminus M$ from A to B. The following claim is easy to prove.

Claim 6 *There is an M-augmenting path in G if and only if there is an X-Y path in the directed graph D described above.*

Non-Bipartite Graphs: In general graphs it is not straight forward to find an M-augmenting path. As we will see, odd cycles form a barrier and Edmonds discovered the idea of shrinking them in order to recursively find a path. The first observation is that one can efficiently find an alternating *walk*.

Definition 7 *A walk in a graph $G = (V, E)$ is a finite sequence of vertices $v_0, v_1, v_2, \ldots, v_t$ such that $v_i v_{i+1} \in E, 0 \leq i \leq t-1$. The length of the walk is t.*

Note that edges and nodes can be repeated on a walk.

Definition 8 *A walk $v_0, v_1, v_2, \ldots, v_t$ is M-alternating walk if for each $1 \leq i \leq t-1$, exactly one of $v_{i-1}v_i$ and $v_i v_{i+1}$ is in M.*

Lemma 9 *Given a graph $G = (V, E)$, a matching M, and M-exposed nodes X, there is an $O(|V| + |E|)$ time algorithm that either finds a shortest M-alternating X-X walk of positive length or reports that there is no such walk.*

Proof Sketch. Define a directed graph $D = (V, A)$ where $A = \{(u, v) : \exists x \in V, ux \in E, xv \in M\}$. Then a X-X M-alternating walk corresponds to a X-$N(X)$ directed path in D where $N(X)$ is the set of neighbors of X in G (we can assume there is no edge between two nodes in X for otherwise that would be a shortest walk). Alternatively, we can create a bipartite graph with $D = (V \cup V', A)$ where V' is a copy of V and $A = \{(u, v') \mid uv \in E \setminus M\} \cup \{(u', v) \mid uv \in M\}$ and find a shortest X-X' directed path in D where X' is the copy of X in V'. $\square$

What is the structure of an X-X M-alternating walk? Clearly, one possibility is that it is actually a path in which case it will be an M-augmenting path. However, there can be alternating walks that are not paths as shown by the figure below.

One notices that if an X-X M-alternating walk has an even cycle, one can remove it to obtain a shorter alternating walk. Thus, the main feature of an alternating walk when it is not a path is the presence of an *odd* cycle called a *blossom* by Edmonds.

Definition 10 *An M-flower is an M-alternating walk $v_0, v_1, \ldots, v_t$ such that $v_o \in X$, t is odd and $v_t = v_i$ for some even $i < t$. In other words, it consists of an even length $v_0, \ldots, v_i$ M-alternating path (called the stem) attached to an odd cycle $v_i, v_{i+1}, \ldots, v_t = v_i$ called the M-blossom. The node v_i is the base of the stem and is M-exposed if $i = 0$, otherwise it is M-covered.*

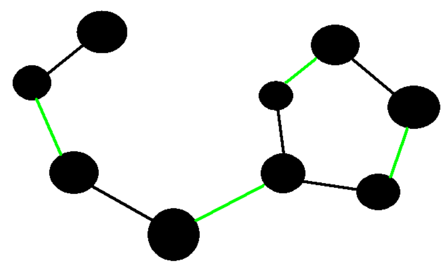


Figure 2: A M-flower. The green edges are in the matching

Lemma 11 *A shortest positive length X-X M-alternating walk is either an M-augmenting path or contains an M-flower as a prefix.*

Proof: Let $v_0, v_1, \ldots, v_t$ be a shortest X-X M-alternating walk of positive length. If the walk is a path then it is M-augmenting. Otherwise let i be the smallest index such that $v_i = v_j$ for some $j > i$ and choose j to be smallest index such that $v_i = v_j$. If $v_i, \ldots, v_j$ is an even length cycle we can eliminated it from the walk and obtain a shorter alternating walk. Otherwise, $v_0, \ldots, v_i, \ldots, v_j$ is the desired M-flower with v_i as the base of the stem. □

Given a M-flower and its blossom B (we think of B as both a set of vertices and an odd cycle), we obtain a graph G/B by shrinking B to a single vertex b and eliminating loops and parallel edges. It is useful to identify b with the base of the stem. We obtain a matching M/B in G/B which consists of eliminating the edges of M with both end points in B. We note that b is M/B-exposed iff b is M-exposed.

Theorem 12 *M is a maximum matching in G if and only if M/B is a maximum matching in G/B.*

Proof: The next two lemmas cover both directions. □

To simplify the proof we do the following. Let $P = v_0, \ldots, v_i$ be the stem of the M-flower. Note that P is an even length M-alternating path and if $v_0 \neq v_i$ then v_0 is M-exposed and v_i is

M-covered. Consider the matching $M' = M\Delta E(P)$, that is by switching the matching edges in P into non-matching edges and vice-versa. Note that $|M'| = |M|$ and hence M is a maximum matching in G iff M' is a maximum matching. Now, the blossom $B = v_i, \ldots, v_t = v_i$ is also a M'-flower but with a degenerate stem and hence the base is M'-exposed. For the proofs to follow we will assume that $M = M'$ and therefore b is an exposed node in G/B. In particular we will assume that $B = v_0, v_1, \ldots, v_t = v_0$ with t odd.

Proposition 13 *For each v_i in B there is an even-length M-alternating path Q_i from v_0 to v_i.*

Proof: If i is even then $v_0, v_1, \ldots, v_i$ is the desired path, else if i is odd, $v_0 = v_t, v_{t-1}, \ldots, v_i$ is the desired path. That is, we walk along the odd cycle one direction or the other to get an even length path. □

Lemma 14 *If there is an M/B augmenting path P in G/B then there is an M-augmenting path P' in G. Moreover, P' can be found from P in $O(m)$ time.*

Proof:

Case 1: P does not contain b. Set $P' = P$.

Case 2: P contains b. b is an exposed node, so it must be an endpoint of P. Without loss of generality, assume b is the first node in P. Then P starts with an edge $bu \notin M/B$ and the edge bu corresponds to an edge v_iu in G where $v_i \in B$. Obtain path P' by concatenating the even length M-alternating path Q_i from v_0 to v_i from Proposition 13 with the path P in which b is replaced by v_i; it is easy to verify that is an M-augmenting path in G. □

Lemma 15 *If P is an M-augmenting path in G, then there exists an M/B augmenting path in G/B.*

Proof: Let $P = u_0, u_1, \ldots, u_s$ be an M-augmenting path in G. If $P \cap B = \emptyset$ then P is an M/B augmenting path in G/B and we are done. Assume $u_0 \neq v_0$ - if this is not true, flip the path backwards. Let u_j be the first vertex in P that is in B. Then $u_0, u_1, \ldots, u_{j-1}, b$ is an M/B augmenting path in G/B. Two cases to verify when $u_j = v_0$ and when $u_j = v_i$ for $i \neq 0$, both are easy. □

Remark 16 *The proof of Lemma 14 is easy when b is not M-exposed. Lemma 15 is not straight forward if b is not M-exposed.*

From the above lemmas we have the following.

Lemma 17 *There is an $O(nm)$ time algorithm that given a graph G and a matching M, either finds an M-augmenting path or reports that there is none. Here $m = |E|$ and $n = |V|$.*

Proof: The algorithm is as follows. Let X be the M-exposed nodes. It first computes a shortest X-X M-alternating walk P in $O(m)$ time — see Lemma 9. If there is no such walk then clearly M is maximum and there is no M-augmenting path. If P is an M-augmenting path we are done. Otherwise there is an M-flower in P and a blossom B. The algorithm shrinks B and obtains G/B and M/B which can be done in $O(m)$ time. It then calls itself recursively to find an M/B-augmenting path or find out that M/B is a maximum matching in G/B. In the latter case, M is a maximum matching in G. In the former case the M/B augmenting path can be extended to an

M-augmenting path in $O(m)$ time as shown in Lemma 14. Since G/B has at least two nodes less than G, it follows that his recursive algorithm takes at most $O(nm)$ time. $\square$

By iteratively using the augmenting algorithm from the above lemma at most $n/2$ times we obtain the following result.

Theorem 18 *There is an $O(n^2m)$ time algorithm to find a maximum cardinality matching in a graph with n nodes and m edges.*

The fastest known algorithm for this problem has a running time of $O(m\sqrt{n})$ and is due to Micali and Vazirani with an involved formal proof appearing in [3]; an exposition of this algorithm can be found in [2].

References

[1] A. Schrijver. Combinatorial Optimization. Springer-Verlag Berlin Heidelberg, 2003

[2] P. Peterson and M. Loui. The General Maximum Matching Algorithm of Micali and Vazirani. *Algorithmica*, 3:511-533, 1998.

[3] V. Vazirani. A Theory of Alternating Paths and Blossoms for Proving Correctness of the $O(|E|\sqrt{|V|})$ General Graph Maximum Matching Algorithm. *Combinatorica*, 14(1):71–109, 1994.

1 Edmonds-Gallai Decomposition and Factor-Critical Graphs

This material is based on [1], and also borrows from [2] (Chapter 24).

Recall the Tutte-Berge formula for the size of a maximum matching in graph G.

Theorem 1 (Tutte-Berge) *Given a graph G, the size of a maximum cardinality matching in G, denoted by $\nu(G)$, is given by:*

$$\nu(G) = \min_{U \subseteq V} \frac{1}{2}(|V| + |U| - o(G - U))$$

where $o(G-U)$ is the number of connected components in $G[V \setminus U]$ with odd cardinality.

We call a set U that achieves the minimum on the right hand side of the Tutte-Berge formula, a Tutte-Berge witness set. Such a set U gives some information on the set of maximum matchings in G. In particular we have the following.

- All nodes in U are covered in every maximum matching of G.
- If K is the vertex set of a component of $G-U$, then every maximum matching in G covers at least $\lfloor K/2 \rfloor$ nodes in K. In particular, every node in an even component is covered by every maximum matching.

A graph can have different Tutte-Berge witness sets as the example in Fig 1 shows. Clearly $U = \{v\}$ is more useful than $U = \emptyset$.

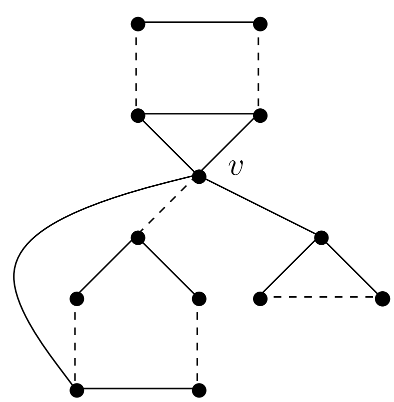


Figure 1: Graph G above has 13 nodes, and $\nu(G) = 6$. $U = \emptyset$ and $U = \{v\}$ are both Tutte-Berge witness sets.

A natural question is whether each graph has a *canonical* Tutte-Berge witness set that gives us as much information as possible. The content of the Edmonds-Gallai decomposition is to give a description of such a canonical witness set. Before we give the theorem, we digress to describe some other settings with canonical witness sets. The reader can skip the next paragraph.

Let $D = (V, A)$ be a directed graph and $s, t \in V$. It is easy to see that s has no directed path to t iff there is a set $X \subseteq V$ such that $s \in X$, $t \notin X$ and $\delta^+(X) = \emptyset$, that is no arcs leave X. Among all such sets X, the set X^* defined as the set of all nodes reachable from s, is a canonical set. It is a simultaneous witness for all nodes that are not reachable from s. Moreover, most algorithms for checking reachability of t from s would output X^*. Similarly, consider the s-t maximum flow problem in a capacitated directed graph D. By the maxflow-mincut theorem, the maximum flow value F is equal to the capacity of a minimum capacity cut $\delta(X)$ that separates s from t. Again, there could be multiple minimum cuts. One can show that if $\delta(X)$ and $\delta(Y)$ are s-t minimum cuts (here X and Y contain s and do not contain t) then $\delta(X \cap Y)$ and $\delta(X \cup Y)$ are also minimum cuts (follows from submodularity of the cut function). From this, it follows that there exists a unique minimal minimum cut $\delta(X^*)$ and a unique maximal minimum cut $\delta(Y^*)$. We note that X^* is precisely the set of vertices reachable from s in the residual graph of *any* maximum flow; similarly $V \setminus Y^*$ is the set of nodes that can reach t in the residual graph of any maximum flow.

Factor-Critical Graphs: If U is a non-empty Tutte-Berge witness set for a graph G, then it follows that there are nodes in G that are covered in every maximum matching.

Definition 2 *A graph $G = (V, E)$ is* factor-critical *if G has no perfect matching but for each $v \in V$, $G - v$ has a perfect matching.*

Factor-critical graphs are connected and have an odd number of vertices. Simple examples include odd cycles and the complete graph on an odd number of vertices.

Theorem 3 *A graph G is factor-critical if and only if for each node v there is a maximum matching that misses v.*

Proof: If G is factor-critical then $G - v$ has a perfect matching and hence a maximum matching in G. We saw the converse direction in the proof of the Tutte-Berge formula — it was shown that if each node v is missed by some maximum matching then G has a matching of size $(|V| - 1)/2$. $\square$

If G is factor-critical then $U = \emptyset$ is the unique Tutte-Berge witness set for G for otherwise there would be a node that is in every maximum matching. In fact the converse is also true, but is not obvious. It is an easy consequence of the Edmonds-Gallai decomposition to be seen shortly. We give a useful fact about factor-critical graphs.

Proposition 4 *Let C be an odd cycle in G. If the graph G/C, obtained by shrinking C into a single vertex, is factor-critical then G is factor-critical.*

Proof Sketch. Let c denote the vertex in G/C in place of the shrunken cycle C. Let v be an arbitrary node in $V(G)$. We need to show that $G - v$ has a perfect matching.

If $v \notin C$ then $G/C - v$ has a perfect matching M that matches c, say via edge cu. When we unshrink c into C, let w be the vertex in C that corresponds to the edge cu. We can extend M a perfect matching in $G - v$ by adding edges in the even length path $C - w$ to cover all the nodes in $C - w$.

If $v \in C$, consider a perfect matching M in $G/C - c$. It is again easy to extend M to a perfect matching in $G - v$ by considering $C - v$. $\square$

We will see later a structural characterization of factor-critical graphs via ear decompositions.

1.1 Edmonds-Gallai Decomposition

Theorem 5 (Edmonds-Gallai) *Given a graph $G = (V, E)$, let*

$$\begin{aligned} D(G) &:= \{v \in V \mid \textit{there exists a maximum matching that misses } v\} \\ A(G) &:= \{v \in V \mid v \textit{ is a neighbor of } D(G) \textit{ but } v \notin D(G)\} \\ C(G) &:= V \setminus (D(G) \cup A(G)). \end{aligned}$$

Then, the following hold.

1. *The set $U = A(G)$ is a Tutte-Berge witness set for G.*
2. *$C(G)$ is the union of the even components of $G - A(G)$.*
3. *$D(G)$ is the union of the odd components of $G - A(G)$.*
4. *Each component in $G - A(G)$ is factor-critical.*

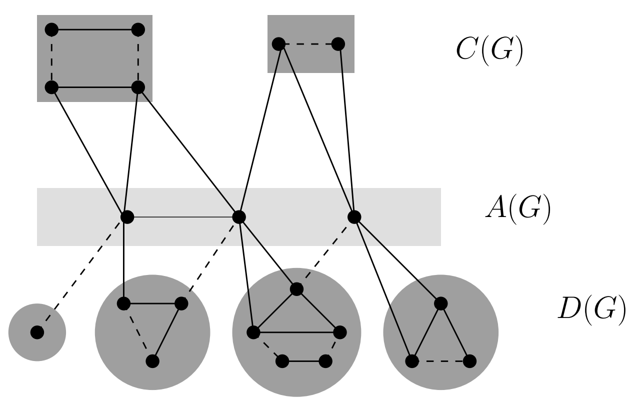


Figure 2: Edmonds-Gallai Decomposition

Corollary 6 *A graph G is factor-critical if and only if $U = \emptyset$ is the unique Tutte-Berge witness set for G.*

We prove the theorem in the rest of this section. We make use of the properties, and proof of correctness, of Edmonds algorithm for maximum cardinality matching that we discussed in the previous lecture.

Let M be any maximum matching in G and let X be the M-exposed nodes. We define three sets of nodes with respect to M and X.

$$\begin{aligned} \textsc{Even}(G, M) &:= \{v \in V \mid \text{there is an even length } M\text{-alternating } X\text{-}v \text{ path}\} \\ \textsc{Odd}(G, M) &:= \{v \in V \mid \text{there is an } M\text{-alternating } X\text{-}v \text{ path}\} \setminus \textsc{Even} \\ \textsc{Free}(G, M) &:= \{v \in V \mid \text{there is no } M\text{-alternating } X\text{-}v \text{ path}\} \end{aligned}$$

Note that $v \in \textsc{Odd}(G, M)$ implies that there is an odd length M-alternating X-v path but no even length path. A node $v \in \textsc{Even}(G, M)$ may have both an even and odd length path; also $X \subseteq \textsc{Even}(G, M)$.

Lemma 7 *For any maximum matching M in G we have (i) $\textsc{Even}(G, M) = D(G)$ (ii) $\textsc{Odd}(G, M) = A(G)$ and (iii) $\textsc{Free}(G, M) = C(G)$.*

Proof: We prove the claims in order. If $v \in \textsc{Even}(G, M)$, let P be an even length M-alternating path from some $x \in X$ to v. Then, $M \Delta E(P)$ is another maximum matching in which v is exposed; hence, $v \in D(G)$. Conversely, if $v \in D(G)$ there is a maximum matching M_v that misses v. Then $M \Delta M_v$ gives an even length X-v M-alternating path implying that $v \in \textsc{Even}(G, M)$. Therefore, $\textsc{Even}(G, M) = D(G)$.

If $v \in \textsc{Odd}(G, M)$, let P be an X-v M-alternating path. Since $v \notin \textsc{Even}(G, M)$, P is of odd length and its last edge is uv where $u \in \textsc{Even}(G, M)$. Therefore v is a neighbor of $\textsc{Even}(G, M) = D(G)$ and $v \notin D(G)$ and hence $v \in A(G)$. Conversely, suppose $v \in A(G)$ and let $uv \in E$ where $u \in D(G) = \textsc{Even}(G, M)$. There is an M-alternating X-u path P of even length which ends in an edge $wu \in M$. If $v \in V(P)$ then clearly there is an X-v alternating path. Otherwise, $P + uv$ is an X-v alternating path ($wu \in M$, hence $uv \notin M$ unless $w = v$ but then $v \in V(P)$). Therefore $v \in \textsc{Odd}(G, M)$ since $v \notin D(G) = \textsc{Even}(G, M)$.

Finally, $C(G) = V \setminus (D(G) \cup A(G))$ and hence $\textsc{Free}(G, M) = C(G)$. □

Lemma 8 *Let M be any maximum matching in G, then each node in $A(G) \cup C(G)$ is covered by M and moreover every node $v \in A(G)$ is matched to some node in $D(G)$.*

Proof: From Lemma 7, $X \subseteq D(G)$ where X is the set of M-exposed nodes. Hence each node in $A(G) \cup C(G)$ is covered by M.

Suppose $u \in A(G)$ and $uv \in M$. Since $u \in \textsc{Odd}(G, M)$, there is an odd length X-v alternating path P which ends in an edge $wu \notin M$. If v is not in P then $P + uv$ is an M-alternating X-v path and hence $v \in \textsc{Even}(G, M) = D(G)$. If v is in P, let Q be the prefix of P till v, then $Q + vu$ is an even length M-alternating X-u path which contradicts the fact that $u \in A(G)$. □

Corollary 9 *Each component in $G[C(G)]$ is even and $|M \cap C(G)| = |C(G)|/2$.*

Proof: All nodes in $C(G)$ are covered by M. Since $A(G)$ separates $D(G)$ from $C(G)$, and $A(G)$ is matched only to $D(G)$ (by the above lemma), nodes in $C(G)$ are matched internally and hence the corollary follows. □

The main technical lemma is the following.

Lemma 10 *Let M be a maximum matching in G and X be the M-exposed nodes. Each component H of $G[D(G)]$ satisfies the following properties:*

1. *Either $|V(H) \cap X| = 1$ and $|M \cap \delta_G(V(H))| = 0$, or $|M \cap \delta_G(V(H))| = 1$.*
2. *H is factor-critical.*

Assuming the above lemma, we finish the proof of the theorem. Since each component of $G[D(G)]$ is factor-critical, it is necessarily odd. Hence, from Corollary 9 and Lemma 10, we have that $G[C(G)]$ contains all the even components of $G - A(G)$ and $G[D(G)]$ contains all the odd components of $G - A(G)$. We only need to show that $A(G)$ is a Tutte-Berge witness. To see this, consider any maximum matching M and the M-exposed nodes X. We need to show $|M| = \frac{1}{2}(|V| + |A(G)| - o(G - A(G)))$. Since $|M| = \frac{1}{2}(|V| - |X|)$, this is equivalent to showing that $|X| + |A(G)| = o(G - A(G))$. From Lemma 8, M matches each node in $A(G)$ to a node in $D(G)$. From Lemma 10, each odd component in $G[D(G)]$ either has a node in X and no M-edge to $A(G)$ or has exactly one M-edge to $A(G)$. Hence $|X| + |A(G)| = o(G - A(G))$ since all the odd components in $G - A(G)$ are in $G[D(G)]$.

We need the following proposition before the proof of Lemma 10.

Proposition 11 *Let M be a maximum matching in G. If there is an edge $uv \in G$ such that $u, v \in \text{EVEN}(G, M)$, then there is an M-flower in G.*

Proof Sketch. Let P and Q be even length M-alternating paths from X to u and v, respectively. If $uv \neq M$ then $P + uv + Q$ is an X-X alternating *walk* of odd length; since M is maximum, this walk has an M-flower. If $uv \in M$, then uv is the last edge of both P and Q and in this case $P - uv + Q$ is again an X-X alternating walk of odd length. $\square$

Proof of Lemma 10. We proceed by induction on $|V|$. Let M be a maximum matching in G and X be the M-exposed nodes. First, suppose $D(G)$ is a stable set (independent set). In this case, each component in $G[D(G)]$ is a singleton node and the lemma is trivially true.

If $G[D(G)]$ is not a stable set, by Proposition 11, there is an M-flower in G. Let B the M-blossom with the node b as the base of the stem. Recall that b has an even length M-alternating path from some node $x \in X$; by going around the odd cycle according to required parity, it can be seen that $B \subseteq \text{EVEN}(G, M) = D(G)$. Let $G' = G/B$ be the graph obtained by shrinking B. We identify the shrunken node with b. Recall from the proof of correctness of Edmonds algorithm that $M' = M/B$ is a maximum matching in G'. Moreover, the set of M'-exposed nodes in G' is also X (note that we identified the shrunken node with b, the base of the stem, which belong to X if the stem consists only of b). We claim the following with an informal proof.

Claim 12 *$D(G') = (D(G) \setminus B) \cup \{b\}$, and $A(G') = A(G)$ and $C(G') = C(G)$.*

Proof Sketch. We observed that X is the set of exposed nodes for both M and M'. We claim that $v \in \text{EVEN}(G', M')$ implies $v \in \text{EVEN}(G, M)$. Let P be an even length X-v M'-alternating path in G'. If it does not contain b then it is also an X-v even length M-alternating path in G. If P contains b, then one can obtain an even length X-v M-alternating path Q in G by expanding b into B and using the odd cycle B according to the desired parity. Conversely, let $v \in \text{EVEN}(G, M) \setminus B$ and let P be an X-v M-alternating path of even length in G. One can obtain an even length X-v M'-alternating path Q in G' as follows. If P does not intersect B then $Q = P$ suffices. Otherwise, we consider the first and last nodes of $P \cap B$ and shortcut P between them using the necessary parity by using the odd cycle B and the matching edges in there. Therefore, $D(G') = (D(G) \setminus B) \cup \{b\}$ and the other claims follow. $\square$

By induction, the components of $G' - A(G')$ satisfy the desired properties. Except for the component H_b that contains b, every other such component is also a component in $G - A(G)$. Therefore, it is not hard to see that it is sufficient to verify the statement for the component H in $G - A(G)$ that contains B which corresponds to H_b in $G' - A(G')$ that contains b. We note that

X is also the set of M'-exposed nodes in G' and since $\delta_G(H) \cap M = \delta_{G'}(H_b) \cap M'$ (B is internally matched by M except possibly for b), the first desired property is easily verified.

It remains to verify that H is factor-critical. By induction, H_b is factor-critical. Since H_b is obtained by shrinking an odd cycle in H, Proposition 4 show that H is factor-critical. $\square$

Algorithmic aspect: Given G, its Edmonds-Gallai decomposition can be efficiently computed by noting that one only needs to determine $D(G)$. A node v is in $D(G)$ iff $\nu(G) = \nu(G - v)$ and hence one can use the maximum matching algorithm to determine this. However, as the above proof shows, one can compute $D(G)$ in the same time it takes to find $\nu(G)$ via the algorithm of Edmonds, which has an $O(n^3)$ implementation. The proof also shows that given a maximum matching M, $D(G)$ can be obtained in $O(n^2)$ time.

1.2 Ear Decompositions and Factor-Critical Graphs

A graph H is obtained by *adding an ear* to G if H is obtained by adding to G a path P that connects two not-necessarily distinct nodes u, v in G. The path P is called an ear. P is a *proper* ear if u, v are distinct. An ear is an *odd* (even) ear if the length of P is odd (even). A sequence of graph $G_0, G_1, \ldots, G_k = G$ is an ear decomposition for G starting with G_0 if for each $1 \leq i \leq k$, G_i is obtained from G_{i-1} by adding an ear. One defines, similarly, proper ear decomposition and odd ear decomposition by restricting the ears to be proper and odd respectively.

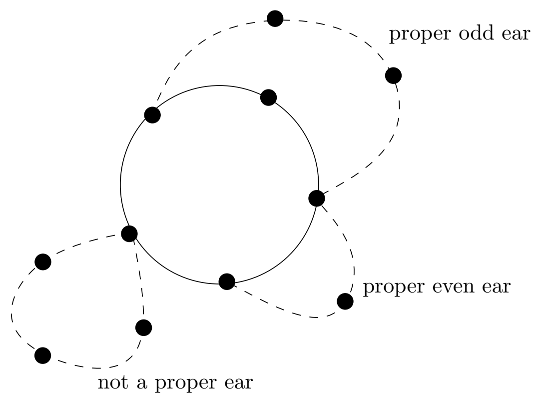


Figure 3: Variety of ears.

The following theorems are well-known and standard in graph theory.

Theorem 13 (Robbins, 1939) *A graph G is 2-edge-connected if and only if it has an ear-decomposition starting with a cycle.*

Theorem 14 (Whitney, 1932) *A graph G is 2-node-connected if and only if it has a proper ear-decomposition starting with a cycle.*

Factor-critical graphs have the following characterization.

Theorem 15 (Lovász, 1972) *A graph G is factor-critical if and only if it has an odd ear decomposition starting with a single vertex.*

Proof: If G has an odd ear decomposition it is factor-critical by inductively using Proposition 4 and noting that an odd cycle is factor-critical for the base case.

We now prove the converse. G is necessarily connected. Let v be an arbitrary vertex and let M_v be a perfect matching in $G - v$. We iteratively build the ear decomposition starting with the empty graph v. At each step we maintain a (edge-induced) subgraph H of G such that H has an odd ear decomposition and no edge $uv \in M_v$ crosses H (that is, $|V(H) \cap \{u, v\}| \neq 1$). The process stops when $E(H) = E(G)$. Suppose $E(H) \neq E(G)$, then since G is connected, there is some edge $ab \in E(G)$ such that $a \in V(H)$ and $b \neq V(H)$. By the invariant, $ab \notin M_v$. Let M_b be a perfect matching in G that misses b. Then $M_b \Delta M_v$ contains an even length M_v-alternating path $Q := u_0 = b, u_1, \ldots, u_t = v$ starting at b and ending at v. Let j be the smallest index such that $u_j \in V(H)$ (j exists since $u_t = v$ belongs to $V(H)$); that is u_j is the first vertex in H that the path Q hits starting from b. Then, by the invariant, $u_{j-1}u_j \notin M_v$ and hence j is even. The path $a, b = u_0, u_1, \ldots, u_j$ is of odd length and is a valid ear to add to H while maintaining the invariant. This enlarges H and hence we eventually reach G and the process generates an odd ear decomposition. □

One can extend the above proof to show that G is 2-node-connected and factor-critical iff it has an proper odd ear decomposition starting from an odd cycle.

From Proposition 4 and Theorem 15, one obtains the following.

Corollary 16 *G is factor-critical iff there is an odd cycle C in G such that G/C is factor-critical.*

References

[1] Lecture notes from Michel Goemans class on Combinatorial Optimization. `http://math.mit.edu/~goemans/18438/lec3.pdf`, 2009.

[2] A. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, 1998.

CS 598CSC: Combinatorial Optimization | **Lecture date: 2/16/2009**
Instructor: Chandra Chekuri | **Scribe: Vivek Srikumar**

1 Perfect Matching and Matching Polytopes

Let $G = (V, E)$ be a graph. For a set $E' \subseteq E$, let $\chi^{E'}$ denote the characteristic vector of E' in $\mathbb{R}^{|E|}$. We define two polytopes:

$$\mathcal{P}_{perfect\_matching}(G) = \textbf{convexhull}(\{\chi^M | M \text{ is a perfect matching in } G\})$$

$$\mathcal{P}_{matching}(G) = \textbf{convexhull}(\{\chi^M | M \text{ is a perfect in } G\})$$

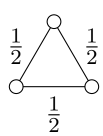
Edmonds gave a description of these polytopes. Recall that for bipartite graphs, $\mathcal{P}_{perfect\_matching}(G)$ is given by

$$\begin{aligned} x\left(\delta(v)\right) &= 1 \quad \forall v \in V \\ x(e) &\geq 0 \quad \forall e \in E \end{aligned}$$

and $\mathcal{P}_{matching}(G)$ is given by

$$\begin{aligned} x\left(\delta(v)\right) &\leq 1 \quad \forall v \in V \\ x(e) &\geq 0 \quad \forall e \in E \end{aligned}$$

We saw an example of a non-bipartite graph, for which $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is an optimum solution.



Hence, (perfect) matching polytope for non-bipartite graphs are not captured by the simple constraints that work for bipartite graphs.

Theorem 1 (Edmonds) *$\mathcal{P}_{perfect\_matching}(G)$ is determined by the following set of inequalities.*

$$\begin{aligned} x(e) &\geq 0; & e &\in E \\ x\left(\delta(v)\right) &= 1; & v &\in V \\ x\left(\delta(U)\right) &\geq 1; & U &\subseteq V, |U| \geq 3, |U| \text{ odd} \end{aligned}$$

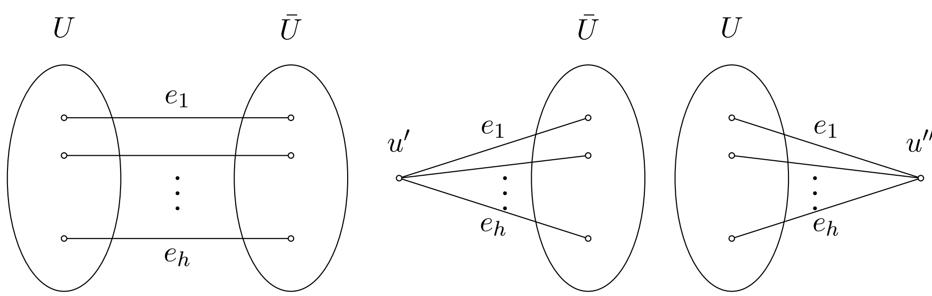
Edmonds gave a proof via an algorithmic method. In particular, he gave a primal-dual algorithm for the minimum cost perfect matching problem, which, as a by product showed that for any cost vector c on the edges, there is a minimum cost perfect matching whose cost is equal the minimum value of cx subjecto to the above set of inequalities. This implies that the polytope is integral. We describe a short non-algorithmic proof that was given later [1] (Chapter 25).

Proof: Let $Q(G)$ denote the polytope described by the inequalities in the theorem statement. It is easy to verify that for each graph G, $\mathcal{P}_{perfect\_matching}(G) \subseteq Q(G)$. Suppose there is a graph G such that $Q(G) \not\subseteq \mathcal{P}_{perfect\_matching}(G)$. Among all such graphs, choose the one that minimizes $|V|+|E|$. Let G be this graph. In particular, there is a basic feasible solution (vertex) x of $Q(G)$ such that x is not in $\mathcal{P}_{perfect\_matching}(G)$.

We claim that $x(e) \in (0,1)$; $\forall e \in E$. If $x(e) = 0$ for some e, then deleting e from G gives a smaller counter example. If $x(e) = 1$ for some e, then deleting e and its end points from G gives a smaller counter example.

We can assume that $|V|$ is even, for otherwise $Q(G) = \emptyset$ (why?) and $\mathcal{P}_{perfect\_matching}(G) = \emptyset$ as well. Since $0 < x(e) < 1$ for each e and $x(\delta(v) = 1$ for all v, we can assume that $deg(v) \geq 2$; $\forall v \in V$. Suppose $|E| = 2|V|$. Then $deg(v) = 2$; $\forall v \in V$ and therefore, G is a collection of vertex disjoint cycles. Then, either G has an odd cycle in its collection of cycles, in which case, $Q(G) = \emptyset = \mathcal{P}_{perfect\_matching}(G)$, or G is a collection of even cycles and, hence bipartite and $Q(G) = \mathcal{P}_{perfect\_matching}(G)$.

Thus $|E| > 2|V|$. Since x is a vertex of $Q(G)$, there are $|E|$ inequalities in the system that are tight and determine x. Therefore, there is some odd set $U \subsetneq V$ such that $x(\delta(U)) = 1$. Let $G' = G/U$, where U is shrunk to a node, say u'. Define $G'' = G/\bar{U}$, where $\bar{U} = V - U$ is shrunk to a node u''; see Figure 1. The vector x when restricted to G' induces $x' \in Q(G')$ and similarly x induces



(a) G (b) G' (c) G''

Figure 1:

$x'' \in Q(G'')$. Since G' and G'' are smaller than G, we have that $Q(G') = \mathcal{P}_{perfect\_matching}(G')$ and $Q(G'') = \mathcal{P}_{perfect\_matching}(G'')$. Hence, x' can be written as a convex combination of perfect matchings in G' and x'' can be written as a convex combination of perfect matchings in G''. The vector x is rational since we chose it as a vertex of $Q(G)$, therefore, x', x'' are also rational; hence, $\exists$ integer k such that $x' = \frac{1}{k}\sum_{i=1}^{k} \chi^{M_i'}$, where $M_1', M_2', \cdots, M_k'$ are perfect matchings in G' and $x'' = \frac{1}{k}\sum_{i=1}^{k} \chi^{M_i''}$, where $M_1'', M_2'', \cdots, M_k''$ are perfect matchings in G''. (Note that k is the same in both expressions.)

Let $e_1, e_2, \cdots, e_h$ be edges in $\delta(U)$. Since $x'(\delta(u') = 1$ and u' is in every perfect matching, we have that e_j is in exactly $kx'(e_j) = kx(e_j)$ matchings of $M_1', \cdots, M_k'$. Similarly, e_j is in exactly $kx(e_j)$ matchings of $M_1'', \cdots, M_k''$. Note that $\sum_{j=1}^{n} kx(e_j) = k$ and moreover, exactly one of $e_1, \cdots, e_h$ can be in M_i' and M_i''. We can, therefore, assume (by renumbering if necessary) that M_i' and M_i'' share exactly one edge from $e_1, \cdots, e_h$. Then, $M_i = M_i' \cup M_i''$ is a perfect matching in G. Hence, $x = \frac{1}{k}\sum_{i=1}^{k} \chi^{M_i}$, which implies that $x \in \mathcal{P}_{perfect\_matching}(G)$, contradicting our assumption. $\square$

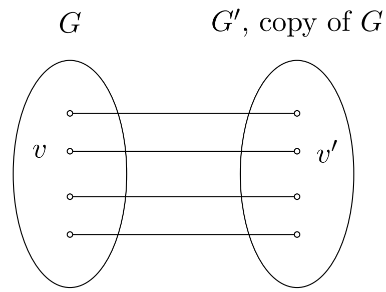
Now, we use the above theorem to derive the following:

Theorem 2 *$\mathcal{P}_{matching}(G)$ is determined by*

$$\begin{aligned} x(e) &\geq 0; && e \in E \\ x\left(\delta(v)\right) &\leq 1; && v \in V \\ x\left(E[U]\right) &\leq \frac{|U|-1}{2}; && U \subseteq V, |U| \text{ odd} \end{aligned}$$

Here $E[U]$ is the set of edges with both end points in U.

Proof: We can use a reduction of weighted matching to weighted perfect matching that is obtained as follows: Given $G = (V, E)$, create a copy $G' = (V', E')$ of G. And let $\tilde{G}$ be the graph $(\tilde{V}, \tilde{E})$ defined as $\tilde{V} = V \cup V'$, $\tilde{E} = E \cup E' \cup \{(v, v') \mid v \in V\}$.



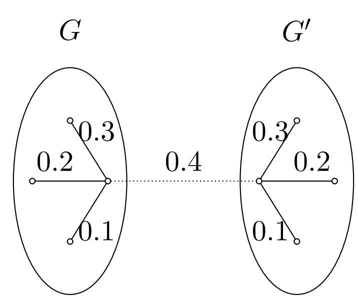
The following claim is easy to prove.

Claim 3 *There is an injective mapping from the matchings in G to the perfect matchings in $\tilde{G}$.*

Corollary 4 *The maximum weight matching problem is poly-time equivalent to maximum weight perfect matching problem.*

We can use the above idea to establish the theorem. Let x be feasible for the system of inequalities in the theorem. We show that x can be written a convex combination of matchings in G. It is clear that χ^M satisfies the inequalities for every matching M. From G, create $\tilde{G}$ as above and define a fractional solution $\tilde{x} : \tilde{E} \to \mathbb{R}^+$ as follows: first, we define $x' : E' \to \mathbb{R}^+$ as the copy of x on E. That is, $x'(e') = x(e)$, where e' is the copy of e. Then,

$$\tilde{x} = \begin{cases} x(e); \text{ if } e \in E \\ x'(e); \text{ if } e \in E' \\ 1 - x\left(\delta(v)\right); \text{ if } e = vv' \end{cases}$$

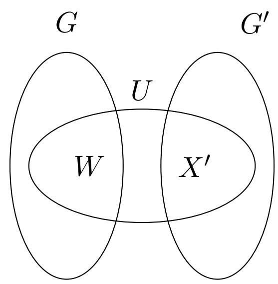


Claim 5 *$\tilde{x}$ belongs to $\mathcal{P}_{perfect\_matching}(\tilde{G})$.*

Assuming the claim, we see that $\tilde{x}$ can be written as a convex combination of perfect matchings in $\tilde{G}$. Each perfect matching in $\tilde{G}$ induces a matching in G and it is easy to verify that x can therefore be written as a convex combination of matchings in G.

It only remains to verify the claim. From the previous theorem, it suffices to show that

$$\tilde{x}\left(\tilde{\delta}(U)\right) \geq 1; \quad \forall U \subseteq \tilde{V}, |U| odd$$

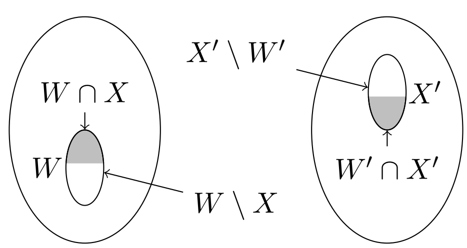


Let $U \subseteq \tilde{V}$ and $|U|$ odd. Let $W = U \cap V$ and $X' = U \cap V'$, where X' is the copy of $X \subseteq V$. First we consider the case that $X' = \emptyset$ and $|W|$ is odd. Then

$$\begin{aligned}
\tilde{x}\left(\tilde{\delta}(U)\right) &= \tilde{x}\left(\tilde{\delta}(W)\right) \\
&= \sum_{v \in W} \tilde{x}\left(\tilde{\delta}(v)\right) - 2\tilde{x}\left(E[W]\right) \\
&= |W| - 2x\left(E[W]\right) \\
&\geq |W| - 2\left(\frac{|W|-1}{2}\right) \\
&\geq 1
\end{aligned}$$

For the general case, we claim that $\tilde{x}\left(\tilde{\delta}(U)\right) \geq \tilde{x}\left(\tilde{\delta}(W \setminus X)\right) + \tilde{x}\left(\tilde{\delta}(X' \setminus W')\right)$. Without loss of generality, $W \setminus X$ is odd. Then $\tilde{x}\left(\tilde{\delta}(U)\right) \geq \tilde{x}\left(\tilde{\delta}(W \setminus X)\right) \geq 1$ from above.

The claim can be verified as follows:



Notice that only edges between W and X' are between $W \cap X$ and $X' \cap W'$. Let $A = W \cap X$, $A' =$

$W' \cap X'$. Then

$$\begin{aligned}
\tilde{x}\left(\tilde{\delta}(U)\right) &= \tilde{x}\left(\tilde{\delta}(W \cup X')\right) \\
&= \tilde{x}\left(\tilde{\delta}(W \setminus X)\right) + \tilde{x}\left(\tilde{\delta}(X' \setminus W')\right) + \\
&\quad x\left(\delta(A)\right) - 2x((\delta\left(E[A, W \setminus X]\right)) + \\
&\quad x\left(\delta'(A')\right) - 2x\left(\delta'\left(E[A', X' \setminus W']\right)\right)
\end{aligned}$$

The claim follows from the observation that $x(\delta(A)) \geq x\left(E[A, W \setminus A]\right) + x\left(\delta(E[A, X \setminus W]\right)$. □

Corollary 6 *$\mathcal{P}_{perfect\_matching}(G)$ is also determined by*

$$\begin{aligned}
x(e) &\geq 0; & e &\in E \\
x\left(\delta(v)\right) &= 1; & v &\in V \\
x\left(E[U]\right) &\leq \frac{|U|-1}{2}; & U &\subseteq V, |U| \textit{ odd}
\end{aligned}$$

We note that although the system in the above corollary and the earlier theorem both determine $\mathcal{P}_{perfect\_matching}(G)$, they are not identical.

2 Separation Oracle for Matching Polytope

The inequality systems that we saw for $\mathcal{P}_{perfect\_matching}(G)$ and $\mathcal{P}_{matching}(G)$ have an exponential number of inequalities. Therefore, we cannot use them directly to solve the optimization problems of interest, namely, the maximum weight matching problem or the minimum weight perfect matching problem. To use the Ellipsoid method, we need a polynomial time separation oracle for the polytopes. Edmonds gave efficient strongly polynomial time algorithms for optimizing over these polytopes. From the equivalence of optimization and separation (via the ellipsoid method) implies that there is a strongly polynomial time separation oracle. However, the oracle obtained via the above approach is indirect and cumbersome. Padberg and Rao [1982] gave a simple and direct separation oracle. We discuss this for the system

$$\begin{aligned}
x(e) &\geq 0; & e &\in E \\
x\left(\delta(v)\right) &= 1; & v &\in V \\
x\left(\delta(U)\right) &\geq 1; & |U| &\text{ odd}, U \subseteq V
\end{aligned} \tag{1}$$

and it can be used with the reduction shown earlier for the matching polytope as well.

Theorem 7 *There is a strongly polynomial time algorithm, that given $G = (V, E)$ and $x : E \to \mathbb{R}$ determines if x satisfies (1) or outputs an inequality from (1) that is violated by x.*

It is trivial to check the first two sets of inequalities. Therefore, we assume that $x \geq 0$ and $x\left(\delta(v)\right) = 1; \forall v \in V$. We can also assume that $|V|$ is even. Thus the question is whether there is a set $U \subset V, |U|$ odd, such that $x\left(\delta(U)\right) < 1$. It is sufficient to give an algorithm for the *minimum odd-cut problem*, which is the following: Given a capacitated graph $G = (V, E)$, find a cut $\delta(U)$ of minimum capacity among all sets U such that $|U|$ is odd. We claim that the following is a correct algorithm for the minimum odd-cut problem.

1. Compute a Gomory-Hu tree $T = (V, E_T)$ for G.

2. Among the odd-cuts induced by the edges of T, output the one with the minimum capacity.

To see the correctness of the algorithm, let $\delta(U^*)$ be a minimum capacity odd cut in G. Then $\delta_T(U^*)$ is a set of edges in E_T. We claim that there is an edge $st \in \delta_T(U^*)$ such that $T - st$ has a component with an odd number of nodes. If this is true, then, by the prperties of the Gomory-Hu tree, $T - st$ induces an odd cut in G of capacity equal to $\alpha_G(s,t)$ (recall that $\alpha_G(s,t)$ is the capacity of a minimum s-t cut in G). Since $\delta(U^*)$ separates s and t, the odd cut induced by $T - st$ has no larger capacity than $\delta(U^*)$. We leave it as an exercise to show that some edge in $\delta_T(U^*)$ induces an odd-cut in T.

3 Edge Covers and Matchings

Given $G = (V, E)$ an edge cover is the subset $E' \subset E$ such that each node is covered by some edge in E'. This is the counterpart to vertex cover. Edge covers are closely related to matchings and hence optimization problems related to them are tractable, unlike the vertex cover problem whose minimization version is NP-Hard.

Theorem 8 (Gallai) *Let $\rho(G)$ be the cardinality of a maximum size edge cover in G. Then*

$$\nu(G) + \rho(G) = |V|$$

where $\nu(G)$ is the cardinality of a maximum matching in G.

Proof: Take any matching M in G. Then M covers $2|M|$ nodes, the end points of M. For each such uncovered node, pick an arbitrary edge to cover it. This gives an edge cover of size $\leq |V| - 2|M| + |M| \leq |V| - |M|$. Hence $\rho(G) \leq |V| - \nu(G)$.

We now show that $\nu(G) + \rho(G) \geq |V|$. Let E' be any inclusion-wise minimal edge cover and let M be an inclusion-wise maximal matching in E'. If v is not incident to an edge of M then since it is covered by E' there is an edge $e_v \in E' \setminus M$ that covers v; since M is maximal the other end point of e_v is covered by M. This implies that $2|M| + |E' \setminus M| \geq |V|$, that is $2|M| + |E'| - |M| \geq |V|$ and hence $|M| + |E'| \geq |V|$. If E' is a minimum edge cover then $|E'| = \rho(G)$ and $|M| \leq \nu(G)$, therefore, $\nu(G) + \rho(G) \geq |V|$. $\square$

The above proof gives an efficient algorithm to compute $\rho(G)$ and also a minimum cardinality edge cover via an algorithm for maximum cardinality matching. One can define the minimum weight edge cover problem and show that this also has a polynomial time algorithm by reducing to matching problems/ideas. The following set of inequalities determine the edge cover polytope (the convex hull of the characterstic vectors of edge covers in G).

$$\begin{array}{rcl} x\left(\delta(V)\right) \geq & 1 & \forall v \in V \\ x\left(E[U] \cup \delta(U)\right) \geq & \frac{|U|+1}{2} & U \subseteq V; |U| odd \\ 0 \leq x(e) \leq 1; & & e \in E \end{array}$$

Exercise 9 *Prove that the polytope above is the edge cover polytope and obtain a polynomial time separation oracle for it.*

References

[1] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, Springer, 2003.

CS 598CSC: Combinatorial Optimization	**Lecture date: Feb 18, 2010**
Instructor: Nitish Korula	**Scribe: Abner Guzmán-Rivera**

1 Maximum Weight Matching in Bipartite Graphs

In these notes we consider the following problem:

Definition 1 (Maximum Weight Bipartite Matching) *Given a bipartite graph $G = (V, E)$ with bipartition (A, B) and weight function $w : E \to \mathbb{R}$ find a matching of maximum weight where the weight of matching M is given by $w(M) = \sum_{e \in M} w(e)$.*

Note that without loss of generality, we may assume that G is a *complete* weighted bipartite graph (we may add edges of zero weight as necessary); we may also assume that G is balanced, i.e. $|A| = |B| = \frac{1}{2}|V|$, as we can add dummy vertices as necessary. Hence, some maximum weight matching is a perfect matching. Furthermore, by negating the weights of the edges we can state the problem as the following *minimization* problem:

Definition 2 (Minimum Weight Perfect Matching in Bipartite Graphs) *Given a bipartite graph $G = (V, E)$ with bipartition (A, B) and weight function $w : E \to \mathbb{R} \cup \{\infty\}$, find a perfect matching M minimizing $w(M) = \sum_{e \in M} w(e)$.*

We could also assume that no edge weights are negative as we may add a large enough constant C to all weights, but this is not required by the algorithms below.

The following is an ILP formulation of the minimum weight perfect matching problem:

$$\min \sum_{(a,b)} w(a,b)x(a,b) \text{ subject to:}$$

$$\begin{aligned}
\sum_b x(a,b) &= 1 \qquad \forall a \in A \\
\sum_a x(a,b) &= 1 \qquad \forall b \in B \\
x(a,b) &\in \{0,1\} \quad \forall a \in A, b \in B
\end{aligned} \tag{1}$$

Definition 3 (Primal) *This is the LP relaxation of the above ILP:*

$$\min \sum_{(a,b)} w(a,b)x(a,b) \text{ subject to:}$$

$$\begin{aligned}
\sum_b x(a,b) &= 1 \quad \forall a \in A \\
\sum_a x(a,b) &= 1 \quad \forall b \in B \\
x(a,b) &\geq 0 \quad \forall a \in A, b \in B
\end{aligned} \tag{2}$$

Recall that we saw, in an earlier lecture, a proof of the following theorem by noting that the constraint matrix of the polytope is totally unimodular.

Theorem 4 *Any extreme point of the polytope defined by the constraints in (2) is integral.*

We obtain a different proof of Theorem 4 via algorithms to find a minimum-weight perfect matching. Our algorithms are *primal-dual*; we will construct a feasible solution to the dual of LP (2) with value equal to the weight of the perfect matching output by the algorithm. By weak duality, this implies that the matching is optimal. More precisely, our algorithms will always maintain a feasible dual solution y, and will attempt to find a primal feasible solution (a perfect matching M) that satisfies complementary slackness.

(Dual) The following LP is the dual for (2):

$$\begin{aligned} \textit{maximize} \sum_{(a \in A} y(a) + \sum_{b \in B} y(b) \text{ subject to:} \\ y(a) + y(b) \quad \leq \quad w(a,b) \quad \forall (a,b) \in E \end{aligned} \tag{3}$$

Given a dual-feasible solution y, we say that an edge $e = (a,b)$ is *tight* if $y(a) + y(b) = w(a,b)$. Let $\hat{y}$ be dual-feasible, and let M be a perfect matching in $G(V,E)$: Then,

$$\begin{aligned} w(M) = \sum_{(a,b) \in M} w(a,b) \quad &\geq \quad \sum_{(a,b) \in M} \hat{y}(a) + \hat{y}(b) \\ &= \quad \sum_{a \in A} \hat{y}(a) \cdot (\delta(a) \cap M) + \sum_{b \in B} \hat{y}(b) \cdot (\delta(b) \cap M) \\ &= \quad \sum_{a \in A} \hat{y}(a) + \sum_{b \in B} \hat{y}(b) \end{aligned}$$

where the first inequality follows from the feasibility of $\hat{y}$, and the final equality from the fact that M is a perfect matching. That is, any feasible primal solution (a perfect matching M) has weight at least as large as the value of any feasible dual solution. (One could conclude this immediately from the principle of weak duality.) Note, though, that if M only uses edges which are *tight* under $\hat{y}$, we have equality holding throughout, and so by weak duality, M must be *optimal*. That is, given any dual feasible solution $\hat{y}$, if we can find a perfect matching M only using tight edges, M must be optimal. (Recall that this is the principle of *complementary slackness*.)

Our primal-dual algorithms apply these observations as follows: We begin with an arbitrary feasible dual solution y, and find a maximum-cardinality matching M that uses only tight edges. If M is perfect, we are done; if not, we *update* our dual solution. This process continues until we find an optimal solution.

We first give a simple algorithm (Algorithm 1 in the following page) exploiting these ideas to prove Theorem 4. The existence of set S in line 6 is a consequence of Hall's theorem. Observe that the value of y increases at the end of every iteration. Also, the value of y remains feasible as tight edges remain tight and it is easy to verify that by the choice of ϵ the constraints for other edges are not violated.

Claim 5 *Algorithm 1 terminates if w is rational.*

Proof: Suppose all weights in w are integral. Then at every iteration ϵ is integral and furthermore $\epsilon \geq 1$. It follows that the number i of iterations is bounded by $i \leq \max w(a,b) \cdot |E|$. If weights are rational we may scale them appropriately so that all of them become integers. □

Algorithm 1 MinWeightPerfectMatching($G = (V, E), w$)

$y \leftarrow 0$
$E' \leftarrow$ set of tight edges
$M \leftarrow$ max cardinality matching for graph $G' = (V, E')$
while M is not a perfect matching **do**
  let $G' = (V, E')$
  let $S \subseteq A$ be such that $|S| > |N(S)|$
  let $\epsilon = \min_{a \in S, b \in B \setminus N(S)} \{w(a, b) - y(a) - y(b)\}$
  $\forall a \in S \; y(a) = y(a) + \epsilon$
  $\forall b \in N(S) \; y(b) = y(b) - \epsilon$
  update E', M
end while
return M

Proof of Theorem 4. The incidence vector of a perfect matching computed by Algorithm 1 is an extreme point of the polytope in (2). This vector is integral. Furthermore, by carefully choosing the cost function one can make any extreme point be the unique optimum solution to the primal linear program. □

Note that Algorithm 1 does not necessarily terminate in strongly polynomial time; in the rest of this section, we describe a more efficient algorithm for the minimum-weight bipartite matching problem.

As before, Algorithm 2 always maintains a feasible dual y and attempts to find a close to primal feasible solution (matching M) that satisfies complementary slackness. One key difference from Algorithm 1 is that we now carefully use the maximum cardinality matching M as a guide in constructing the updated dual solution y; this allows us to argue that we can augment M efficiently. (In contrast, Algorithm 1 effectively "starts over" with a new matching M in each iteration.)

Algorithm 2 MinWeightPerfectMatchingPD($G = (V, E), w$)

$\forall b \in B \; y(b) \leftarrow 0$
$\forall a \in A \; y(a) \leftarrow \min_b \{w(a, b)\}$
$E' \leftarrow$ set of tight edges
$M \leftarrow$ max cardinality matching for graph $G' = (V, E')$
while M is not a perfect matching **do**
  let $E_{dir} \leftarrow \{e$ directed from A to $B \mid e \in E', e \notin M\} \bigcup$
    $\{e$ directed from B to $A \mid e \in E', e \in M\}$
  let $D = (V, E_{dir})$ $\{D$ is a directed graph$\}$
  let $L \leftarrow \{v \mid v$ is reachable in D from an unmatched vertex in $A\}$
  let $\epsilon = \min_{a \in A \cap L, b \in B \setminus L} \{w(a, b) - y(a) - y(b)\}$
  $\forall a \in A \cap L \; y(a) = y(a) + \epsilon$
  $\forall b \in B \cap L \; y(b) = y(b) - \epsilon$
  update E', M
end while
return M

Claim 6 *At every iteration, $C = (A \setminus L) \cup (B \cap L)$ is a vertex cover for graph $G' = (V, E')$. Moreover, $|C| = |M|$.*

Proof: Assume C is not a vertex cover. Then there must be an edge $e = (a, b) \in E'$ with $a \in A \cap L$ and $b \in B \setminus L$. If e is directed from a to b, then since a is reachable from an unmatched vertex in A, so is b; this contradicts the fact that $b \in B \setminus L$. Therefore, e must be directed from b to a, and hence e is in the matching M. As a itself is matched (using edge e) and $a \in L$, it must be reachable from an unmatched vertex of A. But the only incoming edge to a is (b, a) (this is the unique edge incident to a in the matching M), and hence b is reachable from this unmatched vertex of A; again, this contradicts the fact that $b \notin L$. To show the second part of the proof we show that $|C| \leq |M|$, since the reverse inequality is true for any matching and any vertex cover. The proof follows from the following observations:

1. No vertex in $A \setminus L$ is unmatched by the definition of L.
2. No vertex in $B \cap L$ is unmatched since this would imply the existence of an augmenting path (contradicting the maximality of M).
3. There is no edge $e = (a, b) \in M$ such that $a \in A \setminus L$ and $b \in B \cap L$. Otherwise, as this edge would be directed from b to a, a would be in L.

These remarks imply that every vertex in C is matched and moreover the corresponding edges of the matching are distinct. Hence $|C| \leq |M|$, and so C is an optimum vertex cover for $G'(V, E')$. $\square$

At every iteration where the maximum cardinality matching M output is not perfect, the algorithm will use information from the optimum vertex cover C to update the dual solution and improve its value. By the proof of claim 6 there is no tight edge between $a \in A \cap L$ and $b \in B \setminus L$, which implies $\epsilon > 0$; it is easy to check that the updated dual solution is feasible. Moreover, the difference between the new dual solution and the old dual solution is:

$$\epsilon \cdot (|A \cap L| - |B \cap L|) = \epsilon \cdot (|A \cap L| + |A \setminus L| - |A \setminus L| - |B \cap L|) = \epsilon \cdot (\frac{|V|}{2} - |C|),$$

but $|C| = |M| < \frac{|V|}{2}$, since M is not perfect, which implies the value of the dual solution strictly increases. When the algorithm terminates, we obtain a perfect matching M and a dual feasible solution which satisfy complementary slackness.

Claim 7 *Algorithm (2) terminates in $O(|V|^2)$ iterations.*

Proof: We first observe that after any iteration, all edges in M are still tight: The only edges (a, b) that are tight at the beginning of an iteration but not at the end are those with $a \in A \cap L$ and $b \in B \setminus L$; from observation 3 in the proof of Claim 6, there are no edges in M of this form. Thus, after any iteration, the size of a maximum cardinality matching M in $G'(V, E')$ cannot decrease.

Say that an iteration is *successful* if the size of a maximum cardinality matching using the tight edges E' increases. Clearly, after at most $|V|/2$ successful iterations, we have a perfect matching, and the algorithm terminates. We show that there are at most $|B| = |V|/2$ consecutive unsuccessful iterations between any pair of successful iterations. Hence, the total number of iterations is at most $\frac{|V|}{2} \cdot \frac{|V|}{2}$, which is $O(|V|^2)$.

To bound the number of consecutive unsuccessful iterations, we argue below that after an unsuccessful iteration, $|B \cap L|$ increases. Assume for now that this is true: After at most $|B|$ unsuccessful iterations, we have $B \cap L = B$. Once this occurs, every vertex of B (which must include at least one unmatched vertex) is reachable from an unmatched vertex of A, and so we can augment M to find a larger matching, which means that the current iteration is successful.

It remains only to prove that at every unsuccessful iteration, at least one more vertex in B must become reachable from an exposed vertex in A (i.e. $|B \cap L|$ increases). First note that no vertex of A or B becomes unreachable; the only way this could happen is if for some path P from an unmatched vertex $a \in A$ to vertex $v \in L$, an edge $e \in P$ that was previously tight is no longer tight. But the only edges that are no longer tight are between $A \setminus L$ and $B \cap L$, and by definition, no such path P visits a vertex in $A \setminus L$. To see that at least one new vertex of B becomes reachable, note that some edge $e = (a, b)$ with $a \in A \cap L$ and $b \in B \setminus L$ now has become tight by our choice of ϵ. As the edge (a, b) is directed from a to b, b is now reachable. $\square$

It is not hard to see that each iteration takes only $O(|V|^2)$ time, and hence the overall running time of the algorithm is $O(|V|^4)$. A more careful analysis would yield a tighter running time of $O(|V|^3)$.

References

[1] A. Schrijver. *Combinatorial optimization: Polyhedra and Efficiency*, Springer, 2003. Chapter 17.

[2] Lecture notes from Michael Goemans class on Combinatorial Optimization. `http://math.mit.edu/~goemans/18433S09/matching-notes.pdf`, 2009.

CS 598CSC: Combinatorial Optimization	**Lecture date: Feb 23, 2010**
Instructor: Chandra Chekuri	**Scribe: Rajhans Samdani**

1 Min Cost Perfect Matching

In this lecture, we will describe a strongly polynomial time algorithm for the minimum cost perfect matching problem in a **general** graph. Using a simple reduction discussed in lecture 9, one can also obtain an algorithm for the maximum weight matching problem. We also note that when discussing perfect matching, without loss of generality, we can assume that all weights/costs are non-negative (why?).

The algorithm we describe is essentially due to Edmonds. The algorithm is *primal-dual* based on the following LP formulation and its dual.

Primal:

$$\begin{array}{rcll} \min \sum_{e \in E} w(e)x(e) & & & \text{subject to:} \\ x(\delta(v)) & = & 1 & \forall v \in V \\ x(\delta(U)) & \geq & 1 & \forall U \subset V, |U| \geq 3, |U| odd \\ x(e) & \geq & 0 & \forall e \in E \end{array} \tag{1}$$

Dual:

$$\begin{array}{rcll} \max \sum_{U \subseteq V, |U| \geq 3, |U| odd} \pi(U) & & & \text{subject to:} \\ \sum_{U \subseteq V, |U| \geq 3, |U| odd, \delta(U) \ni e} \pi(U) & = & w(e) & \forall e \in E \\ \pi(U) & \geq & 0 & \forall U \subseteq V, |U| \geq 3, |U| odd \end{array}$$

We note that non-negativity constraints on the dual variables are only for odd sets U that are not singletons (because the equations for the singleton sets are equalities). In certain descriptions of the algorithm and details, the dual variables for the singleton are distinguished from the odd sets of size ≥ 3, however we don't do that here.

Like other primal dual algorithms, we maintain a feasible dual solution π and an integral infeasible primal solution x and iteratively reduce the infeasiblity of x. Here x corresponds to a matching and we wish to drive it towards a maximum matching. In particular, we will also maintain the primal complementary slackness, that is,

$$x(e) > 0 \Rightarrow \sum_{U : e \in \delta(U)} \pi(U) = w(e)$$

(a primal variable being positive implies the corresponding dual constraint is tight). Thus, at the end, if we have a perfect matching in the primal, it is feasible and certifies its optimality.

The main question is how to update the dual and the primal. Also, we observe that the dual has an exponential number of variables and hence any polynomial time algorithm can only maintain an implicity representation of a subset of the variables.

1.1 Notation

The algorithm maintains a *laminar family* of odd subsets of V denoted by Ω. Ω always includes the singletons $\{v\}, v \in V$. It maintains the invariant that $\pi(U) = 0$ if $U \notin \Omega$, hence Ω is the implicit representation of the "interesting" dual variables. Note that $|\Omega| \leq 2|V|$.

Given G, Ω and $\pi : \Omega \to \mathbb{R}$ where π is dual feasible, we say an edge is π-*tight* (or *tight* when π is implicit) if $\sum_{U \in \Omega : e \in \delta(U)} \pi(U) = w(e)$.

Let E_π be the set of tight edges (with Ω, π implicit) and G_π be the graph induced by them. We obtain a new graph G' in which we contract each maximal set in Ω into a (pseudo) vertex. For a node $v \in G'$, let $S_v \in \Omega$ be the set of nodes of G contracted to v.

For each $U \in \Omega$, $|U| \geq 3$, consider the graph $G_\pi[U]$ and let H_U be the graph obtained from $G_\pi[U]$ by contracting each maximal *proper* subset $S \subset U$ where $S \in \Omega$. The algorithm also maintains the invariant that H_U has a Hamiltonian cycle B_U.

1.2 Recap on Edmonds Gallai Decomposition

We restae the Edmonds Gallai decomposition and make some observations which help us in proposing and analysing an algorithm for min cost perfect matching. We also use the notation from this section in subsequent sections.

Theorem 1 (Edmonds-Gallai) *Given a graph $G = (V, E)$, let*

$$\begin{aligned} D(G) &:= \{v \in V \mid \text{there exists a maximum matching that misses } v\} \\ A(G) &:= \{v \in V \mid v \text{ is a neighbor of } D(G) \text{ but } v \notin D(G)\} \\ C(G) &:= V \setminus (D(G) \cup A(G)). \end{aligned}$$

Then, the following hold.

1. *The set $U = A(G)$ is a Tutte-Berge witness set for G.*
2. *$C(G)$ is the union of the even components of $G - A(G)$.*
3. *$D(G)$ is the union of the odd components of $G - A(G)$.*
4. *Each component in $G - A(G)$ is factor-critical.*

It is also easy to see the following remark.

Remark 2 *Let M be a maximum matching in G. Then either there exist an $M-$blossom or $D(G)$ consists of singleton nodes.*

1.3 Algorithm

Following is the algorithm for min cost perfect matching using primal dual method.
Initialize: $\Omega = \{\{v\} | v \in V\}, \pi(U) = 0 \ \forall U$ with odd $|U|, M = \phi$, and $G' = G_\pi$.
while(M is not a perfect matching in G') **do**

1. $X \leftarrow M-$exposed nodes in G'.
2. Find $X - X, M-$alternating walk P in G'.

3. If P is an $M-$augmenting path then do

$$M \leftarrow M \Delta E(P)$$

continue.

4. If P has an $M-$blossom B, then do *shrinking* as:

$$U = \cup_{v \in B} S_v, \Omega \leftarrow \Omega \cup U, \pi(U) = 0, G' \leftarrow G'/B, M \leftarrow M/B$$

continue.

5. Else P is empty $\Rightarrow M$ is a maximum matching in G'. Compute $D(G'), A(G')$, and $C(G')$ as in *Edmonds-Gallai decomposition*. Let ϵ be the largest value such that $\pi(S_v) = \pi(S_v) + \epsilon, \forall v \in D(G')$ and $\pi(S_v) = \pi(S_v) - \epsilon, \forall v \in A(G')$ maintains feasibility of π in G.

   If ϵ is unbounded then G has no perfect matching; **STOP**.

   If $\pi(S_v) = 0$ for some $v \in A(G')$ and $|S_v| \geq 3$, then *deshrink* as

   - Remove S_v from Ω.
   - Update G_π and G'.
   - Extend M by a perfect matching in $B_{S_v} - \{v\}$.

end while

Extend M in G' to a perfect matching in G_π and output it.

Example: Consider the execution of this algorithm on the following graph:

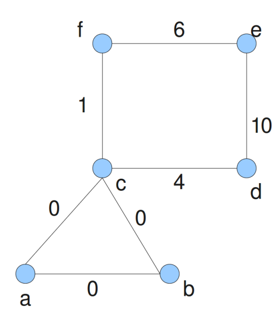


Figure 1: Original graph G

The execution is shown in figures 2 to 9. Red edges are the current edges in the matching; black edges are tight.

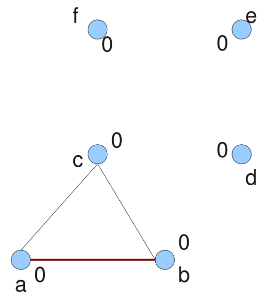


Figure 2: G_π after iteration 1.

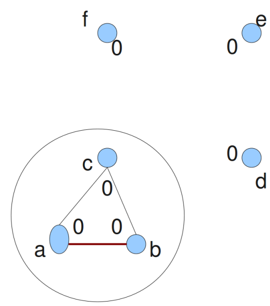


Figure 3: G_π after iteration 2. Shrinking.

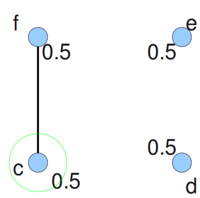


Figure 4: G_π after iteration 3. Edge tight.

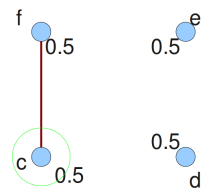


Figure 5: G_π after iteration 4. Augment.

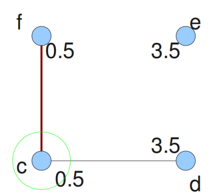


Figure 6: G_π after iteration 5. Edge tight.

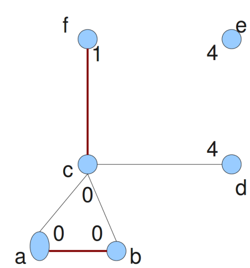


Figure 7: G_π after iteration 6. Deshrink.

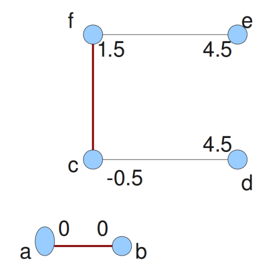


Figure 8: G_π after iteration 7. Edge tight. Some edges become slack and hence disappear from G_π.

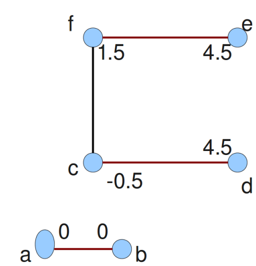


Figure 9: G_π after iteration 8. Augment. Maximum hence STOP.

1.4 Proof

Lemma 3 *The algorithm maintains the following invariants over the iterations*

- π *is dual feasible*
- Ω *is laminar*
- *for each* $U \in \Omega, H_U$ *has a hamiltonian cycle* B_U.

Proof Sketch. We need to check that each iteration maintains the given properties. So we do an analysis of all the considered cases and see that in each case, this property is preserved.
If M is augmented then Ω and π don't change.
If we shrink a blossom B in finding P then we add $U = \cup_{u \in B} S_v$ to Ω. This preserves laminarity since nodes in G' correspond to the maximal sets in Ω. Since we set $\pi(U) = 0$ no dual violation happens. Moreover, B is an odd cycle and hence H_U indeed contains a hamiltonian cycle for the new set U added to Ω.

For the final case we observe that we are not adding any sets to Ω and ϵ is chosen to ensure dual feasiblity. Deshrinking preserves laminarity. $\square$

Claim 4 *If M is a matching in G' then there is a matching N in G_π where number of $N-$exposed nodes is same as $M-$exposed nodes.*

Proof: We can recursively expand the nodes in G' and extend M using the fact that H_U has a Hamiltonian cycle for each $U \in \Omega$. $\square$

Corollary 5 *If M is a perfect matching in G' then it can be extended to perfect matching N in G_π.*

Claim 6 *If the algorithm terminates with a perfect matching then it is an optimal matching.*

Proof: Let π be the feasible dual solution at the end of the algorithm. If M is a perfect matching in G_π then, $\{x(e) = 1$ if $e \in M$ and $x(e) = 0$ otherwise$\}$, is a feasible primal solution and x and π satisfy complementary slackness conditions thus implying that both the solutions are optimal. $\square$

The above claims show that if the algorithm terminates then it outputs an optimum solution. Now we establish that the algorithm indeed terminates.

Lemma 7 (Main Lemma) *The algorithm terminates in $O(|V|^2)$ iterations.*

Each iteration can be implemented in $O(m)$ time with minimal data structures, Thus we have the following theorem due to Edmonds.

Theorem 8 (Edmonds) *There is a an $O(n^2m)$ time algorithm for the min cost perfect matching problem.*

As a corollary we also obtain

Corollary 9 *The polytope $Q(G)$ described by the inequalities below:*

$$\begin{array}{rcll} x(\delta(v)) & = & 1 & \forall v \in V \\ x(\delta(U)) & \geq & 1 & \forall U \subset V, |U| \geq 3, |U| odd \\ x(e) & \geq & 0 & \forall e \in E \end{array} \tag{2}$$

is the convex hull of the perfect matchings in G.

Proof: The algorithm shows that for any given weights $w : E \to \mathcal{R}^+$, the linear program $\min\{w \cdot x \mid x \in Q(G)\}$ has an integral optimum solution whenever $Q(G) \neq \emptyset$. Since $w \geq 0$ can be assumed w.l.o.g. so $Q(G)$ is an integral polyhedron. $\square$

Now we finish the proof of the key lemma on termination. First we observe the following.

Proposition 10 *In any iteration, the number of $M-$exposed nodes in G' does not increase which implies that the induced matching in G cannot decrease in size.*

Proof: It is easy to see that steps (1) - (4) of the algorithm don't increase the number of $M-$exposed nodes in G'. The only non-trivial case is step (5) in which the dual value is changed. Now in this step, recall the Edmonds Gallai decomposition and notice that $A(G')$ is matched only to $D(G')$. The dual update in this step leaves any edge uv between $D(G')$ and $A(G')$ tight and so all the $M-$edges remain tight in this step. $\square$

Claim 11 (Main claim) *If a set U is added to Ω in some iteration ("shrinking") then it is removed from Ω ("deshrinking") only after the matching size has increased.*

Proof: When U is added to Ω, it corresponds to the blossom of an $M-$flower where M is the current matching in G'. Let v be the node in G' corresponding to U after it is shrunk. If X is the set of $M-$exposed nodes then there is an $X - v$, $M-$alternating, even length path. If there is no matching augmentation then v continues to have an $X - v$, $M-$alternating, even length path or v is swallowed by a larger set U' that is shrunk. In the former case, v cannot be in $A(G')$ and hence cannot be "deshrunk". In the latter case, U' is not deshrunk before a matching augmentation and hence U, which is inside U', cannot be deshrunk before a matching augmentation. □

Claim 12 *Suppose iteration i has a matching augmentation and iteration $j > i$ is the next matching augmentation. Then between i and j, there are at most $|V|$ shrinkings and at most $|V|$ deshrinkings.*

Proof: Let Ω be the laminar family of shrunk sets at the end of iteration i. By the previous claim, before iteration i, we can only deshrink sets in Ω. Hence # of deshrinkings is $\leq |\Omega| - |V|$ since we cannot deshrink singletons. Thus the number of deshrinkings is $\leq |V|$.

Similarly number of of shrinkings is at most $|\Omega'| - |V|$ where Ω' is the laminar family just before iteration j. Again this gives an upper bound of $|V|$. □

Now lets have a look at what else can happen in an iteration other than augmentation, shrinking, and deshrinking. In step (5), an edge $uv \in E \backslash E_\pi$ can become tight and join E_π. One of the following two cases must happen since dual values are increased for nodes in $D(G')$, decreased for nodes in $A(G')$, and unchanged for $C(G')$:

1. $u, v \in D(G')$.
2. $u \in D(G'), v \in C(G')$.

The following two claims take care of these cases.

Claim 13 *If edge uv becomes tight in an iteration and $u, v \in D(G')$ then the next iteration is either a shrinking iteration or an augmentation iteration.*

Proof: Let X be the set of $M-$exposed nodes in G'. If $u, v \in D(G')$ then $G' + uv$ creates an $X - X$ $M-$alternating walk of odd length because $D(G')$ consists of all the vertices reachable by a walk of even length from X. This implies that in the next iteration we have a non-empty walk leading to an augmentation or shrinking. □

Claim 14 *If $u \in D(G')$ and $v \in C(G')$ then in $G'' = G' + uv, v$ is reachable from X by an $M-$alternating path.*

Now we can prove the key lemma. We have a total of $\frac{|V|}{2}$ augmentation iterations. Between consecutive augmentation iterations, there are at most $2|V|$ shrinking and deshrinking iterations. Each other iteration is an edge becoming tight. Case 1 iterations can be charged to shrinking iterations. Total number of case 2 iterations is at most $|V|$ since each such iteration increases by 1, the number of nodes reachable from X with an $M-$alternating path. No other iteration decreases the number of nodes reachable before a matching augmentation. Thus the number of iterations between augmentation is $O(|V|)$. Hence total number of iterations is $O(|V|^2)$.

1 Total Dual Integrality

Recall that if A is TUM and b, c are integral vectors, then $\max\{cx : Ax \leq b\}$ and $\min\{yb : y \geq 0, yA = c\}$ are attained by integral vectors x and y whenever the optima exist and are finite. This gives rise to a variety of min-max results, for example we derived König's theorem on bipartite graphs. There are many examples where we have integral polyhedra defined by a system $Ax \leq b$ but A is not TUM; the polyhedron is integral only for some specific b. We may still ask for the following. Given any c, consider the maximization problem $\max\{cx : Ax \leq b\}$; is it the case that the dual minimization problem $\min\{yb : y \geq 0, yA = c\}$ has an integral optimal solution (whenever a finite optimum exists)?

This motivates the following definition:

Definition 1 *A rational system of inequalities $Ax \leq b$ is* totally dual integral (TDI) *if, for all integral c, $\min\{yb : y \geq 0, yA = c\}$ is attained by an integral vector y^* whenever the optimum exists and is finite.*

Remark 2 *If A is TUM, $Ax \leq b$ is TDI for all b.*

This definition was introduced by Edmonds and Giles[2] to set up the following theorem:

Theorem 3 *If $Ax \leq b$ is TDI and b is integral, then $\{x : Ax \leq b\}$ is an integral polyhedron.*

This is useful because $Ax \leq b$ may be TDI even if A is not TUM; in other words, this is a weaker sufficient condition for integrality of $\{x : Ax \leq b\}$ and moreover guarantees that the dual is integral whenever the primal objective vector is integral.

Proof Sketch. Let $P = \{x : Ax \leq b\}$. Recall that we had previously shown that the following are equivalent:

(i) P is integral.

(ii) Every face of P contains an integer vector.

(iii) Every minimal face of P contains an integer vector.

(iv) $\max\{cx : x \in P\}$ is achieved by an integral vector whenever the optimum is finite.

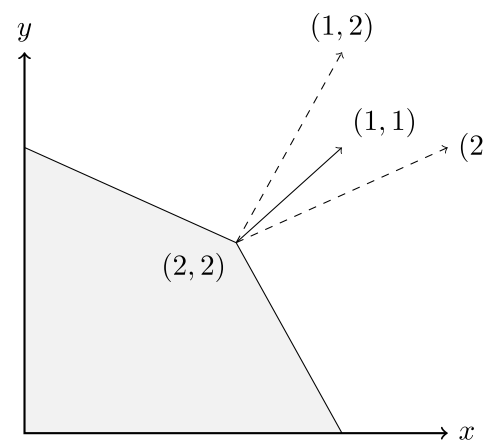
Edmonds and Giles proved two more equivalent conditions:

(v) Every rational supporting hyperplane of P contains an integer vector.

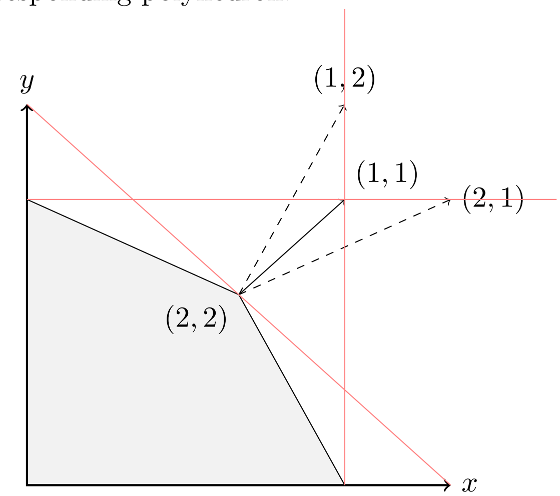
(vi) If c is integral, then $\max\{cx : x \in P\}$ is an integer whenever the optimum exists and is finite.

Condition (vi) implies the theorem as follows. If $Ax \leq b$ is TDI *and* b is integral, $\max\{cx : x \in P\}$ is an integer for all integral c whenever it is finite; this is because the dual optimum is achieved by an integral vector y^* (TDI property) and the objective function by^* is integral because b is integral. This implies that P is integral. $\square$

There's an important subtlety to the definition of total dual integrality: being TDI is a property of a system of inequalities, *not* a property of the corresponding polyhedron.



$x \geq 0, y \geq 0$
$x + 2y \leq 6$
$2x + y \leq 6$



$x \geq 0, y \geq 0$
$x + 2y \leq 6$
$2x + y \leq 6$
$x + y \leq 4$
$x \leq 3, y \leq 3$

We will illustrate this with an example from [3]. Consider the system $Ax \leq b$ drawn above on the left. If we take the cost vector c to be $(1, 1)$, then the primal has an optimum at $(2, 2)$ with value 4. The tight constraints at this vertex have normal vectors $(2, 1)$ and $(1, 2)$ (these are rows of A). Therefore, in order for the dual $yA = c$ to have an integer solution, we must be able to express $(1, 1)$ as an *integer* combination of $(2, 1)$ and $(1, 2)$. Since this is impossible, $Ax \leq b$ is not TDI.

However, suppose we add more constraints to obtain the system $A'x \leq b'$ drawn above on the right. Note that this system corresponds to the same polyhedron as $Ax \leq b$. However, now we have an additional normal vector at $(2, 2)$ – namely, $(1, 1)$. Thus $(1, 1)$ is now an integer combination of the normal vectors at $(2, 2)$. The system $A'x \leq b'$ is in fact TDI, even though it corresponds to the same polytope as the (non-TDI) system $Ax \leq b$.

The example demonstrates a necessary for a system to be TDI. We explain this in the general context. Consider the problem $\max\{cx : Ax \leq b\}$ with c integral, and assume it has a finite optimum β. Then it is achieved by some vector x^* in the face F defined by the intersection of $\{x : Ax \leq b\}$ with the hyperplane $cx = \beta$. For simplicity assume that the face F is an extreme point/vertex of the polyhedron and let $A'x^* = b'$ be the set of all inequalities in $Ax \leq b$ that are tight at x^*. The dual is $\min\{yb : y \geq 0, yA = c\}$. By LP duality theory, any dual optimum solution y corresponds to c being expressed a non-negative combination of the row vectors of A', in other words c is in the cone of the row vectors of A'. If $Ax \leq b$ is TDI then we ask for an integral dual optimum solution; this requires that there is an integer solution to $yA' = c, y \geq 0$. This motivates the following definition.

Definition 4 *A set $\{a_1, \ldots, a_k\}$ of vectors in R^n is a* Hilbert basis *if every integral vector $x \in Cone(\{a_1, \ldots, a_k\})$ can be written as $x = \sum_{i=1}^{k} \mu_i a_i$, $\mu_i \geq 0$, $\mu_i \in \mathbf{Z}$ (that is, x is a non-negative integer combination of $a_1, \ldots, a_k$). If the a_i are themselves integral, we call $\{a_1, \ldots, a_k\}$ an* integral Hilbert basis*.*

The following theorem is not difficult to prove with the background that we have developed.

Theorem 5 *The rational system $Ax \leq b$ is TDI if and only if the following property is true for each face F of P; let $A'x = b'$ be the set of* all *inequalities in $Ax \leq b$ that are tight/active at F, then the rows vectors of A' form a Hilbert basis.*

Corollary 6 *If the system $Ax \leq b, \alpha x \leq \beta$ is TDI then $Ax \leq b, \alpha x = \beta$ is also TDI.*

The example above raises the question of whether one can take any rational system $Ax \leq b$ and make it TDI by adding sufficiently many redundant inequalities. Indeed that is possible, and is based on the following theorem.

Theorem 7 *Every rational polyhedral cone has a finite integral Hilbert basis.*

Theorem 8 (Giles-Pulleyblank) *Any rational polyhedron P has a representation $Ax \leq b$ such that*

(i) $P = \{x : Ax \leq b\}$,

(ii) *A is integral, and*

(iii) *$Ax \leq b$ is TDI.*

Moreover, b is integral if and only if P is integral.

2 The Cunningham-Marsh Theorem

Suppose we have a graph $G = (V, E)$. Let $P_{odd}(V)$ denote the family of all odd subsets of V with size at least 3. Recall that in our study of matchings, we have examined three different systems of inequalities.

$$P_1: \quad \begin{array}{rll} x(\delta(v)) & = 1 & \forall v \in V \\ x(\delta(U)) & \geq 1 & U \in P_{odd}(V) \\ x & \geq 0 & \end{array}$$

$$P_2: \quad \begin{array}{rll} x(\delta(v)) & \leq 1 & \forall v \in V \\ x(E[U]) & \leq \left\lfloor \frac{1}{2}|U| \right\rfloor & U \in P_{odd}(V) \\ x & \geq 0 & \end{array}$$

$$P_3: \quad \begin{array}{rll} x(\delta(v)) & = 1 & \forall v \in V \\ x(E[U]) & \leq \left\lfloor \frac{1}{2}|U| \right\rfloor & U \in P_{odd}(V) \\ x & \geq 0 & \end{array}$$

Here P_2 determines the matching polytope for G, while P_1 and P_3 determine the perfect matching polytope.

It is not hard to see that P_1 is not TDI. Consider K_4 with $w(e) = 1$ for every edge e. In this case, the unique optimal dual solution is $y_v = \frac{1}{2}$ for each vertex v.

On the other hand, P_2 and P_3 *are* TDI; this was proven by Cunningham and Marsh[1]. Consider the primal maximization and dual minimzation problems for P_2 below:

maximize wx subject to
$$\begin{aligned} x(\delta(v)) &\le 1 && \forall v \in V \\ x(E[U]) &\le \left\lfloor \tfrac{1}{2}|U| \right\rfloor && \forall U \in P_{odd}(V) \\ x &\ge 0 \end{aligned}$$

minimize $\displaystyle\sum_{v \in V} y_v + \sum_{U \in P_{odd}(V)} z_U \cdot \left\lfloor \frac{1}{2}|U| \right\rfloor$ subject to
$$y_a + y_b + \sum_{\substack{U \in P_{odd}(V) \\ a,b \in U}} z_U \ge w(ab) \;\; \forall ab \in E$$
$$y \ge 0,\; z \ge 0$$

By integrality of the matching polytope, the maximum value of the primal is the maximum weight of a matching under w; by duality, this equals the minimum value of the dual. The Cunningham-Marsh Theorem tells us that this minimum value is achieved by integral dual vectors y^*, z^* with the additional condition that the sets $\{U : z^*_U > 0\}$ form a laminar family.

Theorem 9 (Cunningham-Marsh) *The system P_2 is TDI (as is P_3). More precisely, for every integral w, there exist integral vectors y and z that are dual feasible such that $\{U : z_U > 0\}$ is laminar and*
$$\sum_{v \in V} y_v + \sum_{U \in P_{odd}(V)} z_U \cdot \left\lfloor \frac{1}{2}|U| \right\rfloor = \nu(w)$$
where $\nu(w)$ is the maximum weight of a matching under w.

Exercise 10 *Show that the Tutte-Berge Formula can be derived from the Cunningham-Marsh Theorem.*

Cunningham and Marsh originally proved this theorem algorithmically, but we present a different proof from [4]; the proof relies on the fact that P_2 is the matching polytope. A different proof is given in [4] that does not assume this and in fact derives that P_2 is the matching polytope as a consequence.

Proof: We will use induction on $|E| + w(E)$ (which is legal because w is integral). Note that if $w(e) \le 0$ for some edge e, we may discard it; hence we may assume that $w(e) \ge 1$ for all $e \in E$.

Case I: Some vertex v belongs to every maximum-weight matching under w. Define $w' : E \to \mathbf{Z}^+$ by
$$\begin{aligned} w'(e) &= w(e) - 1 && \text{if } e \in \delta(v) \\ w'(e) &= w(e) && \text{if } e \notin \delta(v) \end{aligned}$$

Now induct on w'. Let y', z' be an integral optimal dual solution with respect to w' such that $\{U : z'_U > 0\}$ is laminar; the value of this solution is $\nu(w')$. Because v appears in every maximum-weight matching under w, $\nu(w') \le \nu(w) - 1$; by definition of w', $\nu(w') \ge \nu(w) - 1$. Thus $\nu(w') = \nu(w) - 1$.

Let y^* agree with y' everywhere except v, and let $y^*_v = y'_v + 1$. Let $z^* = z'$. Now y^*, z^* is a dual feasible solution with respect to w, the solution is optimal since it has weight $\nu(w') + 1 = \nu(w)$, and $\{U : z^*_U > 0\}$ is laminar since $z^* = z'$.

Case II: No vertex belongs to every maximum-weight matching under w. Let y, z be a *fractional* optimal dual solution. Observe that $y = 0$, since $y_v > 0$ for some vertex v, together with complementary slackness, would imply that every optimal primal solution covers v, i.e. v belongs to every maximum-weight matching under w. Among all optimal dual solutions y, z (with $y = 0$) choose the one that maximizes $\sum_{U \in P_{odd}(V)} z_U \left\lfloor \frac{1}{2} |U| \right\rfloor^2$. To complete the proof, we just need to show that z is integral and $\{U : z_U > 0\}$ is laminar.

Suppose $\{U : z_U > 0\}$ is not laminar; choose $W, X \in P_{odd}(V)$ with $z_W > 0, z_X > 0$, and $W \cap X \neq \emptyset$. We claim that $|W \cap X|$ is odd. Choose $v \in W \cap X$, and let M be a maxmimum-weight matching under w that misses v. Since $z_W > 0$, by complementary slackness, M contains $\left\lfloor \frac{1}{2} |W| \right\rfloor$ edges inside W; thus v is the *only* vertex in W missed by M. Similarly, v is the only vertex in X missed by M. Thus M covers $W \cap X - \{v\}$ using only edges inside $W \cap X - \{v\}$, so $|W \cap X - \{v\}|$ is even, and so $|W \cap X|$ is odd. Let ϵ be the smaller of z_W and z_X; form a new dual solution by decreasing z_W and z_X by ϵ and increasing $z_{W \cap X}$ and $z_{W \cup X}$ by ϵ (this is an uncrossing step).

We claim that this change maintains dual feasibility and optimality. Clearly z_W and z_X are still nonnegative. If an edge e is contained in W and X, then the sum in e's dual constraint loses 2ϵ from z_W and z_X, but gains 2ϵ from $z_{W \cap X}$ and $z_{W \cup X}$, and hence still holds. Likewise, if e is contained in W but not X (or vice-versa), the sum loses ϵ from z_W but gains ϵ from $z_{W \cup X}$. Thus these changes maintained dual feasibility and did not change the value of the solution, so we still have an optimal solution. However, we have increased $\sum_{U \in P_{odd}(V)} z_U \left\lfloor \frac{1}{2} |U| \right\rfloor^2$ (the reader should verify this), which contradicts the choice of z. Thus $\{U : z_U > 0\}$ is laminar.

Suppose instead that z is not integral. Choose a maximal $U \in P_{odd}(V)$ such that z_U is not an integer. Let $U_1, \ldots, U_k$ be maximal odd sets contained in U such that each $z_{U_i} > 0$. (Note that we may have $k = 0$.) By laminarity, $U_1, \ldots, U_k$ are disjoint. Let $\alpha = z_U - \lfloor z_U \rfloor$. Form a new dual solution by decreasing z_U by α and increasing each z_{U_i} by α.

We claim that the resulting solution is dual feasible. Clearly we still have $z_U \geq 0$, and no other dual variable was decreased. Thus we need only consider the edge constraints; moreover, the only constraints affected are those corresponding to edges contained within U. Let e be an edge contained in U. If e is contained in some U_i, then the sum in e's constraint loses α from z_U but gains α from z_{U_i}, so the sum does not change. On the other hand, suppose e is not contained in any U_i. By maximality of U and the U_i, U is the only set in P_{odd} containing e. Thus before we changed z_U we had $z_U \geq w(e)$; because $w(e)$ is integral, we must still have $z_U \geq w(e)$. Hence our new solution is dual feasible.

Since the U_i are disjoint, contained in U, and odd sets, $\left\lfloor \frac{1}{2} |U| \right\rfloor > \sum_{i=1}^{k} \left\lfloor \frac{1}{2} |U_i| \right\rfloor$. Thus our new solution has a smaller dual value than the old solution, which contradicts the optimality of z. It follows that z was integral, which completes the proof.

To show that the system P_3 is TDI, we use Corollary 6 and the fact that system P_2 is TDI. □

References

[1] W. H. Cunningham and A. B. Marsh, III. *A primal algorithm for optimal matching.* Mathematical Programming Study 8 (1978), 50–72.

[2] J. Edmonds and R. Giles. *Total dual integrality of linear inequality systems.* Progress in combinatorial optimization (Waterloo, Ont., 1982), 117–129, Academic Press, Toronto, ON, 1984.

[3] Lecture notes from Michael Goemans's class on Combinatorial Optimization. http://math.mit.edu/ goemans/18438/lec6.pdf. 2009.

[4] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1998.

CS 598CSC: Combinatorial Optimization **Lecture date: 02 March, 2010**
Instructor: Chandra Chekuri **Scribe: Ben Raichel**

1 T-joins and Applications

This material is based on [1] (Chapter 5), and also [2] (Chapter 29).

Edmonds was motivated to study T-joins by the Chinese postman problem which is the following.

Problem 1 *Let $G = (V, E)$ be an undirected graph and $c : E \to \mathbb{R}^+$ be non-negative edge weights on the edges. A Chinese postman tour is a walk that starts at some arbitrary vertex and returns to it after traversing each edge of E. Note that an edge may be traversed more than once. The goal is to find a postman tour of minimum total edge cost.*

Proposition 1 *If G is Eulerian then the optimal postman tour is an Eulerian tour of G and has cost equal to $\sum_{e \in E} c(e)$.*

Thus the interesting case is when G is not Eulerian. Let $T \subseteq V$ be the nodes with odd degree in G.

Fact 1 *$|T|$ is even.*

Consider a postman tour and say it visits an edge $x(e)$ times, where $x(e) \geq 1$ is an integer. Then, it is easy to see that the multigraph induced by placing $x(e)$ copies of e is in fact Eulerian. Conversely if $x(e) \geq 1$ and $x(e) \in \mathbb{Z}^+$ such that the graph is Eulerian, then it induces a postman tour of cost $\sum_{e \in E} c(e)x(e)$.

We observe that if $x(e) > 2$ then reducing $x(e)$ by 2 maintains feasibility. Thus $x(e) \in \{1, 2\}$ for each e in any minimal solution. If we consider the graph induced by $x(e)' = x(e) - 1$ we see that each node in T has odd degree and every other node has even degree. This motivates the definition of T-joins.

Definition 2 *Given a graph, $G = (V, E)$, and a set, $T \subseteq V$, a T-join is a subset $J \subseteq E$ such that in the graph (V, J), T is the set of nodes with odd degree.*

Proposition 3 *There is a T-join in G iff $|K \cap T|$ is even for each connected component K of G. In particular, if G is connected then there exists a T-join iff $|T|$ is even.*

Proof: Necessity is clear. For sufficiency, assume G is connected, otherwise we can work with each connected component separately. Let $T = \{v_1, v_2, \ldots, v_{2k}\}$. Let P_i be an arbitrary path joining v_i and v_{i+k}. Then the union of the paths $P_1, P_2, \ldots, P_k$ induces a multigraph in which the nodes in T are the only ones with odd degree. Let $x(e)$ be the number of copies of e in the above union. Then $x'(e) = x(e)\ mod\ 2$, is the desired T-join. (Note that the pairing of the vertices was arbitrary and hence any pairing would work.) □

Proposition 4 *J is a T-join iff J is the union of edge disjoint cycles and $\frac{1}{2}|T|$ paths connecting disjoint pairs of nodes in T.*

Proof: This is left as an exercise. □

1.1 Algorithm for Min-cost T-joins

Given $G = (V, E)$, $c : E \to \mathbb{R}$ and $T \subseteq V$, where $|T|$ even, we want to find the min-cost T-join. If all edge costs are non-negative then one can easily reduce the problem to a matching problem as follows. Assume without loss of generality that G is connected.

1. For each pair $u, v \in T$ let $w(uv)$ be the shortest path distance between u and v in G, with edge length given by c. Let P_{uv} be the shortest path between u and v.

2. Let H be the complete graph on T with edge weights $w(uv)$.

3. Compute a minimum weight perfect matching M in H.

4. Let $J = \{e \mid e$ occurs in an odd number of paths P_{uv}, $uv \in M\}$. Output J.

Theorem 5 *There is a strongly polynomial time algorithm to compute a min-cost T-join in a graph, $G = (V, E)$ with $c \geq 0$.*

Proof Sketch. To see the correctness of this algorithm first note that it creates a T-join since it will return a collection of $\frac{1}{2}|T|$ disjoint paths, which by Proposition 4 is a T-join (Note the fourth step in the algorithm is required to handle zero cost edges, and is not necessary if $c > 0$). It can be seen that this T-join is of min-cost since the matching is of min-cost (and since, ignoring zero cost edges, the matching returned must correspond to disjoint paths in G). $\square$

The interesting thing is that min-cost T-joins can be computed even when edge lengths can be negative. This has several non-trivial applications. We reduce the general case to the non-negative cost case by making the following observations.

Fact 2 *If A,B are two subsets of U then $|A \Delta B|$ is even iff $|A|$ and $|B|$ have the same parity, where we define $X \Delta Y$ as the symmetric difference of X and Y.*

Proposition 6 *Let J be a T-join and J' be a T'-join then $J \Delta J'$ is a $(T \Delta T')$-join.*

Proof: Verify using the above fact that each $v \in T \Delta T'$ has odd degree and every other node has even degree in $J \Delta J'$. $\square$

Corollary 7 *If J' is a T'-join and $J \Delta J'$ is a $(T \Delta T')$-join then J is a T-join.*

Proof: Note that $(T \Delta T') \Delta T' = T$ and similarly $(J \Delta J') \Delta J' = J$. Hence the corollary is implied by application of the above proposition.

$\square$

Given $G = (V, E)$ with $c : E \to \mathbb{R}$, let $N = \{e \in E \mid c(e) < 0\}$. Let T' be the set of nodes with odd degree in $G[N]$. Clearly N is a T'-join by definition. Let J'' be a $(T \Delta T')$-join in G with the costs on edges in N negated (i.e. $c(e) = |c(e)|, \forall e \in E$).

Claim 8 *$J = J'' \Delta N$ is a T-join, where $N = \{e \in E \mid c(e) < 0\}$, $T' = \{v \in G[N] \mid \delta_{G[N]}(v)$ is odd$\}$, and J'' is a $(T \Delta T')$-join.*

Proof: By the above corollary, since J'' is a $(T \Delta T')$-join and N is a T'-join, $J'' \Delta N$ is a $(T \Delta T') \Delta T' =$ T-join. $\square$

Claim 9 *$c(J) = |c|(J'') + c(N)$, where $|c|(X) = \sum_{x \in X} |c(x)|$ and J, J'', and N are as defined above.*

Proof:

$$\begin{aligned} c(J) = c(J''\Delta N) &= c(J'' \setminus N) + c(N \setminus J'') \\ &= c(J'' \setminus N) - c(J'' \cap N) + c(J'' \cap N) + c(N \setminus J'') \\ &= c(J'' \setminus N) + |c|(J'' \cap N) + c(N) = |c|(J'') + c(N). \end{aligned}$$

□

Corollary 10 *$J = J''\Delta N$ is a min cost T-join in G iff J'' is a min cost $(T\Delta T')$-join in G with edge costs $|c|$, where $N = \{e \in E \mid c(e) < 0\}$, $T' = \{v \in G[N] \mid \delta_{G[N]}(v)$ is odd$\}$, and J'' is a $(T\Delta T')$-join.*

Proof Sketch. By using the last claim, necessity is clear since $c(N)$ is a constant and hence when $c(J)$ is minimized so is $|c|(J'')$. To use the same argument for sufficiency, one must argue that for any T-join, J, we have that $J = J''\Delta N$ for some $(T\Delta T')$-join, J''. □

The above corollary gives a natural algorithm to solve the general case by first reducing it to the non-negative case. In the algorithm below, let $c : E \to \mathbb{R}$, $|c| : E \to \mathbb{R}^+$ such that $|c|(e) = |c(e)|$, $G_{|c|}$ be the graph with the weight function $|c|$, $N = \{e \in E \mid c(e) < 0\}$, and $T' = \{v \in G[N] \mid \delta_{G[N]}(v)$ is odd$\}$.

1. Compute a $(T\Delta T')$-join , J'', on $G_{|c|}$ using the algorithm above for $c \geq 0$
2. Output $J = J''\Delta N$.

Theorem 11 *There is a strongly polynomial time algorithm for computing a min-cost T-join in a graph, even with negative costs on the edges.*

Proof: We know the above algorithm outputs a T-join by Claim 8. Since J'' was computed on $G_{|c|}$, which has non-negative edge weights, by the proof of Theorem 5, J'' is a min-cost T-join. Hence by Corollary 10 J is a min-cost T-join. □

1.2 Applications

1.2.1 Chinese Postman

We saw earlier that a min-cost postman tour in G is the union of E and a T-join where T is the set of odd degree nodes in G. Hence we can compute a min-cost postman tour.

1.2.2 Shortest Paths and Negative lengths

In directed graphs the well known Bellman-Ford and Floyd-Warshall algorithms can be used to check whether a given directed graph, $D = (V, A)$, has negative length cycles or not in $O(mn)$ and $O(n^3)$ time respectively. Moreover, if there is no negative length cycle then the shortest s-t path can be found in the same time. However, one cannot use directed graph algorithms for undirected graphs when there are negative lengths, since bi-directing an undirected edge creates a negative length cycle. However, we can use T-join techniques.

Proposition 12 *An undirected graph, $G = (V, E)$, with $c : E \to \mathbb{R}$ has a negative length cycle iff an $\emptyset$-join has negative cost.*

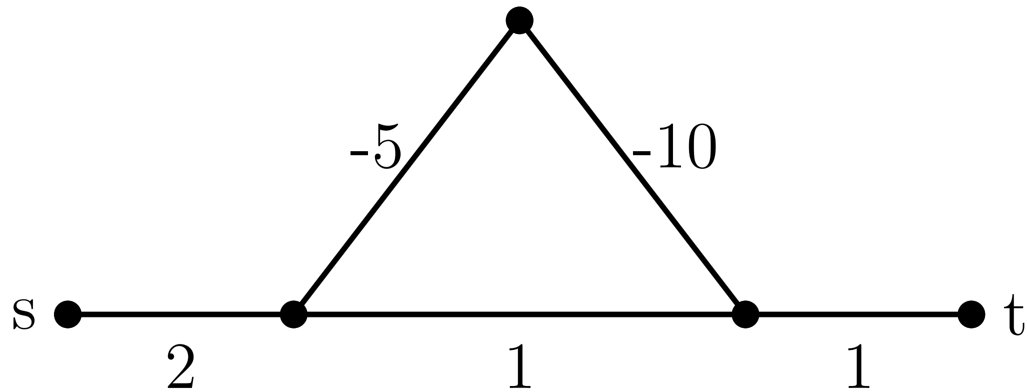


Figure 1: An example of a graph with a negative cost $\emptyset$-join

Proposition 13 *If G has no negative length cycle then the min-cost $\{s,t\}$-join gives an s-t shortest path.*

Remark 14 *It is important to first check for negative length cycles before finding an $\{s,t\}$-join.*

Theorem 15 *There is a strongly polynomial time algorithm that given an undirected graph, $G(V, E)$, with $c : E \to \mathbb{R}$, either outputs a negative length cycle or an s-t shortest path.*

Proof Sketch. We first compute a min-cost $\emptyset$-join. By Proposition 12 we know that if this $\emptyset$-join has negative cost then we can produce a negative length cycle. Otherwise, we know there is no negative length cycle and hence by Proposition 13 we can compute a min-cost $\{s,t\}$-join in order to find an s-t shortest path. (In each case the T-join can be computed using the algorithm from the previous section.) $\square$

1.2.3 Max-cut in planar graphs

Since one can compute min-cost T-joins with negative costs, one can compute max-cost T-joins as well. The max-cut problem is the following.

Problem 2 *Given an undirected graph with non-negative edge weights, find a partition of V into $(S, S \setminus V)$ so as to maximize $w(\delta(S))$.*

Max-cut is NP-hard in general graphs, but Hadlock showed how T-joins can be used to solve it in polynomial time for planar graphs. A basic fact is that in planar graphs, cuts in G correspond to collections of edge disjoint cycles in the dual graph G^*. Thus to find a max-cut in G we compute a max $\emptyset$-join in G^* where the weight of an edge in G^* is the same as its corresponding edge in the primal.

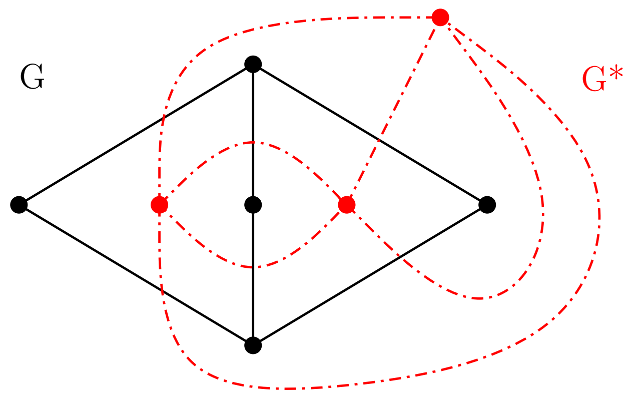


Figure 2: A planar graph, G, in black, and its dual, G^*, in dashed red.

1.2.4 Polyhedral aspects

The following set of inequalities can be shown to determine the characteristic vectors of the set of T-joins in a graph G.

$$\begin{aligned} 0 \le x(e) \le 1 & \\ x(\delta(U) \setminus F) - x(F) \ge 1 - |F| & \qquad U \subseteq V,\ F \subseteq \delta(U),\ |U \cap T| + |F| \text{ is odd} \end{aligned}$$

References

[1] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1998.

[2] A. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, 1998.

CS 598CSC: Combinatorial Optimization	**Lecture date: March 4, 2010**
Instructor: Chandra Chekuri	**Scribe: Vineet Abhishek**

The presentation here is based on [1] and [2].

1 Introduction to Matroids

Matroids (formally introduced by Whitney in 1935) are combinatorial structures that capture the abstract properties of linear independence defined for vector spaces.

Definition 1 *A matroid $\mathcal{M}$ is a tuple $(S, \mathcal{I})$, where S is a finite ground set and $\mathcal{I} \subseteq 2^S$ (the power set of S) is a collection of* independent sets*, such that:*

1. *$\mathcal{I}$ is nonempty, in particular, $\emptyset \in \mathcal{I}$,*
2. *$\mathcal{I}$ is downward closed; i.e., if $X \in \mathcal{I}$ and $Y \subseteq X$, then $Y \in \mathcal{I}$,*
3. *If $X, Y \in \mathcal{I}$, and $|X| < |Y|$, then $\exists y \in Y \backslash X$ such that $X + y \in \mathcal{I}$.*

Exercise 2 *Show that the third property in Definition 1 can be replaced by the following: if $X, Y \in \mathcal{I}$, $|Y| = |X| + 1$, then $\exists y \in Y \backslash X$ such that $X + y \in \mathcal{I}$.*

Example 3 (Vector Matroid) *Let M be a $m \times n$ matrix with entries in some field $\mathbb{F}$ and v_i be the i^{th} column of M, viewed as a vector in the vector space $\mathbb{F}^m$. Let $S = \{1, 2, \ldots, n\}$ and $\mathcal{I} = \{I : I \subseteq S, \{v_i\}_{i \in I}$ are linearly independent$\}$ (under the usual definition of linear independence in linear algebra). Then $\mathcal{M} = (S, \mathcal{I})$ is a matroid. To see this, notice that properties 1 and 2 of Definition 1 are trivially satisfied. To show property 3, suppose $X, Y \in \mathcal{I}$ and $|X| < |Y|$. If there is no $y \in Y \backslash X$ such that $X + y \in \mathcal{I}$, then Y is in the span of $\{v_x\}_{x \in X}$. Hence, $|Y| \leq |X|$ which is a contradiction.*

Example 4 (Graphic Matroid) *Let $\mathcal{G} = (V, E)$ be an undirected multi-graph (loops allowed). Let $\mathcal{I} = \{I : I \subseteq E,\ I$ induces a forest in $\mathcal{G}\}$. Then $\mathcal{M} = (E, \mathcal{I})$ is a matroid. Again, the first two properties of Definition 1 are easy to verify. To show property 3, suppose $X, Y \in \mathcal{I}$ such that $|X| < |Y|$. Both X and Y induce forests in $\mathcal{G}$. Let $V_1, V_2, \ldots, V_{k(X)}$ be the vertex sets of the connected components in $G[X]$ ($\mathcal{G}$ restricted to the edge set X). Here, $k(X)$ denotes the number of connected components in $G[X]$. Each connected component is a tree. Hence, if there is an edge $y \in Y$ that connects two different components of $\mathcal{G}[X]$ then $\mathcal{G}[X + y]$ is again a forest and we are done. If not, then every edge $y \in Y$ have its both ends in the same component of $\mathcal{G}[X]$. Thus, the number of connected components in $\mathcal{G}[Y]$, denoted by $k(Y)$, is at least $k(X)$. Thus, $|X| = |V| - k(X) \geq |V| - k(Y) = |Y|$, which is a contradiction.*

Example 5 (Uniform Matroid) *Let $\mathcal{M} = (S, \mathcal{I})$, where S is any finite nonempty set, and $\mathcal{I} = \{I : I \subseteq S, |I| \leq k\}$ for some positive integer k. Then $\mathcal{M}$ is a matroid.*

Example 6 (Partition Matroid) *Let $S_1, S_2, \ldots, S_n$ be a partition of S and $k_1, k_2, \ldots, k_n$ be positive integers. Let $\mathcal{I} = \{I : I \subseteq S,\ |I \cap S_i| \leq k_i$ for all $1 \leq i \leq n\}$. Then $\mathcal{M} = (S, \mathcal{I})$ is a matroid.*

Example 7 (Laminar Matroid) *Let $\mathcal{F}$ be a laminar family on S (i.e., if $X, Y \in \mathcal{F}$ then $X, Y \subseteq S$; and either $X \cap Y = \emptyset$, or $X \subseteq Y$, or $Y \subseteq X$) such that each $x \in S$ is in some set $X \in \mathcal{F}$. For each $X \in \mathcal{F}$, let $k(X)$ be a positive integer associated with it. Let $\mathcal{I} = \{I : I \subseteq S,\ |I \cap X| \leq k(X)\ \forall X \in \mathcal{F}\}$. Then $\mathcal{M} = (S, \mathcal{I})$ is a matroid. Notice that laminar matroids generalize partition matroids, which in turn generalize uniform matroids.*

Exercise 8 *Verify Example 7.*

Example 9 (Transversal Matroid) *Let $\mathcal{G} = (V, E)$ be a bipartite graph with bipartition V_1 and V_2. let $\mathcal{I} = \{I : I \subseteq V_1, \exists$ a matching M in $\mathcal{G}$ that covers $I\}$. Then $\mathcal{M} = (V_1, \mathcal{I})$ is a matroid.*

Example 10 (Matching Matroid) *Let $\mathcal{G} = (V, E)$ be an undirected graph. Let $\mathcal{I} = \{I : I \subseteq V, \exists$ a matching M in $\mathcal{G}$ that covers $I\}$. Then $\mathcal{M} = (V, \mathcal{I})$ is a matroid.*

Exercise 11 *Verify Examples 9 and 10.*

1.1 Base, Circuit, Rank, Span and Flat

Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid.

Definition 12 *A set $X \subseteq S$ such that $X \notin \mathcal{I}$ is called a* dependent set *of $\mathcal{M}$.*

Definition 13 *A* loop *is an element $x \in S$ such that $\{x\}$ is dependent.*

Notice that a loop cannot appear in any sets in $\mathcal{I}$ and can be effectively removed from S.

Definition 14 *A* base *is an inclusion wise maximal set in $\mathcal{I}$.*

Proposition 15 *If B and $\widehat{B}$ are bases of $\mathcal{M}$ then $|B| = |\widehat{B}|$.*

Proof: If $|B| < |\widehat{B}|$ then from Definition 1, $\exists x \in \widehat{B} \backslash B$ such that $B + x \in \mathcal{I}$, contradicting the maximality of B. □

Notice that the notion of base here is similar to that of a *basis* in linear algebra.

Lemma 16 *Let B and $\widehat{B}$ be bases of $\mathcal{M}$ and $x \in \widehat{B} \backslash B$. Then $\exists y \in B \backslash \widehat{B}$ such that $\widehat{B} - x + y$ is a base of $\mathcal{M}$.*

Proof: Since $\widehat{B} - x \in \mathcal{I}$ and $|\widehat{B} - x| < |B|$, $\exists y \in B \backslash \widehat{B}$ such that $\widehat{B} - x + y \in \mathcal{I}$. Then $|\widehat{B} - x + y| = |B|$, implying that $\widehat{B} - x + y$ is a base of $\mathcal{M}$. □

Definition 17 *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. Given $\widehat{S} \subseteq S$, let $\widehat{\mathcal{I}} = \{I : I \subseteq \widehat{S}, I \in \mathcal{I}\}$. Then $\widehat{\mathcal{M}} = (\widehat{S}, \widehat{\mathcal{I}})$ is also a matroid and is referred to as the* restriction of $\mathcal{M}$ to $\widehat{S}$.

Definition 18 *Given $\mathcal{M} = (S, \mathcal{I})$ and $\widehat{S} \subseteq S$, $\widehat{B}$ is a* base for $\widehat{S}$ *if $\widehat{B}$ is a base of $\widehat{\mathcal{M}}$, where $\widehat{\mathcal{M}}$ is a restriction of $\mathcal{M}$ to $\widehat{S}$.*

Proposition 19 *Given $\mathcal{M} = (S, \mathcal{I})$, let $B \subseteq X$ be a base for X. Then for any $Y \supseteq X$, there exist a base $\widehat{B}$ for Y that contains B.*

Proof: Notice that B is independent in the restriction of $\mathcal{M}$ to Y (henceforth *independent in* Y). Let $\widehat{B}$ be the maximal independent set in Y that contains B. Since all maximal independent sets have same size, $\widehat{B}$ is a base of Y. $\square$

Definition 20 *Given $\mathcal{M} = (S, \mathcal{I})$, a circuit is a minimal dependent set (i.e., an inclusion wise minimal set in $2^S \backslash \mathcal{I}$). Thus, if C is a circuit then $\forall x \in C$, $C - x \in \mathcal{I}$.*

The definition of a circuit is related to graph theory in the following sense: if $\mathcal{M}$ is the graphic matroid of a graph $\mathcal{G}$, then the circuits of $\mathcal{M}$ are the cycles of $\mathcal{G}$. Single element circuits of a matroid are loops; if $\mathcal{M}$ is a graphic matroid of a graph $\mathcal{G}$, then the set of loops of $\mathcal{M}$ is precisely the set of loops of $\mathcal{G}$.

Lemma 21 *Let C_1 and C_2 be two circuits such that $C_1 \neq C_2$ and $x \in C_1 \cap C_2$. Then for every $x_1 \in C_1 \backslash C_2$ there is a circuit C such that $x_1 \in C$ and $C \subseteq C_1 \cup C_2 - x$. In particular, $C_1 \cup C_2 - x$ contains a circuit.*

Proof: Notice that $C_1 \backslash C_2$ is nonempty (and so is $C_2 \backslash C_1$), otherwise, $C_1 \subseteq C_2$. Since $C_1 \neq C_2$, C_1 is a strict subset of C_2, contradicting the minimality of C_2.

Let $C_1 \cup C_2 - x$ contain no circuits. Then $B = C_1 \cup C_2 - x$ is independent, and hence, a base for $C_1 \cup C_2$ (since it is maximal). Also, $|B| = |C_1 \cup C_2| - 1$. Since $C_1 \cap C_2$ is an independent set, we can find a base $\widehat{B}$ for $C_1 \cup C_2$ that contains $C_1 \cap C_2$. Then $|\widehat{B}| = |B| = |C_1 \cup C_2| - 1$. Since $C_1 \setminus C_2$ and $C_2 \setminus C_1$ are both non-empty, this is possible only if either $C_1 \subseteq \widehat{B}$ or $C_2 \subseteq \widehat{B}$, contradicting that $\widehat{B}$ is a base. Hence, $C_1 \cup C_2 - x$ must contain a circuit.

Now let $x_1 \in C_1 \backslash C_2$. Let B_1 be a base for $C_1 \cup C_2$ that contains $C_1 - x_1$, and B_2 be a base for $C_1 \cup C_2$ that contains $C_2 - x$. Clearly, $x_1 \notin B_1$ and $x \notin B_2$. If $x_1 \notin B_2$ then $B_2 + x_1$ must have a circuit and we are done. If $x_1 \in B_2$, then from Lemma 16, there exists $\widehat{x} \in B_1 \backslash B_2$ such that $\widehat{B} = B_2 - x_1 + \widehat{x}$ is a base for $C_1 \cup C_2$. Notice that $\widehat{x} \neq x$, otherwise $C_2 \subseteq \widehat{B}$. Thus, $x_1 \notin \widehat{B}$ and $\widehat{B} + x_1$ contains the circuit satisfying the condition of Lemma 21. $\square$

Corollary 22 *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. If $X \in \mathcal{I}$ and $y \notin X$ then either $X + y \in \mathcal{I}$ or there is a unique circuit C in $X + y$. Moreover, for each $\widehat{y} \in C$, $X + y - \widehat{y} \in \mathcal{I}$.*

Proof: If $X + y \notin \mathcal{I}$, then it must contain a circuit C_1. Assume there is another circuit $C_2 \subseteq X + y$, and $C_1 \neq C_2$. Since $X \in \mathcal{I}$, both C_1 and C_2 must contain y. From Lemma 21, $C_1 \cup C_2 - y$ contains a circuit. But this is a contradiction since $C_1 \cup C_2 - y \subseteq X$. Hence, $X + y$ contains a unique circuit, call it C. Now, if for some $\widehat{y} \in C$, $X + y - \widehat{y} \notin \mathcal{I}$, then $X + y - \widehat{y}$ is dependent and contains a circuit $\widehat{C}$. However, $\widehat{C} \neq C$ since $\widehat{y} \notin \widehat{C}$, contradicting that C is unique. $\square$

Corollary 23 *If B and $\widehat{B}$ are bases. Let $\widehat{x} \in \widehat{B} \backslash B$, then $\exists x \in B \backslash \widehat{B}$ such that $B + \widehat{x} - x$ is a base.*

Proof Sketch. Follows from Corollary 22. $\square$

Definition 24 *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. The* rank function*, denoted by $r_{\mathcal{M}}$, of $\mathcal{M}$ is $r_{\mathcal{M}} : 2^S \mapsto \mathbb{Z}_+$, where for $X \subseteq S$, $r_{\mathcal{M}}(X)$ is the size of a maximum independent set contained in X.*

Note that the above definition assigns a unique number to each set X since all maximal independent sets contained in X have the same cardinality.

Proposition 25 *Given a matroid $\mathcal{M} = (S, \mathcal{I})$, the rank function $r_{\mathcal{M}}$ has the following properties:*

1. $0 \leq r_{\mathcal{M}}(X) \leq |X|$ *for all* $X \subseteq S$.
2. $r_{\mathcal{M}}$ *is submodular; i.e., for any* $X, Y \subseteq S$, $r_{\mathcal{M}}(X \cup Y) + r_{\mathcal{M}}(X \cap Y) \leq r_{\mathcal{M}}(X) + r_{\mathcal{M}}(Y)$.

Proof: Property 1 is by the definition of $r_{\mathcal{M}}$. To show the second property, we use the equivalent definition of submodularity; i.e., we show that if $X \subseteq Y$ and $z \in S$, then $r_{\mathcal{M}}(X + z) - r_{\mathcal{M}}(X) \geq r_{\mathcal{M}}(Y + z) - r_{\mathcal{M}}(Y)$. First notice that for any $X \subseteq S$ and $z \in S$, $r_{\mathcal{M}}(X + z) \leq r_{\mathcal{M}}(X) + 1$. Thus, we only need to show that if $r_{\mathcal{M}}(Y + z) - r_{\mathcal{M}}(Y) = 1$ then $r_{\mathcal{M}}(X + z) - r_{\mathcal{M}}(X) = 1$ for any $X \subseteq Y$.

If $r_{\mathcal{M}}(Y + z) - r_{\mathcal{M}}(Y) = 1$, then every base B of $Y + z$ contains z. Let $\widehat{B}$ be a base of X. Since $X \subseteq Y + z$, from Proposition 19, there exists a base $\bar{B}$ of $Y + z$ such that $\bar{B} \supseteq \widehat{B}$. Then $\widehat{B} + z$ is independent, implying $r_{\mathcal{M}}(X + z) - r_{\mathcal{M}}(X) = 1$ as $\widehat{B} + z$ is a base in $X + z$. $\square$

Definition 26 *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. For any $X \subseteq S$, the* span *of X, denoted by $span_M(X)$, is defined as $span_M(X) = \{y : y \in S, r_{\mathcal{M}}(X + y) = r_{\mathcal{M}}(X)\}$. A set $X \subseteq S$ is* spanning *if $span_M(X) = S$.*

Exercise 27 *Prove the following properties about the span function $span_M : 2^S \to 2^S$.*

- *If $T, U \subseteq S$ and $U \subseteq span_M(T)$ then $span_M(U) \subseteq span_M(T)$.*
- *If $T \subseteq S$, $t \in S \setminus T$ and $s \in span_M(T + t) \setminus span_M(T)$ then $t \in span_M(T + s)$.*

Definition 28 *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. A subset $X \subseteq S$ is a* flat *of $\mathcal{M}$ iff $span_M(X) = X$.*

Exercise 29 *Prove the following properties about flats.*

- *If F_1 and F_2 are flats then $F_1 \cap F_2$ is a flat.*
- *If F is a flat and $t \in S \setminus F$ and F' is a smallest flat containing $F + t$ then there is no flat F'' with $F \subset F'' \subset F'$.*

Remark 30 *We showed basic properties of bases, circuits, rank, span and flats of a matroid. One can show that a matroid can alternatively be specified by defining its bases or circuits or rank or span or flats that satisfy these properties. We refer the reader to [2].*

References

[1] J. Lee. "Matroids and the greedy algorithm". In *A First Course in Combinatorial Optimization*, Ch. 1, 49–74, Cambridge University Press, 2004.

[2] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, Chapter 39 (Vol B), Springer, 2003.

CS 598CSC: Combinatorial Optimization	**Lecture date: March 9, 2009**
Instructor: Chandra Chekuri	**Scribe: Sreeram Kannan**

1 Maximum Weight Independent Set in a Matroid, Greedy Algorithm, Independence and Base Polytopes

1.1 More on Matroids

We saw the definition of base, circuit, rank, span and flat of matroids last lecture. We begin this lecture by studying some more basic properties of a matroid.

Exercise 1 *Show that a set $I \subseteq S$ is independent in a matroid $\mathcal{M}$ iff $\forall y \in I$, there exists a flat F such that $I - y \subseteq F$ and $y \notin F$.*

Definition 2 *A matroid $\mathcal{M} = (S, \mathcal{I})$ is defined as connected if $r_{\mathcal{M}}(U) + r_{\mathcal{M}}(S \backslash U) > r_{\mathcal{M}}(S)$ for each $U \subseteq S, U \neq \emptyset$. Equivalently, for each $s, t \in S, s \neq t$, there is a circuit containing both s, t.*

1.2 Operations on a Matroid

1.2.1 Dual

Given a matroid $\mathcal{M} = (S, \mathcal{I})$ its dual matroid $\mathcal{M}^* = (S, \mathcal{I}^*)$ is defined as follows:

$$\mathcal{I}^* = \{I \in S \mid S \backslash I \text{ is spanning in } \mathcal{M}, \text{ i.e., } r_{\mathcal{M}}(S \backslash I) = r_{\mathcal{M}}(S)\}.$$

Exercise 3 *Verify that $(S, \mathcal{I}^*)$ is indeed a matroid.*

The following facts are easy to prove:

1. $\mathcal{M}^{**} = \mathcal{M}$.
2. B is a base of $\mathcal{M}^*$ iff $S \backslash B$ is a base of $\mathcal{M}$.
3. $r_{\mathcal{M}^*}(U) = |U| + r_{\mathcal{M}}(S \backslash U) - r_{\mathcal{M}}(S)$

Remark 4 *Having an independence or rank oracle for $\mathcal{M}$ implies one has it for $\mathcal{M}^*$ too.*

Exercise 5 *Prove that $\mathcal{M}$ is connected if and only if $\mathcal{M}^*$ is.*

Hint: Use the third property from above.

1.2.2 Deletion

Definition 6 *Given a matroid $\mathcal{M} = (S, \mathcal{I})$ and $e \in S$, deleting e from $\mathcal{M}$ generates a new matroid*

$$\mathcal{M}' = \mathcal{M} \backslash e = (S - e, \mathcal{I}'), \tag{1}$$

where $\mathcal{I}' = \{I - e | I \in \mathcal{I}\}$. For $Z \subseteq S$, the matroid $\mathcal{M} \backslash Z$ is obtained similarly by restricting the matroid $\mathcal{M}$ to $S \backslash Z$.

1.2.3 Contraction

Definition 7 *Given a matroid $\mathcal{M} = (S, \mathcal{I})$ and $e \in S$, the contraction of the matroid with respect to e is defined as $\mathcal{M}/e = (\mathcal{M}^* \backslash e)^*$, i.e., it is obtained by deleting e in the dual and taking its dual. Similarly for a set $Z \subseteq S$, we can similarly define $\mathcal{M}/Z = (\mathcal{M}^* \backslash Z)^*$*

It is instructive to view the contraction operation from a graph theoretic perspective. If e is a loop, $\mathcal{M}/e = \mathcal{M}\backslash e$, else $\mathcal{M}/e = (S - e, \mathcal{I}')$ where

$$\mathcal{I}' = \{I \in S - e | I + e \in \mathcal{I}\}.$$

For the case of contracting a subset Z, we can take a base $X \subseteq Z$ and $M/Z = (S\backslash Z, \mathcal{I}')$, where

$$\mathcal{I}' = \{I \in S\backslash Z | I \cup Z \in \mathcal{I}\}.$$

Also

$$r_{\mathcal{M}/Z}(X) = r_{\mathcal{M}}(X \cup Z) - r_{\mathcal{M}}(Z)$$

Exercise 8 *Show that*

$$\mathcal{M}/\{e_1, e_2\} = (\mathcal{M}/e_1)/e_2 = (\mathcal{M}/e_2)/e_1. \tag{2}$$

1.2.4 Minor

Similar to graph minors, matroid minors can be defined, and they play an important role in characterizing the type of matroids.

Definition 9 *A matroid $\mathcal{M}'$ is a minor of a matroid $\mathcal{M}$ if $\mathcal{M}'$ is obtained from $\mathcal{M}$ by a sequence of contractions and deletions.*

1.3 Maximum Weight Independent Set in a Matroid

Matroids have some important algorithmic properties, the simplest one being that the problem of determining the maximum weight independent set in a matroid can be solved using a greedy algorithm. The maximum weight independent set problem is stated as follows: Given $\mathcal{M} = (S, \mathcal{I})$ and $w : S \to R$, output

$$\arg\max_{I \in \mathcal{I}} w(I). \tag{3}$$

1.3.1 Greedy Algorithm

The greedy algorithm can be stated as follows:

1. Discard all $e \in S$ where $w(e) \leq 0$ or e is a loop.
2. Let $S = \{e_1, ..., e_n\}$ such that $w(e_1) \geq w(e_2) ... \geq w(e_n)$.
3. $X \leftarrow \emptyset$.
4. For $i = 1$ to n, do
   if($X + e_i \in \mathcal{I}$) then $X \leftarrow X + e_i$.

5. Output X.

The above algorithm had to specifically take care of loops and edges with non-negative weights. An equivalent algorithm without this hassle is:

1. Let $S = \{e_1, ..., e_n\}$ such that $w(e_1) \geq w(e_2) ... \geq w(e_n)$.

2. $X \leftarrow \emptyset$.

3. For $i = 1$ to n, do
   if$(X + e_i \in \mathcal{I})$ and $w(X + e_i) \geq w(X)$, then $X \leftarrow X + e_i$.

4. Output X.

Theorem 10 *The greedy algorithm outputs an optimum solution to the maximum weight independent set problem.*

Proof: Without loss of generality, assume $w(e) > 0, \forall e \in S$ and that there are no loops.

Claim 11 *There exists an optimum solution that contains e_1.*

Assuming this claim is true, we can use induction to show that greedy algorithm has to yield an optimum solution. This is because the greedy algorithm is recursively finding an optimum solution in the matroid $\mathcal{M}' = (S - e, \mathcal{I}')$ where $\mathcal{I}' = \{I - e_1 | I \in \mathcal{I}'\}$.

To prove the claim, let I^* be an optimum solution. If $e_1 \in I^*$, we are done, else, we can see that $I^* + e_1$ is not independent, otherwise $w(I^* + e_1) > w(I^*)$. Thus $I^* + e_1$ contains a circuit, and hence $\exists e \in I^*$ such that $I^* - e + e_1 \in \mathcal{I}$ (See Corollary 22 in the last lecture). $w(I^* - e + e_1) \geq w(I^*)$ since $w(e_1)$ has the largest weight among all the elements in the set. Thus there is an optimum solution $I^* - e + e_1$ that contains e_1. $\square$

Remark 12 *If all weights are non-negative then it is easy to see that the greedy algorithm outputs a base of $\mathcal{M}$. We can adapt the greedy algorithm to solve the maximum weight base problem by making all weights non-negative by adding a large constant to each of the weights. Thus max-weight base problem, and equivalently min-cost base problem can be solved (by taking the weights to be the negative of the costs).*

Remark 13 *Kruskal's algorithm for finding the maximum weight spanning tree can be interpreted as a special case of the greedy algorithm for matroids when applied to the graphic matroid corresponding to the graph.*

1.3.2 Oracles for a Matroid

Since the set of all independence sets could be exponential in $|S|$, it is infeasible to use this representation. Instead we resort to one of the two oracles in order to efficiently solve optimization problems:

- An independence oracle that given $A \subseteq S$, returns whether $A \in \mathcal{I}$ or not.

- A rank oracle that given $A \subseteq S$, returns $r_{\mathcal{M}}(A)$.

These two oracles are equivalent in the sense that one can be recovered from the other in polynomial time.

1.4 Matroid Polytopes

Edmonds utilized the Greedy algorithm in proving the following theorem:

Theorem 14 *The following polytope is the convex hull of the characteristic vectors of the independent sets of a matroid $\mathcal{M} = (S, \mathcal{I})$ with rank function $r_{\mathcal{M}} : 2^S \rightarrow \mathcal{Z}_+$,*

$$x(A) \leq r_{\mathcal{M}}(A) \quad \forall A \subseteq S,$$
$$x(A) \geq 0.$$

Also, the system of inequalities described above is TDI.

Proof: We will show that the above system of inequalities is TDI (Totally Dual Integral), which will in turn imply that the polytope is integral since $r_{\mathcal{M}}(.)$ is integer valued.

Let us consider the primal and dual linear programs for some integral weight vector $w : S \rightarrow \mathcal{Z}$. We will show that the solution picked by Greedy algorithm is the optimal solution for primal by producing a dual solution that attains the same value. Alternately we could show that the dual solution and the primal solution picked by Greedy satisfy complementary slackness.

$$\begin{aligned} \text{Primal:} \quad \max \sum_{e \in S} w(e)x(e) & \\ x(A) \leq r(A), & \quad \forall\ A \subseteq S \\ x \geq 0 & \\ \text{Dual:} \quad \min \sum_{A \in S} r(A)y(A) & \\ \sum_{A: e \in A} y(A) \geq w(e), & \quad \forall e \in S \\ y \geq 0 & \end{aligned}$$

Let $S = \{e_1, ..., e_n\}$ such that $w(e_1) \geq w(e_2) ... \geq w(e_n) \geq 0$, since setting $w(e_i) = 0$ whenever $w(e_i) < 0$ does not alter the solution of the primal or dual. Define $A_j = \{e_1, ..., e_j\}$ with $A_0 = \emptyset$. It is easy to see that $r(A_j) = r(A_{j-1}) + 1$ iff e_j is picked by Greedy. Consider the following dual solution

$$\begin{aligned} y(A_j) &= w(e_j) - w(e_{j+1}), j < n \\ &= w(e_n), j = n \\ y(A) &= 0, \text{ if } A \neq A_j \text{ for some } j \end{aligned}$$

Claim 15 *y is dual feasible.*

Clearly, $y \geq 0$. Since $w(e_j) \geq w(e_{j-1}) \forall j$, for any e_i,

$$\begin{aligned} \sum_{A: e_i \in A} y(A) &= \sum_{j \geq i} y(A_i) \\ &= \sum_{j=i}^{n-1} \{w(e_j) - w(e_{j+1})\} + w(e_n) \\ &= w(e_i) \end{aligned}$$

Define $I = \{i | e_i \text{ is picked by Greedy}\}$. As we noted earlier, $i \in I \iff r(A_i) = r(A_{i-1}) + 1$.

Claim 16
$$\sum_{i \in I} w(e_i) = \sum_{A \subseteq S} r(A) y(A)$$

$$\begin{aligned}
\sum_{i \in I} w(e_i) &= \sum_{i \in I} w(e_i)(r(A_i) - r(A_{i-1})) \\
&= \sum_{j=1}^{n} w(e_j)(r(A_j) - r(A_{j-1})) \\
&= w(e_n) y(A_n) + \sum_{j=1}^{n-1} (w(e_j) - w(e_{j+1})) r(A_j) \\
&= \sum_{j=1}^{n} y(A_j) r(A_j) \\
&= \sum_{A \subseteq S} r(A) y(A)
\end{aligned}$$

Thus y has to be dual optimal, and the solution produced by Greedy has to be primal optimal. This means that the dual optimal solution is integral whenever w is integral, and therefore the system is TDI.

□

Corollary 17 *The base polytope of $\mathcal{M} = (S, I)$, i.e., the convex hull of the bases of $\mathcal{M}$ is determined by*

$$\begin{aligned}
x(A) &\leq r(A), \forall A \subseteq S, \\
x(S) &= r(S) \\
x &\geq 0
\end{aligned}$$

1.4.1 Spanning Set Polytope

Another polytope associated with a matroid is the spanning set polytope, which is the convex hull of the incidence vectors of all spanning sets.

Theorem 18 *The spanning set polytope of a matroid $\mathcal{M} = (S, I)$ with rank function $r_{\mathcal{M}}$ is determined by*

$$\begin{aligned}
0 \leq x(e) \leq 1, &\qquad \forall e \in S \\
x(U) \geq r_{\mathcal{M}}(S) - r_{\mathcal{M}}(S \backslash U), &\qquad \forall U \subseteq S.
\end{aligned}$$

Proof: A given set $A \subseteq S$ is spanning in $\mathcal{M}$ iff $S \backslash A$ is independent in $\mathcal{M}^*$. Thus $x \in \mathcal{P}_{\text{spanning}}(\mathcal{M})$ iff $1 - x \in \mathcal{P}_{\text{independence}}(\mathcal{M}^*)$. Now, by the relation between the ranks of dual matroids,

$$r_{\mathcal{M}^*}(U) = |U| + r_{\mathcal{M}}(S \backslash U) - r_{\mathcal{M}}(S).$$

Thus $1 - x \in \mathcal{P}_{\text{independence}}(\mathcal{M}^*)$ iff

$$\begin{array}{rcl} 1 - x & \geq & 0, \\ |U| - x(U) & \leq & r_{\mathcal{M}^*}(U) = |U| + r_{\mathcal{M}}(S \backslash U) - r_{\mathcal{M}}(S), \end{array}$$

which matches the statement of the theorem. □

1.4.2 Separation Oracle

We have now determined that the independence polytope of a matroid is given by the linear conditions $x \geq 0$ and $x(U) \leq r_{\mathcal{M}}(U), U \subseteq S$. The greedy algorithm allows us to optimize over the polytope and by the equivalence between optimization and separation, there is a polynomial time separation oracle for the polytope. It is instructive to consider it explicitly.

For the separation problem, given a test vector $w : S \rightarrow \mathbb{R}$, we need to find out if $w \in \mathcal{P}_{\text{independence}}(\mathcal{M})$. We can easily test for non-negativity. To test the second condition, it is sufficient to check that $\min_{A \subseteq S}(r_{\mathcal{M}}(A) - w(A)) \geq 0$. In fact any violated inequality in the linear system can be found by constructing the set

$$U = \arg\min_{A \subseteq S}(r_{\mathcal{M}}(A) - w(A)).$$

Define $f : 2^S \rightarrow \mathbb{R}$, where $f(A) = r_{\mathcal{M}}(A) - w(A)$. f is a submodular set function since $r(.)$ is the submodular rank function and $-w(.)$ is modular. Thus if we can minimize an arbitrary submodular function specified by a value oracle, we can use the same for separating over a matroid polytope. However, there is a more efficient algorithm for separating over the independence polytopes given by Cunningham. See [1] for details.

References

[1] Lex Schrijver, "Combinatorial Optimization: Polyhedra and Efficiency, Vol. B," Springer-Verlag 2003.

4. Lecture notes on matroid optimization

4.1 Definition of a Matroid

Matroids are combinatorial structures that generalize the notion of linear independence in matrices. There are many equivalent definitions of matroids, we will use one that focus on its *independent sets*. A matroid M is defined on a finite ground set E (or $E(M)$ if we want to emphasize the matroid M) and a collection of subsets of E are said to be *independent*. The family of independent sets is denoted by $\mathcal{I}$ or $\mathcal{I}(M)$, and we typically refer to a matroid M by listing its ground set and its family of independent sets: $M = (E, \mathcal{I})$. For M to be a matroid, $\mathcal{I}$ must satisfy two main axioms:

(I_1) if $X \subseteq Y$ and $Y \in \mathcal{I}$ then $X \in \mathcal{I}$,

(I_2) if $X \in \mathcal{I}$ and $Y \in \mathcal{I}$ and $|Y| > |X|$ then $\exists e \in Y \setminus X : X \cup \{e\} \in \mathcal{I}$.

In words, the second axiom says that if X is independent and there exists a larger independent set Y then X can be extended to a larger independent by adding an element of $Y \setminus X$. Axiom (I_2) implies that every *maximal* (inclusion-wise) independent set is maximum; in other words, all maximal independent sets have the same cardinality. A maximal independent set is called a *base* of the matroid.

Examples.

- One trivial example of a matroid $M = (E, \mathcal{I})$ is a **uniform** matroid in which

  $$\mathcal{I} = \{X \subseteq E : |X| \leq k\},$$

  for a given k. It is usually denoted as $U_{k,n}$ where $|E| = n$. A base is any set of cardinality k (unless $k > |E|$ in which case the only base is $|E|$).

  A **free** matroid is one in which all sets are independent; it is $U_{n,n}$.

- Another is a **partition** matroid in which E is partitioned into (disjoint) sets $E_1, E_2, \cdots, E_l$ and

  $$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ for all } i = 1, \cdots, l\},$$

  for some given parameters $k_1, \cdots, k_l$. As an exercise, let us check that (I_2) is satisfied. If $X, Y \in \mathcal{I}$ and $|Y| > |X|$, there must exist i such that $|Y \cap E_i| > |X \cap E_i|$ and this means that adding any element e in $E_i \cap (Y \setminus X)$ to X will maintain independence.

  Observe that M would *not* be a matroid if the sets E_i were *not* disjoint. For example, if $E_1 = \{1, 2\}$ and $E_2 = \{2, 3\}$ with $k_1 = 1$ and $k_2 = 1$ then both $Y = \{1, 3\}$ and $X = \{2\}$ have at most one element of each E_i, but one can't find an element of Y to add to X.

- **Linear** matroids (or representable matroids) are defined from a matrix A, and this is where the term *matroid* comes from. Let E denote the index set of the columns of A. For a subset X of E, let A_X denote the submatrix of A consisting only of those columns indexed by X. Now, define

  $$\mathcal{I} = \{X \subseteq E : rank(A_X) = |X|\},$$

  i.e. a set X is independent if the corresponding columns are linearly independent. A base B corresponds to a linearly independent set of columns of cardinality $rank(A)$.

  Observe that (I_1) is trivially satisfied, as if columns are linearly independent, so is a subset of them. (I_2) is less trivial, but corresponds to a fundamental linear algebra property. If A_X has full column rank, its columns span a space of dimension $|X|$, and similarly for Y, and therefore if $|Y| > |X|$, there must exist a column of A_Y that is not in the span of the columns of A_X; adding this column to A_X increases the rank by 1.

  A linear matroid can be defined over any field $\mathbb{F}$ (not just the reals); we say that the matroid is **representable over** $\mathbb{F}$. If the field is $\mathbb{F}_2$ (field of 2 elements with operations (mod 2)) then the matroid is said to be **binary**. If the field is $\mathbb{F}_3$ then the matroid is said to be **ternary**.

  For example, the binary matroid corresponding to the matrix

  $$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

  corresponds to $U_{2,3}$ since the sum of the 3 columns is the 0 vector when taking components modulo 2. If A is viewed over the reals or over $\mathbb{F}_3$ then the matroid is the free matroid on 3 elements.

  Not every matroid is linear. Among those that are linear, some can be represented over some fields $\mathbb{F}$ but not all. For example, there are binary matroids which are not ternary and vice versa (for example, $U_{2,4}$ is ternary but not binary). Matroids which can be represented over *any* field are called **regular**.

- Here is an example of something that is not a matroid. Take a graph $G = (V, E)$, and let $\mathcal{I} = \{F \subseteq E : F \text{ is a matching}\}$. This is not a matroid since (I_2) is not necessarily satisfied ((I_1) is satisfied[1], however). Consider, for example, a graph on 4 vertices and let $X = \{(2,3)\}$ and $Y = \{(1,2), (3,4)\}$. Both X and Y are matchings, but one cannot add an edge of Y to X and still have a matching.

- There is, however, another matroid associated with matchings in a (general, not necessarily bipartite) graph $G = (V, E)$, but this time the ground set of M corresponds to V. In the **matching matroid**, $\mathcal{I} = \{S \subseteq V : S \text{ is covered by some matching } M\}$. In this definition, the matching does not need to cover precisely S; other vertices can be covered as well.

[1]When (I_1) alone is satisfied, $(E, \mathcal{I})$ is called an *independence system.*

- A very important class of matroids in combinatorial optimization is the class of **graphic** matroids (also called cycle matroids). Given a graph $G = (V, E)$, we define independent sets to be those subsets of edges which are forests, i.e. do not contain any cycles. This is called the graphic matroid $M = (E, \mathcal{I})$, or $M(G)$.

  (I_1) is clearly satisfied. To check (I_2), first notice that if F is a forest then the number of connected components of the graph (V, F) is given by $K(V, F) = |V| - |F|$. Therefore, if X and Y are 2 forests and $|Y| > |X|$ then $K(V, Y) < K(V, X)$ and therefore there must exist an edge of $Y \setminus X$ which connects two different connected components of X; adding this edge to X results in a larger forest. This shows (I_2).

  If the graph G is connected, any base will correspond to a spanning tree T of the graph. If the original graph is disconnected then a base corresponds to taking a spanning tree in each connected component of G.

  A graphic matroid is a linear matroid. We first show that the field $\mathbb{F}$ can be chosen to be the reals. Consider the matrix A with a row for each vertex $i \in V$ and a column for each edge $e = (i, j) \in E$. In the column corresponding to (i, j), all entries are 0, except for a 1 in i or j (arbitrarily) and a -1 in the other. To show equivalence between the original matroid M and this newly constructed linear matroid M', we need to show that any independent set for M is independent in M' and vice versa. This is left as an exercise.

  In fact, a graphic matroid is *regular*; it can be represented over any field $\mathbb{F}$. To obtain a representation for a field $\mathbb{F}$, one simply needs to take the representation given above for $\mathbb{R}$ and simply view/replace all -1 by the additive inverse of 1 (i.e. by $p - 1$ for $\mathbb{F}_p$).

4.1.1 Circuits

A minimal (inclusionwise) dependent set in a matroid is called a *circuit*. In a graphic matroid $M(G)$, a circuit will be the usual notion of a cycle in the graph G; to be dependent in the graphic matroid, one needs to contain a cycle and the minimal sets of edges containing a cycle are the cycles themselves. In a partition matroid, a circuit will be a set $C \subseteq E_i$ with $|C \cap E_i| = k_i + 1$.

By definition of a circuit C, we have that if we remove any element of a circuit then we get an independent set. A crucial property of circuit is given by the following property,

Theorem 4.1 (Unique Circuit Property) *Let $M = (E, \mathcal{I})$ be a matroid. Let $S \in \mathcal{I}$ and e such that*[2] *$S + e \notin \mathcal{I}$. Then there exists a unique circuit $C \subseteq S + e$.*

The unicity is very important. Indeed, if we consider any $f \in C$ where C is this unique circuit then we have that $C + e - f \in \mathcal{I}$. Indeed, if $C + e - f$ was dependent, it would contain a circuit C' which is distinct from C since $f \notin C'$, a contradiction.

[2] For a set S and an element e, we often write $S + e$ for $S \cup \{e\}$ and $S - e$ for $S \setminus \{e\}$.

As a special case of the theorem, consider a graphic matroid. If we add an edge to a forest and the resulting graph has a cycle then it has a unique cycle.

Proof:

Suppose $S+e$ contains more than one circuit, say C_1 and C_2 with $C_1 \neq C_2$. By minimality of C_1 and C_2, we have that there exists $f \in C_1 \setminus C_2$ and $g \in C_2 \setminus C_1$. Since $C_1 - f \in \mathcal{I}$ (by minimality of the circuit C_1), we can extend it to a maximal independent set X of $S + e$. Since S is also independent, we must have that $|X| = |S|$ and since $e \in C_1 - f$, we must have that $X = S + e - f \in \mathcal{I}$. But this means that $C_2 \subseteq S + e - f = X$ which is a contradiction since C_2 is dependent. $\triangle$

Exercise 4-1. Show that any partition matroid is also a linear matroid over $\mathbb{F} = \mathbb{R}$. (No need to give a precise matrix A representing it; just argue its existence.)

Exercise 4-2. Prove that a matching matroid is indeed a matroid.

Exercise 4-3. Show that $U_{2,4}$ is representable over $\mathbb{F}_3$.

Exercise 4-4. Consider the linear matroid (over the reals) defined by the 3×5 matrix:

$$A = \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 & -1 \\ 1 & 2 & 0 & 1 & -1 \end{pmatrix}.$$

The ground set $E = \{1, 2, 3, 4, 5\}$ has cardinality 5, corresponds to the columns of A, and the independent sets are the set of columns which are linearly independent (over the reals).

1. Give all bases of this matroid.
2. Give all circuits of this matroid.
3. Choose a base B and an element e not in B, and verify the unique circuit property for $B + e$.

Exercise 4-5. Given a family $A_1, A_2, \cdots, A_n$ of sets (they are not necessarily disjoint), a *transversal* is a set T such that $T = \{a_1, a_2, \cdots, a_n\}$, the a_i's are distinct, and $a_i \in A_i$ for all i. A partial transversal is a transversal for $A_{i_1}, A_{i_2}, \cdots, A_{i_k}$ for some subfamily of the A_i's.

Show that the family of all partial transversals forms a matroid (on the ground set $E = \cup A_i$). (Hint: Think of bipartite matchings.)

Exercise 4-6. Let $M = (E, \mathcal{I})$ be a matroid. Let $k \in \mathbb{N}$ and define

$$\mathcal{I}_k = \{X \in \mathcal{I} : |I| \leq k\}.$$

Show that $M_k = (E, \mathcal{I}_k)$ is also a matroid. This is known as a truncated matroid.

Exercise 4-7. A family $\mathcal{F}$ of sets is said to be *laminar* if, for any two sets $A, B \in \mathcal{F}$, we have that either (i) $A \subseteq B$, or (ii) $B \subseteq A$ or (iii) $A \cap B = \emptyset$. Suppose that we have a laminar family $\mathcal{F}$ of subsets of E and an integer $k(A)$ for every set $A \in \mathcal{F}$. Show that $(E, \mathcal{I})$ defines a matroid (a *laminar* matroid) where:

$$\mathcal{I} = \{X \subseteq E : |X \cap A| \leq k(A) \text{ for all } A \in \mathcal{F}\}.$$

4.2 Matroid Optimization

Given a matroid $M = (E, \mathcal{I})$ and a cost function $c : E \to \mathbb{R}$, we are interested in finding an independent set S of M of maximum total cost $c(S) = \sum_{e \in S} c(e)$. This is a fundamental problem.

If all $c(e) \geq 0$, the problem is equivalent to finding a maximum cost *base* in the matroid. If $c(e) < 0$ for some element e then, because of (I_1), e will not be contained in any optimum solution, and thus we could eliminate such an element from the ground set. In the special case of a graphic matroid $M(G)$ defined on a connected graph G, the problem is thus equivalent to the maximum spanning tree problem which can be solved by a simple greedy algorithm. This is actually the case for any matroid and this is the topic of this section.

The greedy algorithm we describe actually returns, for every k, a set S_k which maximizes $c(S)$ over all independent sets of size k. The overall optimum can thus simply be obtained by outputting the best of these. The greedy algorithm is the following:

- ▷ Sort the elements (and renumber them) such that $c(e_1) \geq c(e_2) \geq \cdots \geq c(e_{|M|})$
- ▷ $S_0 = \emptyset$, k=0
- ▷ For $j = 1$ to $|E|$
  - ▷ if $S_k + e_j \in \mathcal{I}$ then
    - ▷ $k \leftarrow k + 1$
    - ▷ $S_k \leftarrow S_{k-1} + e_j$
    - ▷ $s_k \leftarrow e_j$
- ▷ Output S_1, $S_2, \cdots, S_k$

Theorem 4.2 *For any matroid $M = (E, \mathcal{I})$, the greedy algorithm above finds, for every k, an independent set S_k of maximum cost among all independent sets of size k.*

Proof: Suppose not. Let $S_k = \{s_1, s_2, \cdots, s_k\}$ with $c(s_1) \geq c(s_2) \geq \cdots \geq c(s_k)$, and suppose T_k has greater cost $(c(T_k) > c(S_k))$ where $T_k = \{t_1, t_2, \cdots, t_k\}$ with $c(t_1) \geq c(t_2) \geq \cdots \geq c(t_k)$. Let p be the first index such that $c(t_p) > c(s_p)$. Let $A = \{t_1, t_2, \cdots, t_p\}$ and $B = \{s_1, s_2, \cdots, s_{p-1}\}$. Since $|A| > |B|$, there exists $t_i \notin B$ such that $B + t_i \in \mathcal{I}$. Since $c(t_i) \geq c(t_p) > c(s_p)$, t_i should have been selected when it was considered. To be more precise and detailed, when t_i was considered, the greedy algorithm checked whether t_i could be added to the current set at the time, say S. But since $S \subseteq B$, adding t_i to S should have resulted in an independent set (by (I_1)) since its addition to B results in an independent set. This gives the contradiction and completes the proof. △

Observe that, as long as $c(s_k) \geq 0$, we have that $c(S_k) \geq c(S_{k-1})$. Therefore, to find a maximum cost set over all independent sets, we can simply replace the loop

$\triangleright$ For $j = 1$ to $|E|$

by

$\triangleright$ For $j = 1$ to q

where q is such that $c(e_q) \geq 0 > c(e_{q+1})$, and output the last S_k.

For the maximum cost spanning tree problem, the greedy algorithm reduces to Kruskal's algorithm which considers the edges in non-increasing cost and add an edge to the previously selected edges if it does not form a cycle.

One can show that the greedy algorithm actually characterizes matroids. If M is an independence system, i.e. it satisfies (I_1), then M is a matroid if and only if the greedy algorithm finds a maximum cost set of size k for every k and every cost function.

Exercise 4-8. We are given n jobs that each take one unit of processing time. All jobs are available at time 0, and job j has a profit of c_j and a deadline d_j. The profit for job j will only be earned if the job completes by time d_j. The problem is to find an ordering of the jobs that maximizes the total profit. First, prove that if a subset of the jobs can be completed on time, then they can also be completed on time if they are scheduled in the order of their deadlines. Now, let $E(M) = \{1, 2, \cdots, n\}$ and let $\mathcal{I}(M) = \{J \subseteq E(M) : J$ can be completed on time $\}$. Prove that M is a matroid and describe how to find an optimal ordering for the jobs.

4.3 Rank Function of a Matroid

Similarly to the notion of rank for matrices, one can define a rank function for any matroid. The rank function of M, denoted by either $r(\cdot)$ or $r_M(\cdot)$, is defined by:

$$r_M : 2^E \to \mathbb{N} : r_M(X) = \max\{|Y| : Y \subseteq X, Y \in \mathcal{I}\}.$$

Here are a few specific rank functions:

- For a linear matroid, the rank of X is precisely the rank in the linear algebra sense of the matrix A_X corresponding to the columns of A in X.
- For a partition matroid $M = (E, \mathcal{I})$ where

  $$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ for } i = 1, \cdots, l\}$$

  (the E_i's forming a partition of E) its rank function is given by:

  $$r(X) = \sum_{i=1}^{l} \min(|E_i \cap X|, k_i).$$

- For a graphic matroid $M(G)$ defined on graph $G = (V, E)$, the rank function is equal to:
$$r_{M(G)}(F) = n - K(V, F),$$
where $n = |V|$ and $K(V, F)$ denotes the number of connected components (including isolated vertices) of the graph with edges F.

The rank function of any matroid $M = (E, \mathcal{I})$ has the following properties:

(R_1) $0 \leq r(X) \leq |X|$ and is integer valued for all $X \subseteq E$

(R_2) $X \subseteq Y \Rightarrow r(X) \leq r(Y)$,

(R_3) $r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)$.

The last property is called *submodularity* and is a key concept in combinatorial optimization. It is clear that, as defined, any rank function satisfies (R_1) and (R_2). Showing that the rank function satisfies submodularity needs a proof.

Lemma 4.3 *The rank function of any matroid is submodular.*

Proof: Consider any two sets $X, Y \subseteq$. Let J be a maximal independent subset of $X \cap Y$; thus, $|J| = r(X \cap Y)$. By (I_2), J can be extended to a maximal (thus maximum) indenpendent subset of X, call it J_X. We have that $J \subseteq J_X \subseteq X$ and $|J_X| = r(X)$. Furthermore, by maximality of J within $X \cap Y$, we know

$$J_X \setminus Y = J_X \setminus J. \tag{1}$$

Now extend J_X to a maximal independent set J_{XY} of $X \cup Y$. Thus, $|J_{XY}| = r(X \cup Y)$.

In order to be able to prove that

$$r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)$$

or equivalently

$$|J_X| + r(Y) \geq |J| + |J_{XY}|,$$

we need to show that $r(Y) \geq |J| + |J_{XY}| - |J_X|$. Observe that $J_{XY} \cap Y$ is independent (by (I_1)) and a subset of Y, and thus $r(Y) \geq |J_{XY} \cap Y|$. Observe now that

$$J_{XY} \cap Y = J_{XY} \setminus (J_X \setminus Y) = J_{XY} \setminus (J_X \setminus J),$$

the first equality following from the fact that J_X is a maximal independent subset of X and the second equality by (1). Therefore,

$$r(Y) \geq |J_{XY} \cap Y| = |J_{XY} \setminus (J_X \setminus J)| = |J_{XY}| - |J_X| + |J|,$$

proving the lemma. △

4.3.1 Span

The following definition is also motivated by the linear algebra setting.

Definition 4.1 *Given a matroid $M = (E, \mathcal{I})$ and given $S \subseteq E$, let*

$$span(S) = \{e \in E : r(S \cup \{e\}) = r(S)\}.$$

Observe that $S \subseteq span(S)$. We claim that $r(S) = r(span(S))$; in other words, if adding an element to S does not increase the rank, adding many such elements also does not increase the rank. Indeed, take a maximal independent subset of S, say J. If $r(span(S)) > |J|$ then there exists $e \in span(S) \setminus J$ such that $J + e \in \mathcal{I}$. Thus $r(S+e) \geq r(J+e) = |J|+1 > |J| = r(S)$ contradicting the fact that $e \in span(S)$.

Definition 4.2 *A set S is said to be closed if $S = span(S)$.*

Exercise 4-9. Given a matroid M with rank function r and given an integer $k \in \mathbb{N}$, what is the rank function of the truncated matroid M_k (see Exercise 4-6 for a definition).

Exercise 4-10. What is the rank function of a laminar matroid, see exercise 4-7?

4.4 Matroid Polytope

Let

$$X = \{\chi(S) \in \{0,1\}^{|E|} : S \in \mathcal{I}\}$$

denote the incidence (or characteristic) vectors of all independent sets of a matroid $M = (E, \mathcal{I})$, and let the *matroid polytope* be defined as $conv(X)$. In this section, we provide a complete characterization of $conv(X)$ in terms of linear inequalities. In addition, we illustrate the different techniques proposed in the polyhedral chapter for proving a complete description of a polytope.

Theorem 4.4 *Let*

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \leq r(S) & \forall S \subseteq E \\ x_e \geq 0 & \forall e \in E\} \end{array}$$

where $x(S) := \sum_{e \in S} x_e$. Then $conv(X) = P$.

It is clear that $conv(X) \subseteq P$ since $X \subseteq P$. The harder part is to show that $P \subseteq conv(X)$. In the next three subsections, we provide three different proofs based on the three techniques to prove complete polyhedral descriptions.

4.4.1 Algorithmic Proof

Here we provide an algorithmic proof based on the greedy algorithm. From $conv(X) \subseteq P$, we know that

$$\max\{c^T x : x \in X\} = \max\{c^T x : x \in conv(X)\} \leq \max\{c^T x : \begin{array}{ll} x(S) \leq r(S) & S \subseteq E \\ x_e \geq 0 & e \in E\}. \end{array}$$

Using LP duality, we get that this last expression equals:

$$\min\{\textstyle\sum_S r(S) y_S : \begin{array}{ll} \sum_{S: e \in S} y_S \geq c(e) & \forall e \in E \\ y_S \geq 0 & S \subseteq E\}. \end{array}$$

Our goal now is, for any cost function c, to get an independent set S and a dual feasible solution y such that $c^T \chi(S) = \sum_S r(S) y_S$ which proves that $conv(X) = P$.

Consider any cost function c. We know that the maximum cost independent set can be obtained by the greedy algorithm. More precisely, it is the last set S_k returned by the greedy algorithm when we consider only those elements up to e_q where $c(e_q) \geq 0 \geq c(e_{q+1})$. We need now to exhibit a dual solution of the same value as S_k. There are exponentially many variables in the dual, but this is not a problem. In fact, we will set most of them to 0.

For any index $j \leq k$, we have $S_j = \{s_1, s_2, \cdots, s_j\}$, and we define U_j to be all elements in our ordering up to and excluding s_{j+1}, i.e. $U_j = \{e_1, e_2, \cdots, e_l\}$ where $e_{l+1} = s_{j+1}$. In other words, U_j is all the elements in the ordering just before s_{j+1}. One important property of U_j is that

$$r(U_j) = r(S_j) = j.$$

Indeed, by independence $r(S_j) = |S_j| = j$, and by (R_1), $r(U_j) \geq r(S_j)$. If $r(U_j) > r(S_j)$, there would be an element say $e_p \in U_j \setminus S_j$ such that $S_j \cup \{e_p\} \in \mathcal{I}$. But the greedy algorithm would have selected that element (by (I_1)) contradicting the fact that $e_p \in U_j \setminus S_j$.

Set the non-zero entries of y_S in the following way. For $j = 1, \cdots, k$, let

$$y_{U_j} = c(s_j) - c(s_{j+1}),$$

where it is understood that $c(s_{k+1}) = 0$. By the ordering of the $c(\cdot)$, we have that $y_S \geq 0$ for all S. In addition, for any $e \in E$, we have that

$$\sum_{S: e \in S} y_S = \sum_{j=t}^{k} y_{U_j} = c(s_t) \geq c(e),$$

where t is the least index such that $e \in U_t$ (implying that e does not come before s_t in the ordering). This shows that y is a feasible solution to the dual. Moreover, its dual value is:

$$\sum_S r(S) y_S = \sum_{j=1}^{k} r(U_j) y_{U_j} = \sum_{j=1}^{k} j(c(s_j) - c(s_{j+1})) = \sum_{j=1}^{k} (j-(j-1)) c(s_j) = \sum_{j=1}^{k} c(s_j) = c(S_k).$$

This shows that the dual solution has the same value as the independent set output by the greedy algorithm, and this is true for all cost functions. This completes the algorithmic proof.

4.4.2 Vertex Proof

Here we will focus on any vertex x of

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \leq r(S) & \forall S \subseteq E \\ x_e \geq 0 & \forall e \in E\} \end{array}$$

and show that x is an integral vector. Since $x(\{e\}) \leq r(\{e\}) \leq 1$, we get that $x \in \{0,1\}^{|E|}$ and thus it is the incidence vector of an independent set.

Given any $x \in P$, consider the *tight* sets S, i.e. those sets for which $x(S) = r(S)$. The next lemma shows that these tight sets are closed under taking intersections or unions. This lemma is really central, and follows from submodularity.

Lemma 4.5 *Let $x \in P$. Let*

$$\mathcal{F} = \{S \subseteq E : x(S) = r(S)\}.$$

Then

$$S \in \mathcal{F}, T \in \mathcal{F} \Rightarrow S \cap T \in \mathcal{F}, S \cup T \in \mathcal{F}.$$

Observe that the lemma applies even if S and T are disjoint. In that case, it says that $\emptyset \in \mathcal{F}$ (which is always the case as $x(\emptyset) = 0 = r(\emptyset)$) and $S \cup T \in \mathcal{F}$.

Proof: The fact that $S, T \in \mathcal{F}$ means that:

$$r(S) + r(T) = x(S) + x(T). \tag{2}$$

Since $x(S) = \sum_{e \in S} x_e$, we have that

$$x(S) + x(T) = x(S \cap T) + x(S \cup T), \tag{3}$$

i.e. that the function $x(\cdot)$ is modular (both x and $-x$ are submodular). Since $x \in P$, we know that $x(S \cap T) \leq r(S \cap T)$ (this is true even if $S \cap T = \emptyset$) and similarly $x(S \cup T) \leq r(S \cup T)$; this implies that

$$x(S \cap T) + x(S \cup T) \leq r(S) + r(T). \tag{4}$$

By submodularity, we have that

$$r(S \cap T) + r(S \cup T) \leq r(S) + r(T). \tag{5}$$

Combining (2)–(5), we get

$$r(S) + r(T) = x(S) + x(T) = x(S \cap T) + x(S \cup T) \leq r(S \cap T) + r(S \cup T) \leq r(S) + r(T),$$

and therefore we have equality throughout. This implies that $x(S \cap T) = r(S \cap T)$ and $x(S \cup T) = r(S \cup T)$, i.e. $S \cap T$ and $S \cup T$ in $\mathcal{F}$. △

To prove that any vertex or extreme point of P is integral, we first characterize any face of P. A *chain* $\mathcal{C}$ is a family of sets such that for all $S, T \in \mathcal{C}$ we have that either $S \subseteq T$ or $T \subseteq S$ (or both if $S = T$).

Theorem 4.6 *Consider any face F of P. Then there exists a chain $\mathcal{C}$ and a subset $J \subseteq E$ such that:*

$$F = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \le r(S) & \forall S \subseteq E \\ x(C) = r(C) & \forall C \in \mathcal{C} \\ x_e \ge 0 & \forall e \in E \setminus J \\ x_e = 0 & \forall e \in J.\} \end{array}$$

Proof: By Theorem 3.5 of the polyhedral notes, we know that any face is characterized by setting some of the inequalities of P by equalities. In particular, F can be expressed as

$$F = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \le r(S) & \forall S \subseteq E \\ x(C) = r(C) & \forall C \in \mathcal{F} \\ x_e \ge 0 & \forall e \in E \setminus J \\ x_e = 0 & \forall e \in J.\} \end{array}$$

where $J = \{e : x_e = 0 \text{ for all } x \in F\}$ and $\mathcal{F} = \{S : x(S) = r(S) \text{ for all } x \in F\}$. To prove the theorem, we need to argue that the system of equations:

$$x(C) = r(C) \quad \forall C \in \mathcal{F}$$

can be replaced by an equivalent (sub)system in which $\mathcal{F}$ is replaced by a chain $\mathcal{C}$. To be equivalent, we need that

$$span(\mathcal{F}) = span(\mathcal{C})$$

where by $span(\mathcal{L})$ we mean

$$span(\mathcal{L}) := span\{\chi(C) : C \in \mathcal{L}\}.$$

Let $\mathcal{C}$ be a maximal subchain of $\mathcal{F}$, i.e. $\mathcal{C} \subseteq \mathcal{F}$, $\mathcal{C}$ is a chain and for all $S \in \mathcal{F} \setminus \mathcal{C}$, there exists $C \in \mathcal{C}$ such that $S \not\subseteq C$ and $C \not\subseteq S$. We claim that $span(\mathcal{C}) = span(\mathcal{F})$.

Suppose not, i.e. $H \neq span(\mathcal{F})$ where $H := span(\mathcal{C})$. This means that there exists $S \in \mathcal{F} \setminus \mathcal{C}$ such that $\chi(S) \notin H$ but S cannot be added to $\mathcal{C}$ without destroying the chain structure. In other words, for any such S, the set of 'chain violations'

$$V(S) := \{C \in \mathcal{C} : C \not\subseteq S \text{ and } S \not\subseteq C\}$$

is non-empty. Among all such sets S, choose one for which $|V(S)|$ is as small as possible ($|V(S)|$ cannot be 0 since we are assuming that $V(S) \neq \emptyset$ for all possible S). Now fix some set $C \in V(S)$. By Lemma 4.5, we know that both $C \cap S \in \mathcal{F}$ and $C \cup S \in \mathcal{F}$. Observe that there is a linear dependence between $\chi(C), \chi(S), \chi(C \cup T), \chi(C \cap T)$:

$$\chi(C) + \chi(S) = \chi(C \cup S) + \chi(C \cap S).$$

This means that, since $\chi(C) \in H$ and $\chi(S) \notin H$, we must have that either $\chi(C \cup S) \notin H$ or $\chi(C \cap S) \notin H$ (otherwise $\chi(S)$ would be in H). Say that $\chi(B) \notin H$ where B si either $C \cup S$ or $C \cap S$. This is a contradiction since $|V(B)| < |V(S)|$, contradicting our choice of S. Indeed, one can see that $V(B) \subset V(S)$ and $C \in V(S) \setminus V(B)$. $\triangle$

As a corollary, we can also obtain a similar property for an extreme point, starting from Theorem 3.6.

Corollary 4.7 *Let x be any extreme point of P. Then there exists a chain $\mathcal{C}$ and a subset $J \subseteq E$ such that x is the unique solution to:*

$$\begin{array}{ll} x(C) = r(C) & \forall C \in \mathcal{C} \\ x_e = 0 & \forall e \in J. \end{array}$$

From this corollary, the integrality of every extreme point follows easily. Indeed, if the chain given in the corollary consists of $C_1 \subset C_2 \subset C_p$ the the system reduces to

$$\begin{array}{ll} x(C_i \setminus C_{i-1}) = r(C_i) - r(C_{i-1}) & i = 1, \cdots, p \\ x_e = 0 & \forall e \in J, \end{array}$$

where $C_0 = \emptyset$. For this to have a unique solution, we'd better have $|C_i \setminus C_{i-1} \setminus J| \leq 1$ for all i and the values for the resulting x_e's will be integral.

4.4.3 Facet Proof

Our last proof of Theorem 4.4 focuses on the facets of $conv(X)$.

First we need to argue that we are missing any equalities. Let's focus on the (interesting) case in which any singleton set is independent: $\{e\} \in \mathcal{I}$ for every $e \in E$. In that case $dim(conv(X)) = |E|$ since we can exhibit $|E| + 1$ affinely independent points in X: the 0 vector and all unit vectors $\chi(\{e\})$ for $e \in E$. Thus we do not need any equalities. See exercise 4-11 if we are not assuming that every singleton set is independent.

Now consider any facet F of $conv(X)$. This facet is induced by a valid inequality $\alpha^T x \leq \beta$ where $\beta = \max\{\sum_{e \in I} \alpha_e : I \in \mathcal{I}\}$. Let

$$\mathcal{O} = \{I \in \mathcal{I} : \sum_{e \in I} \alpha_e = \beta\},$$

i.e. $\mathcal{O}$ is the set of all independent sets whose incidence vectors belong to the face. We'll show that there exists an inequality in opur description of P which is satisfied at equality by the incidence vectors of all sets $I \in \mathcal{O}$.

We consider two cases. If there exists $e \in M$ such that $\alpha_e < 0$ then $I \in \mathcal{O}$ implies that $e \notin I$, implying that our face F is included in the face induced by $x_e \geq 0$ (which is in our description of P).

For the other case, we assume that for all $e \in E$, we have $\alpha_e \geq 0$. We can further assume that $\alpha_{\max} := \max_{e \in E} \alpha_e > 0$ since otherwise F is trivial. Now, define S as

$$S = \{e \in E : \alpha_e = \alpha_{\max}\}.$$

Claim 4.8 *For any $I \in \mathcal{O}$, we have $|I \cap S| = r(S)$.*

This means that the face F is contained in the face induced by the inequality $x(S) \leq r(S)$ and therefore we have in our description of P one inequality inducing each facet of $conv(X)$. Thus we have a complete description of $conv(X)$.

To prove the claim, suppose that $|I \cap S| < r(S)$. Thus $I \cap S$ can be extended to an independent set $X \in \mathcal{I}$ where $X \subseteq S$ and $|X| > |I \cap S|$. Let $e \in X \setminus (I \cap S)$; observe that $e \in S$ by our choice of X. Since $\alpha_e > 0$ we have that $I + e \notin \mathcal{I}$, thus there is a circuit $C \subseteq I + e$. By the unique circuit property (see Theorem 4.1), for any $f \in C$ we have $I + e - f \in \mathcal{I}$. But $C \setminus S \neq \emptyset$ since $(I \cap S) + e \in \mathcal{I}$, and thus we can choose $f \in C \setminus S$. The cost of $I + e - f$ satisfies:

$$c(I + e - f) = c(I) + c(e) - c(f) > c(I),$$

contradicting the definition of $\mathcal{O}$.

4.5 Facets?

Now that we have a description of the matroid polytope in terms of linear inequalities, one may wonder which of these (exponentially many) inequalities define facets of $conv(X)$.

For simplicity, let's assume that $r(\{e\}) = 1$ for all $e \in E$ (e belongs to some independent set). Then, every nonnegativity constraint defines a facet of $P = conv(X)$. Indeed, the 0 vector and all unit vectors except $\chi(\{e\})$ constitute $|E|$ affinely independent points satisfying $x_e = 0$. This mean that the corresponding face has dimension at least $|E| - 1$ and since the dimension of P itself is $|E|$, the face is a facet.

We now consider the constraint $x(S) \leq r(S)$ for some set $S \subseteq E$. If S is not closed (see Definition 4.2) then $x(S) \leq r(S)$ definitely does not define a facet of $P = conv(X)$ since it is implied by the constraints $x(span(S)) \leq r(S)$ and $x_e \geq 0$ for $e \in span(S) \setminus S$.

Another situation in which $x(S) \leq r(S)$ does not define a facet is if S can be expressed as the disjoint union of $U \neq \emptyset$ and $S \setminus U \neq \emptyset$ and $r(U) + r(S \setminus U) = r(S)$. In this case, the inequality for S is implied by those for U and for $S \setminus U$.

Definition 4.3 *S is said to be inseparable if there is no U with $\emptyset \neq U \subset S$ such that $r(S) = r(U) + r(S \setminus U)$.*

From what we have just argued, a necessary condition for $x(S) \leq r(S)$ to define a facet of $P = conv(X)$ is that S is *closed and inseparable*. This can be shown to be sufficient as well, although the proof is omitted.

As an example, consider a partition matroid with $M = (E, \mathcal{I})$ where

$$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ for all } i = 1, \cdots, l\},$$

for disjoint E_i's. Assume that $k_i \geq 1$ for all i. The rank function for this matroid is:

$$r(S) = \sum_{i=1}^{l} \min(k_i, |S \cap E_i|).$$

For a set S to be inseparable, there must exist (i) $i \in \{1, \cdots, l$ with $S \subseteq E_i$, and (ii) $|S \cap E_i|$ is either ≤ 1 or $> k_i$ for every i. Furthermore, for $S \subseteq E_i$ to be closed, we must have that if

$|S \cap E_i| > k_i$ then $S \cap E_i = E_i$. Thus the only sets we need for the description of a partition matroid polytope are (i) sets $S = E_i$ for i with $|E_i| > k_i$ and (ii) singleton sets $\{e\}$ for $e \in E$. The partition matroid polytope is thus given by:

$$\begin{array}{llll} P = \{x \in \mathbb{R}^{|E|}: & x(E_i) \leq k_i & i \in \{1, \cdots, l\} : |E_i| > k_i \\ & 0 \leq x_e \leq 1 & e \in E\}. \end{array}$$

As another example, take M to be the graphic matroid $M(G)$. For a set of edges $F \subseteq E$ to be inseparable, we need that the subgraph (V, F) has only one non-trivial (i.e. with more than 1 vertex) connected component; indeed, if we partition F into the edge sets $F_1, \cdots, F_c$ of the (c non-trivial) connected components, we have that $r(F) = \sum_{i=1}^{c} r(F_i)$ and thus c must be 1 for F to be inseparable. Given a set F of edges, its span (with respect to the graphic matroid) consists of all the edges with both endpoints within the same connected component of F; these are the edges whose addition does not increase the size of the largest forest. Thus, for F to be inseparable and closed, we must have that there exists a vertex set $S \subseteq V$ such that $F = E(S)$ ($E(S)$ denotes all the edges with both endpoints in S) and $(S, E(S))$ is connected. Thus the forest polytope (convex hull of all forests in a graph $G = (V, E)$) is given by:

$$\begin{array}{llll} P = \{x \in \mathbb{R}^{|E|}: & x(E(S)) \leq |S| - 1 & S \subseteq V : \text{E(S) connected} \\ \\ & 0 \leq x_e & e \in E\}. \end{array}$$

(As usual, $x(E(S))$ denotes $\sum_{e \in E(S)} x_e$.) Observe that this polyhedral description still has a very large number of inequalities.

From this, we can also easily derive the *spanning tree polytope* of a graph, namely the convex hull of incidence vectors of all spanning trees in a graph. Indeed, this is a face of the forest polytope obtained by replacing the inequality for $S = V$ ($x(E) \leq |V| - 1$) by an equality:

$$\begin{array}{llll} P = \{x \in \mathbb{R}^{|E|}: & X(E) = |V| - 1 & \\ & x(E(S)) \leq |S| - 1 & S \subset V : \text{E(S) connected} \\ \\ & 0 \leq x_e & e \in E\}. \end{array}$$

Exercise 4-11. Let $M = (E, \mathcal{I})$ be a matroid and let $S = \{e \in E : \{e\} \in \mathcal{I}\}$. Show that $dim(conv(X)) = |S|$ (where X is the set of incidence vectors of indpendent sets) and show that the description for P has the required number of linearly independent equalities.

Exercise 4-12. Let $M = (E, \mathcal{I})$ be a matroid and let P be the corresponding matroid polytope, i.e. the convex hull of characteristic vectors of independent sets. Show that two independent sets I_1 and I_2 are adjacent on P if and only if either (i) $I_1 \subseteq I_2$ and $|I_1|+1 = |I_2|$, or (ii) $I_2 \subseteq I_1$ and $|I_2| + 1 = |I_1|$, or (iii) $|I_1 \setminus I_2| = |I_2 \setminus I_1| = 1$ and $I_1 \cup I_2 \notin \mathcal{I}$.

1 Matroid Polytope

In the previous lecture we saw that for a matroid $M = (S, \mathcal{I})$ the following system of inequalities determines the convex hull of the independent sets of M (i.e., sets in $\mathcal{I}$):

$$\begin{array}{rcll} x(U) & \leq & r_M(U) & U \subseteq S \\ x(e) & \geq & 0 & e \in S \end{array}$$

where $r_M(\cdot)$ is the rank function of M. The proof was based on a dual fitting technique via the Greedy algorithm for a maximum weight independent set problem.

In this lecture, we will give a different primal proof that is built on uncrossing. This is based on [2].

Theorem 1 *Let x be an extreme point of the polytope*

$$(*) \begin{cases} x(U) \leq r_M(U) & U \subseteq S \\ x(e) \geq 0 & e \in S \end{cases}$$

Then there is some $e \in S$ such that $x(e) \in \{0, 1\}$.

The following corollary follows by induction from Theorem 1. We leave a formal proof as an exercise.

Corollary 2 *The system of inequalities $(*)$ determine the independent set polytope of $M = (S, \mathcal{I})$.*

Now we turn our attention to the proof of Theorem 1.
Proof of Theorem 1. Let x be an extreme solution for the polytope $(*)$. Suppose that M has a loop e. Since $r_M(\{e\}) = 0$, it follows that $x(e) = 0$ and we are done. Therefore we may assume that M does not have any loops and thus the polytope $(*)$ is full dimensional[1]. Now suppose that $x(e) \in (0, 1)$ for all elements $e \in S$. Let n denote the number of elements in S. Let

$$\mathcal{F} = \{U \mid U \subseteq S, x(U) = r_M(U)\}$$

Differently said, $\mathcal{F}$ is the set of all sets whose constraints are tight at x (i.e., sets whose constraints are satisfied with equality by the solution x).

Before proceeding with the proof, we note that the submodularity for the rank function $r_M(\cdot)$ implies that $\mathcal{F}$ has the following "uncrossing" property.

[1] A polytope is full dimensional if it has an interior point, i.e., a point x that does not satisfy any of the constraints with equality. Consider x such that, for all e, $x(e) = \epsilon$ for some $0 < \epsilon < 1/|S|$. Clearly, $x(e) > 0$ for any e. For any set U, we have $x(U) = \epsilon|U| < 1$. If M does not have any loops, $r_M(U) \geq 1$ for all sets U. Thus M is full-dimensional if there are no loops.

Lemma 3 *If $A, B \in \mathcal{F}$ then $A \cap B$ and $A \cup B$ are in $\mathcal{F}$.*

Proof: Let A and B be two sets in $\mathcal{F}$; thus $x(A) = r_M(A)$ and $x(B) = r_M(B)$. It follows from the submodularity of the rank function that

$$x(A) + x(B) = r_M(A) + r_M(B) \geq r_M(A \cap B) + r_M(A \cup B)$$

Additionally,

$$x(A) + x(B) = x(A \cap B) + x(A \cup B)$$

Therefore $x(A \cap B) + x(A \cup B) \geq r_M(A \cap B) + r_M(A \cup B)$. Since $x(A \cap B) \leq r_M(A \cap B)$ and $x(A \cup B) \leq r_M(A \cup B)$, it follows that $x(A \cap B) = r_M(A \cap B)$ and $x(A \cup B) = r_M(A \cup B)$. Thus $A \cap B$ and $A \cup B$ are also in $\mathcal{F}$. $\square$

Let $\chi(U)$ denote the characteristic vector of U. Since x is a vertex solution, $(*)$ is full dimensional, and $x(e) \neq 0$ for all e, there is a collection $\{U_1, U_2, \ldots, U_n\}$ of n sets such that x satisfies the constraint corresponding to each U_i with equality (i.e., $x(U_i) = r_M(U_i)$ for $1 \leq i \leq n$) and the vectors $\chi(U_1), \ldots, \chi(U_n)$ are linearly independent. Therefore the set $\{\chi(U) \mid U \in \mathcal{F}\}$ has n linearly independent vectors.

For a set $\mathcal{A} \subseteq 2^S$, let $span(\mathcal{A})$ denote $span(\{\chi(U) \mid U \in \mathcal{A}\})$, where $\chi(U)$ is the characteristic vector of U.

Lemma 4 *There exists a laminar family $\mathcal{C} \subseteq \mathcal{F}$ such that $span(\mathcal{C}) = span(\mathcal{F})$. Moreover, $\mathcal{C}$ is a* ***chain****, i.e., for any two sets $A, B \in \mathcal{C}$, either $A \subseteq B$ or $B \subseteq A$.*

Assuming Lemma 4, we can complete the proof of Theorem 1 as follows. Let $\mathcal{C}$ be the chain guaranteed by Lemma 4. Since $span(\mathcal{C}) = span(\mathcal{F})$, there exists a chain $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| = n$ and x is the unique solution to the system

$$x(U) = r_M(U) \qquad U \in \mathcal{C}'$$

Let $\mathcal{C}' = \{A_1, A_2, \ldots, A_n\}$; wlog, $A_1 \subset A_2 \subset \cdots \subset A_n$. Let $A_0 = \emptyset$. Suppose that there exists an i such that $|A_i \setminus A_{i-1}| = 1$, and let $e \in A_i \setminus A_{i-1}$. Now we claim that we must have $x(e) \in \{0, 1\}$. To see why this is true, note that we have

$$x(e) = x(A_i) - x(A_{i-1}) = r_M(A_i) - r_M(A_{i-1})$$

Since $r_M(A_i) - r_M(A_{i-1})$ is an integer and $r_M(A_{i-1}) \leq r_M(A_i) \leq r_M(A_{i-1}) + 1$, it follows that $r_M(A_i) - r_M(A_{i-1}) \in \{0, 1\}$. But this contradicts the fact that $x(e) \in (0, 1)$. Therefore we may assume that $|A_i \setminus A_{i-1}| \geq 2$. But then $|S| \geq 2n$, which is a contradiction.

Finally, we turn our attention to the proof of Lemma 4.

Proof of Lemma 4. Let $\mathcal{C}$ be a chain in $\mathcal{F}$ that is maximal with respect to inclusion (i.e., $\mathcal{C}$ is not a proper subset of any chain in $\mathcal{F}$). We claim that $span(\mathcal{C}) = span(\mathcal{F})$. Suppose not and let $A \in \mathcal{F}$ be such that $\chi(A) \in span(\mathcal{F}) \setminus span(\mathcal{C})$. If there are several such sets A, we choose one that minimizes the number of sets in $\mathcal{C}$ that it *properly* intersects[2].

[2]Two sets X and Y properly intersect if $X \cap Y, X - Y, Y - X$ are all non-empty.

Now suppose that A does not *properly* intersect any set in $\mathcal{C}$. Clearly, $\mathcal{C}+A$ is not a chain, since this contradicts the maximality of $\mathcal{C}$. Therefore there exist $B, B' \in \mathcal{C}$ such that B is the minimal set in $\mathcal{C}$ that contains A and B' is the maximal set in $\mathcal{C}$ that is contained in B. By Lemma 3, $A \cup B'$ is in $\mathcal{F}$. If $A \cup B'$ is a proper subset of B, $\mathcal{C}+(A \cup B')$ is a chain, which contradicts the maximality of $\mathcal{C}$. Therefore we must have $A \cup B' = B$. Since A and B' are disjoint, we have $\chi(A)+\chi(B') = \chi(B)$ and thus $\chi(A)$ is in the span of $\chi(B)$ and $\chi(B')$, which contradicts the fact that $\chi(A) \notin span(\mathcal{C})$.

Therefore we may assume that A properly intersects a set B in $\mathcal{C}$. By Lemma 3, $A \cup B$ and $A \cap B$ are in $\mathcal{F}$.

Proposition 5 *Each of $A \cup B$, $A \cap B$ properly intersects fewer sets in $\mathcal{C}$ than A.*

Assuming Proposition 5, we can complete the proof as follows. It follows from our choice of A that $A \cup B$ and $A \cap B$ are both in $span(\mathcal{C})$. Since $\chi(A)+\chi(B) = \chi(A \cup B)+\chi(A \cap B)$, it follows that $\chi(A)$ is in $span(\mathcal{C})$ as well, which is a contradiction. Therefore it suffices to prove Proposition 5.

Proof of Proposition 5. Since each of $A \cup B$, $A \cap B$ does not properly intersect B, it suffices to show that if a set $B' \in \mathcal{C}$ properly intersects $A \cup B$ (or $A \cap B$) then it properly intersects A as well.

Let $B' \in \mathcal{C}$ be a set that properly intersects $A \cup B$. Since B and B' are both in C, it follows that one of B, B' is a subset of the other. If B' is a subset of B, B' is contained in $A \cup B$ (and thus does not properly intersect $A \cup B$). Therefore B must be a proper subset of B'. Clearly, B' intersects A (since $A \cap B$ is nonempty). If B' does not properly intersect A, it follows that one of A, B' is a subset of the other. If $A \subseteq B'$, it follows that $A \cup B \subseteq B'$, which is a contradiction. Therefore we must have $B \subset B' \subseteq A$, which is a contradiction as well. Thus B' properly intersects A.

Let $B' \in \mathcal{C}$ be a set that properly intersects $A \cap B$. Clearly, B' intersects A and thus it suffices to show that $B' \setminus A$ is nonempty. As before, one of B, B' is a subset of the other. Clearly, B' must be a subset of B (since otherwise $A \cap B \subseteq B \subseteq B'$). Now suppose that $B' \subseteq A$. Since B' is a subset of B, it follows that $B' \subseteq A \cap B$, which is a contradiction. Therefore $B' \setminus A$ is non-empty, as desired. $\square$

2 Facets and Edges of Matroid Polytopes

Recall that the following system of inequalities determines the matroid polytope.

$$(*) \begin{cases} x(U) \leq r_M(U) & U \subseteq S \\ x(e) \geq 0 & e \in S \end{cases}$$

Throughout this section, we assume that the matroid has no loops and thus the polytope is full dimensional.

It is useful to know which inequalities in the above system are redundant. As we will see shortly, for certain matroids, the removal of redundant inequalities gives us a system with only polynomially many constraints.

Recall that a *flat* is a subset $U \subseteq S$ such that $U = span(U)$. Consider a set U that is *not* a flat. Since $r_M(U) = r_M(span(U))$ and $U \subset span(U)$, any solution x that satisfies the constraint

$$x(span(U)) \leq r_M(span(U))$$

also satisfies the inequality

$$x(U) \leq r_M(U)$$

Therefore we can replace the system $(*)$ by

$$(**)\begin{cases} x(F) \leq r_M(F) & F \subseteq S,\ F \text{ is a flat} \\ x(e) \geq 0 & e \in S \end{cases}$$

Definition 1 *A flat F is separable if there exist flats F_1, F_2 such that F_1 and F_2 partition F and*

$$r_M(F_1) + r_M(F_2) = r_M(F)$$

If F is a separable flat, any solution x that satisfies the constraints

$$x(F_1) \leq r_M(F_1)$$

$$x(F_2) \leq r_M(F_2)$$

also satisfies the constraint

$$x(F) \leq r_M(F)$$

since $x(F) = x(F_1)+x(F_2)$ and $r_M(F) = r_M(F_1)+r_M(F_2)$. Therefore we can remove the constraint $x(F) \leq r_M(F)$ from $(**)$. Perhaps surprisingly, the resulting system does not have any redundant constraints. The interested reader can consult Chapter 40 in Schrijver [**?**] for a proof.

Theorem 6 *The system of inequalities*

$$\begin{array}{rcll} x(F) & \leq & r_M(F) & F \subseteq S,\ F \text{ is an inseparable flat} \\ x(e) & \geq & 0 & e \in S \end{array}$$

is a minimal system for the independent set polytope of a loopless matroid M.

As an example, consider the uniform matroid. The independent set polytope for the uniform matroid is determined by the following constraints:

$$\sum_{e \in S} x(e) \leq k$$

$$x(e) \geq 0 \qquad e \in S$$

Similarly, the independent set polytope for the partition matroid induced by the partition $S_1, \ldots, S_h$ of S and integers $k_1, \ldots, k_h$ is determined by the following constraints:

$$\sum_{e \in S_i} x(e) \leq k_i \qquad 1 \leq i \leq k$$

$$x(e) \geq 0 \qquad e \in S$$

Finally, consider the graphic matroid induced by a graph $G = (V, E)$. The base polytope of a graphic matroid corresponds to the the spanning tree polytope, which is determined by the following constraints:

$$\begin{array}{rcll} x(E[U]) & \leq & |U| - 1 & U \subseteq V \\ x(E) & = & |V| - 1 & \\ x(e) & \geq & 0 & e \in E \end{array}$$

where $E[U]$ is the set of edges inside the vertex set $U \subseteq V$.

Definition 2 *Two vertices x, x' of a polyhedron P are* adjacent *if they are contained in a face F of P of dimension one, i.e., a line.*

Theorem 7 *Let $M = (S, \mathcal{I})$ be a loopless matroid. Let $I, J \in \mathcal{I}$, $I \neq J$. Then $\chi(I)$ and $\chi(J)$ are adjacent vertices of the independent set polytope of M if and only if $|I \triangle J| = 1$ or $|I \setminus J| = |J \setminus I| = 1$ and $r_M(I) = r_M(J) = |I| = |J|$.*

The interested reader can consult Schrijver [**?**] for a proof.

3 Further Base Exchange Properties

We saw earlier the following base exchange lemma.

Lemma 8 *Let B and B' be two bases of a matroid M, and let y be an element of $B' \setminus B$. Then*

(i) there exists $x \in B \setminus B'$ such that $B' - y + x$ is a base

(ii) there exists $x \in B \setminus B'$ such that $B + y - x$ is a base

We will prove a stronger base exchange theorem below and derive some corollaries that will be useful in matroid intersection and union.

Theorem 9 (Strong Base Exchange Theorem) *Let B, B' be two bases of a matroid M. Then for any $x \in B \setminus B'$ there exists an $y \in B' \setminus B$ such that $B - x + y$ and $B' - y + x$ are both bases.*

Proof: Let x be any element in $B \setminus B'$. Since B' is a base, $B' + x$ has a unique circuit C. Then $(B \cup C) - x$ contains a base. Let B'' be a base of $(B \cup C) - x$ that contains $B - x$. We have $B'' = B - x + y$, for some $y \in C - x$.

Now suppose that $B' - y + x$ is not a base. Then $B' - y + x$ has a circuit C'. Since $y \in C \setminus C'$, $B' + x$ has two distinct circuits C, C', which is a contradiction (see Corollary 22 in Lecture 14). Therefore $B' - y + x$ is independent and, since $|B' - y + x| = |B'|$, $B' - y + x$ is a base. $\square$

In fact, Theorem 9 holds when B, B' are independent sets of the same size instead of bases.

Corollary 10 *Let I, J be two independent sets of a matroid $M = (S, \mathcal{I})$ such that $|I| = |J|$. Then for any $x \in I \setminus J$ there exists an $y \in J \setminus I$ such that $I - x + y$ and $J - y + x$ are both independent sets.*

Proof: Let $k = |I| = |J|$. Let $M' = (S, \mathcal{I}')$, where

$$\mathcal{I}' = \{I \mid I \in \mathcal{I} \text{ and } |I| \leq k\}$$

It is straightforward to verify that M' is a matroid as well. Additionally, since every independent set in M' has size at most k, I and J are bases in M'. It follows from Theorem 9 that for any $x \in I \setminus J$ there exists an $y \in J \setminus I$ such that $I - x + y$ and $J - y + x$ are both bases in M', and thus independent sets in M. □

Let $M = (S, \mathcal{I})$ be a matroid, and let $I \in \mathcal{I}$. We define a directed bipartite graph $D_M(I)$ as follows. The graph $D_M(I)$ has vertex set S; more precisely, its bipartition is $(I, S \setminus I)$. There is an edge from $y \in I$ to $z \in S \setminus I$ iff $I - y + z$ is an independent set.

Lemma 11 *Let $M = (S, \mathcal{I})$ be a matroid, and let I, J be two independent sets in M such that $|I| = |J|$. Then $D_M(I)$ has a perfect matching on $I \triangle J$*[3].

Proof: We will prove the lemma using induction on $|I \triangle J|$. If $|I \triangle J| = 0$, the lemma is trivially true. Therefore we may assume that $|I \triangle J| \geq 1$. It follows from Corollary 10 that there exists an $y \in I$ and $z \in J$ such that $I' = I - y + z$ and $J' = J + y - z$ are independent sets. Note that $|I' \triangle J'| < |I \triangle J|$ and $|I'| = |J'|$. It follows by induction that $D_M(I)$ has a perfect matching N on $I' \triangle J'$. Then $N \cup \{(y, z)\}$ is a perfect matching on $I \triangle J$. □

Lemma 12 *Let $M = (S, \mathcal{I})$ be a matroid. Let I be an independent set in M, and let J be a subset of S such that $|I| = |J|$. If $D_M(I)$ has a* unique *perfect matching on $I \triangle J$ then J is an independent set.*

Before proving the lemma, we note the following useful property of unique perfect matchings.

Proposition 13 *Let $G = (X, Y, E)$ be a bipartite graph such that G has a* unique *perfect matching N. Then we can label the vertices of X as $x_1, \ldots, x_t$, and we can label the vertices of Y as $y_1, \ldots, y_t$ such that*

$$N = \{(x_1, y_1), \ldots, (x_t, y_t)\}$$

and $(x_i, y_j) \notin E$ for all i, j such that $i < j$.

Proof: We start by noting that there is an edge $xy \in N$ such that one of x, y has degree one. We construct a trail[4] by alternately taking an edge in N and an edge not in N, until either we cannot extend the trail or we reach a previously visited vertex. Now suppose that the trail has a cycle C. Since G is bipartite, C has even length. Thus we can construct a perfect matching from N by removing the edges of C that are in N and adding the edges of C that are not in N, which contradicts the fact that G has a unique perfect matching. Therefore we may assume that the trail is a path. If the last edge of the trail is not in N, we can extend the trail by taking the edge of N incident to the last vertex. Therefore the last edge must be in N. Then the last vertex on the trail has degree one, since otherwise we could extend the trail using one of the edges incident to it that are not in N. It follows that the last edge of the trail is the desired edge.

Now let xy be an edge in N such that one of its endpoints has degree one in G. Suppose that x has degree one. We let $x_1 = x$, $y_1 = y$, and we remove x and y to get a graph G'. Since $N - xy$

[3] A perfect matching on a set U is a matching such that S is the set of endpoints of the edges in the matching.

[4] A trail is a walk in which all edges are distinct.

is the unique perfect matching in G', it follows by induction that we can label the vertices of G' such that

$$N - xy = \{(x_2, y_2), \ldots, (x_t, y_t)\}$$

such that (x_i, y_j) is not an edge in G', for all $2 \leq i < j \leq t$. Since x_1 has degree one in G, we are done. Therefore we may assume that y has degree one. We let $x_t = x$, $y_t = y$, and we remove x and y to get a graph G'. As before, it follows by induction that we can label the vertices of G' such that

$$N - xy = \{(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})\}$$

such that (x_i, y_j) is not an edge in G', for all $1 \leq i < j \leq t-1$. Since y_t has degree one in G, we are done. □

Proof of Lemma 12. Let G denote the (undirected) subgraph of $D_M(I)$ induced by $I \triangle J$, and let N denote the unique perfect matching in G. Since G is a bipartite graph, it follows from Proposition 13 that we can label the vertices of $I \setminus J$ as $y_1, \ldots, y_t$, and we can label the vertices of $J \setminus I$ as $z_1, \ldots, z_t$ such that

$$N = \{(y_1, z_1), \ldots, (y_t, z_t)\}$$

and $(y_i, z_j) \notin E(G)$, for all $1 \leq i < j \leq t$.

Now suppose that J is not independent, and let C be a circuit in J. Let i be the smallest index such that $z_i \in C$. Consider any element z_j in $C - z_i$. Since $j > i$, it follows that $(y_i, z_j) \notin D_M(I)$. Therefore any element z in $C - z_i$ is in $span_M(I - y_i)$, since for any $z \in C - z_i$, either z is in $I \cap J$ or $z = z_j$ for some j. Hence $C - z_i$ is a subset of $span(I - y_i)$. Since C is a circuit,

$$C \subseteq span(C - z_i) \subseteq span(I - y_i)$$

Thus $z_i \in span(I - y_i)$, which contradicts the fact that $I - y_i + z_i$ is independent. □

Corollary 14 *Let $M = (S, \mathcal{I})$ be a matroid, and let $I \in \mathcal{I}$. Let J be a subset of S with the following properties:*

(i) $|I| = |J|$

(ii) $r_M(I \cup J) = |I|$

(iii) $D_M(I)$ *has a unique perfect matching on* $I \triangle J$

Let e be any element not in $I \cup J$ such that $I + e \in \mathcal{I}$. Then $J + e \in \mathcal{I}$.

Proof: It follows from Lemma 12 that J is independent. Since $r_M(I \cup J) = |I|$, both I and J are maximal independent sets in $I \cup J$. Thus $I \subseteq span(J)$ and $J \subseteq span(I)$. Since $I + e$ is independent, $e \notin span(I)$. As we have seen in Lecture 14, since $J \subseteq span(I)$, it follows that $span(J) \subseteq span(I)$. Therefore $e \notin span(J)$ and thus $J + e$ is independent. □

References

[1] Alexander Schrijver. Combinatorial Optimization: Polyhedra and Efficiency, Chapters 39-40, Vol. B, Springer-Verlag 2003.

[2] L. C. Lau, R. Ravi and M. Singh. Iterative Methods in Combinatorial Optimization. Draft of upcoming book, March 2009.

1 Matroid Intersection

One of several major contributions of Edmonds to combinatorial optimization is algorithms and polyhedral theorems for matroid intersection, and more generally polymatroid intersection.

From an optimization point of view, the matroid intersection problem is the following: Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be two matroids on the same ground set S. Then $\mathcal{I}_1 \cap \mathcal{I}_2$ is the collection of all sets that are independent in both matroids.

One can ask the following algorithmic questions:

1. Is there a common base in the two matroids? That is, is there $\mathcal{I} \in \mathcal{B}_1 \cap \mathcal{B}_2$ where $\mathcal{B}_1$ and $\mathcal{B}_2$ are the bases of M_1 and M_2.
2. Output a maximum cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$.
3. Given $w : S \to \Re$, output a maximum weight set in $\mathcal{I}_1 \cap \mathcal{I}_2$. Or output a maximum weight common base, if it exists.

Remark 1 *It is easy to see that the intersection of two matroids, i.e.,* $(S, \mathcal{I}_1 \cap \mathcal{I}_2)$*, is not necessarily a matroid.*

Exercise 2 *If* $M_1 = (S, \mathcal{I}_1)$ *is a matroid and* $M_2 = (S, \mathcal{I}_2)$ *is the uniform matroid, then* $M_3 = (S, \mathcal{I}_1 \cap \mathcal{I}_2)$ *is a matroid.*

As one can imagine, matroid intersection can capture several additional optimization problems.

Example: Bipartite Matching. Let $G = (V, E)$ be a bipartite graph with bipartition $A \cup B$. Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two partition matroids on E, where

$$\mathcal{I}_1 = \{E' \subseteq E \mid |\delta(v) \cap E'| \leq 1,\ v \in A\}$$
$$\mathcal{I}_2 = \{E' \subseteq E \mid |\delta(v) \cap E'| \leq 1,\ v \in B\}.$$

Then it is easy to see that $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ if and only if I induces a matching in G. Thus bipartite matching problems are special cases of matroid intersection problems.

Example: Branchings and Arborescences. Let $D = (V, A)$ be a directed graph. A *branching* in D is a set of edges $A' \subseteq A$ such that the in-degree of each node is at most one and the edges in A form a forest. (An example is shown in Figure 1.) An *arborescence* rooted at a node $r \in V$ is a directed out-tree such that r has a path to each node $v \in V$. Thus an arborescence is a branching in which r is the only node with in-degree 0.

Consider two matroids $M_1 = (A, \mathcal{I}_1)$ and $M_2 = (A, \mathcal{I}_2)$ where $M_1 = (A, \mathcal{I}_1)$ is a partition matroid:

$$\mathcal{I}_1 = \{A' \subseteq A \mid |\delta^-(v) \cap A'| \leq 1,\ v \in V\}$$

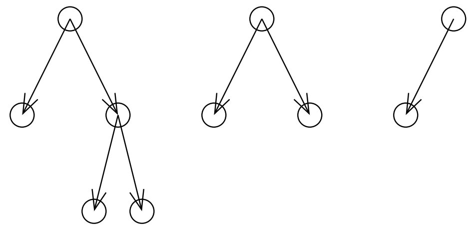


Figure 1: Example of a branching

and M_2 is a graphic matroid on $G = (V, A^u)$ obtained by making an undirected graph on V by removing directions from arcs in A with:

$$\mathcal{I}_2 = \{A' \subseteq A \mid A' \text{ induces a forest in } G^u\}$$

It is easy to see that $\mathcal{I}_1 \cap \mathcal{I}_2$ is the set of all branchings, and a common basis corresponds to arborescences.

Example: Colorful Spanning Trees. Let $G = (V, E)$ where edges in E are colored with k colors. That is, $E = E_1 \uplus E_2 \uplus \ldots \uplus E_k$. Suppose we are given integers $h_1, h_2, \ldots, h_k$ and wish to find a spanning tree that has at most h_i edges of color i (i.e., from E_i). Observe that this can be phrased as a matroid intersection problem: it is the combination of a spanning tree matroid and a partition matroid.

We now state a min-max theorem for the size of the maximum cardinality set in the intersection of two matroids.

Theorem 3 *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be two matroids with rank functions r_1 and r_2. Then the size of the maximum cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$ is given by:*

$$\min_{U \subseteq S} \; r_1(U) + r_2(S \backslash U)$$

Proof: Let $I \in \mathcal{I}_1 \cap \mathcal{I}_2$. Take any set $U \subseteq S$. Then

$$I = |I \cap U| + |I \backslash U| \leq r_1(U) + r_2(S \backslash U)$$

since $I \cap U \in \mathcal{I}_1$ and $I \backslash U \in \mathcal{I}_2$. $\square$

We prove the difficult direction algorithmically. That is, we describe an algorithm for the maximum cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$ that, as a byproduct, proves the other direction.

The algorithm is an "augmenting" path type algorithm inspired by bipartite matching and matroid base exchange properties that we discussed earlier. Given $I \in \mathcal{I}_1 \cap \mathcal{I}_2$, the algorithm outputs a $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ such that $|J| = |I| + 1$, or certifies correctly that I is a maximum cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$ by exhibiting a set $U \subseteq S$ such that $|I| = r_1(U) + r_2(S \backslash U)$.

Recall that for a matroid $M = (S, \mathcal{I})$ and $I \in \mathcal{I}$, we defined a directed graph $D_M(I) = (S, A(I))$ where

$$A(I) = \{(y, z) \mid y \in I,\; z \in S \backslash I,\; I - y + z \in \mathcal{I}\}$$

as a graph that captures exchanges for I.

Now we have two matroids M_1 and M_2 and $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ and we wish to augment I to another set $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ if possible. For this purpose we define a graph $D_{M_1,M_2}(I) = (S, A(I))$ where

$$\begin{aligned} A(I) = \{(y,z) \mid y \in S,\ z \in S\backslash I,\ I - y + z \in \mathcal{I}_1\} \\ \cup \{(z',y') \mid z' \in S\backslash I,\ y' \in I,\ I - y' + z' \in \mathcal{I}_2\} \end{aligned}$$

In other words, $D_{M_1,M_2}(I)$ is the union of $D_{M_1}(I)$ and the reverse of $D_{M_2}(I)$. In this sense there is asymmetry in M_1 and M_2. (An example is shown in Figure 2.)

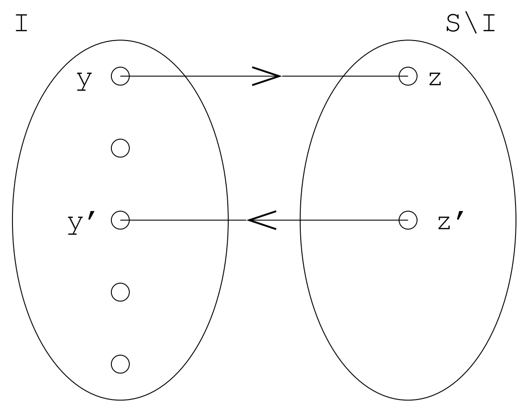


Figure 2: Exchange Graph $D_{M_1,M_2}(I)$

$(y,z) \in A(I) \Rightarrow I - y + z \in \mathcal{I}_1$

$(z',y') \in A(I) \Rightarrow I - y' + z' \in \mathcal{I}_2$

Let $X_1 = \{z \in S\backslash I \mid I + z \in \mathcal{I}_1\}$ and $X_2 = \{z \in S\backslash I \mid I + z \in \mathcal{I}_2\}$, and let P be a shortest path from X_1 to X_2 in $D_{M_1,M_2}(I)$. Note that the shortest path could consist of a single $z \in X_1 \cap X_2$. There may not be any path P between X_1 and X_2.

Lemma 4 *If there is no $X_1 - X_2$ path in $D_{M_1,M_2}(I)$, then I is a maximum cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$.*

Proof: Note that if X_1 or X_2 are empty then I is a base in one of M_1 or M_2 and hence a max cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$. So assume $X_1 \neq \emptyset$ and $X_2 \neq \emptyset$. Let U be the set of nodes that can reach X_2 in $D_{M_1,M_2}(I)$. No $X_1 - X_2$ path implies that $X_1 \cap U = \emptyset$, $X_2 \subseteq U$, and $\delta^-(U) = \emptyset$ (i.e., no arcs enter U). Then we have the following:

Claim 5 $r_1(U) \leq |I \cap U|$

Proof: If $r_1(U) > |I \cap U|$, then $\exists z \in U\backslash(I \cap U)$ such that $(I \cap U) + z \in \mathcal{I}_1$ with $I + z \notin \mathcal{I}_1$. If $I + z \in \mathcal{I}_1$, then $z \in X_1$ and $X_1 \cap U \neq \emptyset$, contradicting the fact that there is no $X_1 - X_2$ path. Since $(I \cap U) + z \in \mathcal{I}_1$ but $I + z \notin \mathcal{I}_1$, there must exist a $y \in I\backslash U$ such that $I - y + z \in \mathcal{I}_1$. But then $(y,z) \in A(I)$, contradicting the fact that $\delta^-(U) = \emptyset$ (shown in Figure 3).

□

Claim 6 $r_2(S\backslash U) \leq |I\backslash U|$ *(The proof is similar to the previous proof.)*

Thus $|I| = |I \cap U| + |I\backslash U| \geq r_1(U) + r_2(S\backslash U)$, which establishes that $|I| = r_1(U) + r_2(S\backslash U)$. Therefore, I is a max cardinality set in $\mathcal{I}_1 \cap \mathcal{I}_2$. □

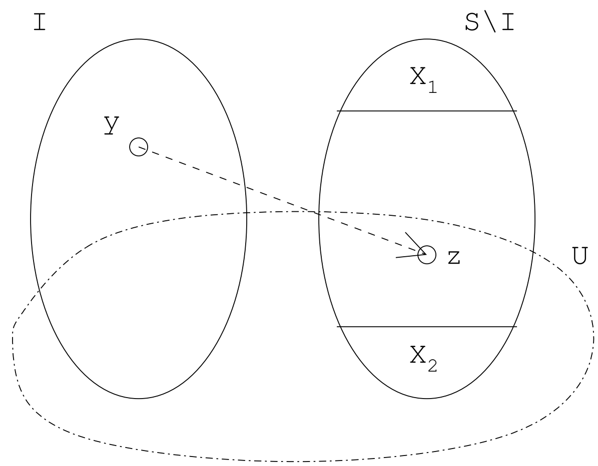


Figure 3: Exchange Graph with a (y, z) arc entering U

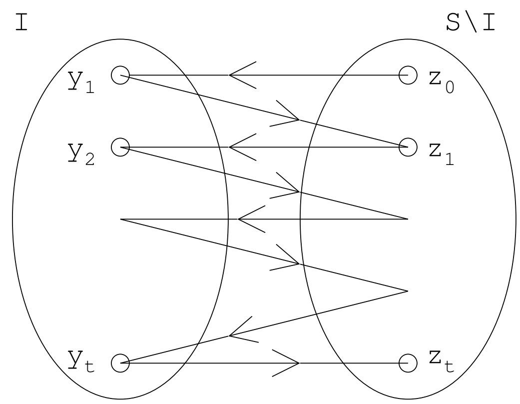


Figure 4: A path P in $D_{M_1,M_2}(I)$

Lemma 7 *If P is a shortest $X_1 - X_2$ path in $D_{M_1,M_2}(I)$, then $I' = I \Delta V(P)$ is in $\mathcal{I}_1 \cap \mathcal{I}_2$.*

Proof: Recall the following lemma from the previous lecture which we will use here:

Lemma 8 *Let $M = (S, \mathcal{I})$ be a matroid. Let $I \in \mathcal{I}$ and $J \subseteq S$ such that $|I| = |J|$. If there is a unique perfect matching on $I \Delta J$ in $A(I)$, then $J \in \mathcal{I}$.*

Let $P = z_0, y_1, z_1, \ldots, y_t, z_t$ (shown in Figure 4) be a shortest path from X_1 to X_2. Let $J = \{z_1, \ldots, z_t\} \cup (I \backslash \{y_1, \ldots, y_t\})$. Then $J \subseteq S$, $|J| = |I|$, and the arcs from $\{y_1, \ldots, y_t\}$ to $\{z_1, \ldots, z_t\}$ form a unique perfect matching from $I \backslash J$ to $J \backslash I$ (otherwise P has a short cut and is not a shortest path). Then by Lemma 8, $J \in \mathcal{I}_1$.

Also, $z_i \notin X_1$ for $i \geq 1$, otherwise P would not be the shortest possible $X_1 - X_2$ path. This implies that $z_i + I \notin \mathcal{I}_1$, which implies that $r_1(I \cup J) = r_1(I) = r_1(J) = |I| = |J|$. Then since $I + z_0 \in \mathcal{I}_1$, it follows that $J + z_0 \in \mathcal{I}_1$ (i.e., $I' = (I \backslash \{y_1, \ldots, y_t\}) \cup \{z_0, z_1, \ldots, z_t\} \in \mathcal{I}_1$).

By symmetry, $I' \in \mathcal{I}_2$. This implies that $I' \in \mathcal{I}_1 \cap \mathcal{I}_2$. $\square$

Theorem 9 *There is a polynomial time algorithm to find a maximum cardinality set in the intersection of two matroids.*

Algorithm 1 will compute a maximum cardinality independent set in the intersection of two matroids M_1 and M_2 in polynomial time. This algorithm can be adapted to find a maximum weight independent set in the intersection of two matroids by adding appropriate weights to the vertices in $D_{M_1,M_2}(I)$ and searching for the shortest weight path with the fewest number of arcs among all such paths of shortest weight.

Algorithm 1 Algorithm for Maximum Cardinality Independent Set in Intersection of Two Matroids

procedure $\textsc{maxIndepSet}(M_1 = (S, \mathcal{I}_1),\ M_2 = (S, \mathcal{I}_2))$
  $I \leftarrow \emptyset$
  repeat
    Construct $D_{M_1,M_2}(I)$
    $X_1 \leftarrow \{z \in S \backslash I \mid I + z \in \mathcal{I}_1\}$
    $X_2 \leftarrow \{z \in S \backslash I \mid I + z \in \mathcal{I}_2\}$
    Let P be a shortest $X_1 - X_2$ path in $D_{M_1,M_2}(I)$
    if P is not empty **then**
      $I \leftarrow I \Delta V(P)$ ▷ $I' = (I \backslash \{y_1, \ldots, y_t\}) \cup \{z_0, z_1, \ldots, z_t\}$
    end if ▷ Else P is empty and I is maximal
  until I is maximal
end procedure

CS 598CSC: Combinatorial Optimization — **Lecture date: March 18, 2010**
Instructor: Chandra Chekuri — **Scribe: Jason Sauppe**

1 Weighted Matroid Intersection

We saw an algorithm for finding a maximum cardinality set in the intersection of two matroids. The algorithm generalized in a straightforward fashion to the weighted case. The correctness is more complicated and we will not discuss it here.

The algorithm for the weighted case is also an augmenting path algorithm. Recall the cardinality algorithm 1:

Algorithm 1 Algorithm for Maximum Cardinality Independent Set in Intersection of Two Matroids

procedure MAXINDEPSET($M_1 = (S, \mathcal{I}_1)$, $M_2 = (S, \mathcal{I}_2)$)
  $I \leftarrow \emptyset$
  repeat
    Construct $D_{M_1,M_2}(I)$
    $X_1 \leftarrow \{z \in S \backslash I \mid I + z \in \mathcal{I}_1\}$
    $X_2 \leftarrow \{z \in S \backslash I \mid I + z \in \mathcal{I}_2\}$
    Let P be a shortest $X_1 - X_2$ path in $D_{M_1,M_2}(I)$
    if P is not empty **then**
      $I \leftarrow I \Delta V(P)$ ▷ $I' = (I \backslash \{y_1, \ldots, y_t\}) \cup \{z_0, z_1, \ldots, z_t\}$
    end if   ▷ Else P is empty and I is maximal
  until I is maximal
end procedure

The weighted case differs only in finding P. Let $w : S \to \Re^+$ be the weights. Then in computing P we assign weights to each vertex $x \in D_{M_1,M_2}(I)$ as $w(x)$ if $x \in I$ and $-w(x)$ to $x \notin I$. The desired path P should now be a minimum length path according to the weights; further, P should have the smallest number of arcs among all minimum length paths.

Theorem 1 *There is a polynomial time combinatorial algorithm for weighted matroid intersection.*

2 Matroid Intersection Polytope

Edmonds proved the following theorem about the matroid intersection polytope:

Theorem 2 *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be two matroids on S. Then the convex hull of the characteristic vectors of sets in $\mathcal{I}_1 \cap \mathcal{I}_2$ is determined by the following set of inequalities:*

$$\begin{aligned} x &\geq 0 \\ x(U) &\leq r_1(U) \qquad \forall\, U \subseteq S \\ x(U) &\leq r_2(U) \qquad \forall\, U \subseteq S \end{aligned}$$

where r_1 and r_2 are the rank functions of M_1 and M_2, respectively. Moreover, the system of inequalities is TDI. In other words,

$$P_{\textit{common indep. set}}(M_1, M_2) = P_{\textit{indep. set}}(M_1) \cap P_{\textit{indep. set}}(M_2).$$

Proof: Consider the primal-dual pair

$$\begin{aligned} \max \sum_{e \in S} w(e)x(e) & \\ \text{subject to} \quad x(U) \leq r_1(U) & \qquad \forall\, U \subseteq S \\ x(U) \leq r_2(U) & \qquad \forall\, U \subseteq S \\ x \geq 0 & \end{aligned}$$

$$\begin{aligned} \min \sum_{U \subseteq S} \Big(r_1(U) y_1(U) + r_2(U) y_2(U) \Big) & \\ \text{subject to} \quad \sum_{\substack{U \subseteq S \\ U \ni e}} \Big(y_1(U) + y_2(U) \Big) \geq w(e) & \qquad \forall\, e \in S \\ y_1 \geq 0 & \\ y_2 \geq 0 & \end{aligned}$$

We will prove that the dual has an integral optimum solution whenever w is integral. We can assume that $w(e) \geq 0$ for each e without loss of generality.

Lemma 3 *There exists an optimum solution y_1^*, y_2^* to the dual such that*

$$\mathcal{F}_1 = \{U \subseteq S \mid y_1^*(U) > 0\}$$
$$\mathcal{F}_2 = \{U \subseteq S \mid y_2^*(U) > 0\}$$

are chains.

Proof: Suppose that no optimum solution to the dual satisfies the above property. Then choose an optimum y_1^*, y_2^* with $\mathcal{F}_1 = \{U \subseteq S \mid y_1^*(U) > 0\}$ and $\mathcal{F}_2 = \{U \subseteq S \mid y_2^*(U) > 0\}$ such that the number of proper intersections plus the number of disjoint sets in $\mathcal{F}_1$ and $\mathcal{F}_2$ is minimal.

Then for $A, B \in \mathcal{F}_1$, if A and B properly intersect or are disjoint, we can increase $y_1^*(A \cap B)$ and $y_1^*(A \cup B)$ by ϵ and decrease $y_1^*(A)$ and $y_1^*(B)$ by ϵ to create a new dual solution. This new solution is still dual feasible since

$$\chi(A \cup B) + \chi(A \cap B) = \chi(A) + \chi(B).$$

and the dual objective value changes by

$$-\epsilon\Big(r_1(A) + r_1(B)\Big) + \epsilon\Big(r_1(A \cup B) + r_1(A \cap B)\Big).$$

By the submodularity of r_1, this is ≤ 0. If this value is < 0, then this contradicts the optimality of the original solution y_1^*, y_2^*. On the other hand, if this value equals 0, then we have a new optimum solution for the dual with a smaller number of proper intersections plus disjoint sets in $\mathcal{F}_1, \mathcal{F}_2$, contradicting the choice of y_1^*, y_2^*. This follows similarly for $A, B \in \mathcal{F}_2$. $\square$

Corollary 4 *There exists a vertex solution y_1^*, y_2^* such that the support of y_1^* and y_2^* are chains.*

Proof Sketch. Suppose that no vertex solution to the dual satisfies this property. Then choose a vertex solution y_1^*, y_2^* with $\mathcal{F}_1 = \{U \subseteq S \mid y_1^*(U) > 0\}$ and $\mathcal{F}_2 = \{U \subseteq S \mid y_2^*(U) > 0\}$ such that the number of proper intersections plus the number of disjoint sets in $\mathcal{F}_1$ and $\mathcal{F}_2$ is minimal. Perform the uncrossing technique as in the previous proof.

Then for all $e \in S$, the constraint for e remains tight after uncrossing. This holds trivially for $e \notin A \cup B$. If $e \in A$ but $e \notin B$, then $e \in A \cup B$ but $e \notin A \cap B$, so the net change in the constraint for e is $\epsilon - \epsilon = 0$ and it therefore remains at equality (similarly for $e \in B$ but $e \notin A$). If $e \in A \cap B$, then the net change in the constraint for e is $2\epsilon - 2\epsilon = 0$ and the contraint remains tight.

The uncrossing technique then creates a new vertex solution with fewer proper intersections plus disjoint sets in $\mathcal{F}_1$ and $\mathcal{F}_2$, contradicting the choice of y_1^*, y_2^*. $\square$

The following very useful lemma was shown by Edmonds.

Lemma 5 *Let S be a set and $\mathcal{F}_1$ and $\mathcal{F}_2$ be two laminar families on S. Let $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ and let A be the $S \times \mathcal{F}$ incidence matrix. Then A is TUM.*

Proof: By $S \times \mathcal{F}$ incidence matrix we mean $A_{x,C} = 1$ if $x \in C$ where $x \in S$, $C \in \mathcal{F}$, and $A_{x,C} = 0$ otherwise. Without loss of generality, we can assume that each $x \in S$ appears in at least one set C in either $\mathcal{F}_1$ or $\mathcal{F}_2$, otherwise we can remove x from S without affecting A (since the row for x would consist of all 0's).

Let A be a counterexample with $|\mathcal{F}| + |S|$ minimal, and among such with a minimal number of 1's in A. There are two cases to consider: First, if $\mathcal{F}_1$ and $\mathcal{F}_2$ are collections of disjoint sets, then every row of A has at most two nonzero entries. If every row of A has exactly two nonzero entries, then A represents the edge-vertex incidence matrix of a bipartite graph, which is TUM (see Figure 1). Since A is a counterexample, this cannot be the case. Therefore, at least one row of A must have only one nonzero entry.

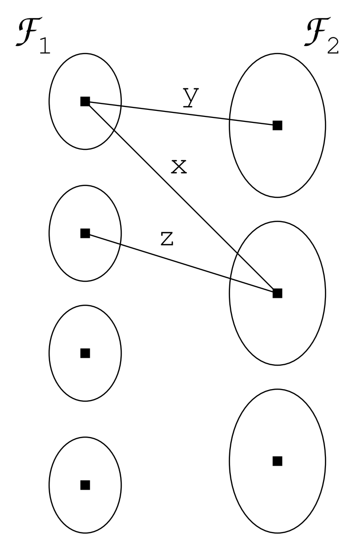


Figure 1: Case where $\mathcal{F}_1$ and $\mathcal{F}_2$ are collections of disjoint sets

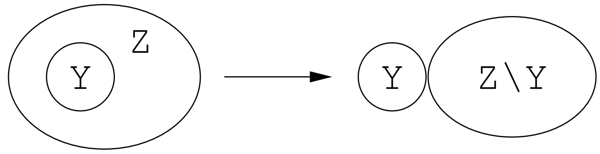


Figure 2: Replacing $Y, Z \in \mathcal{F}_1$ with $Y, Z\backslash Y$

Let A' be the the matrix consisting of the rows of A with two nonzero entries. Then A' is TUM (for the reason given before). We claim that this implies that A is also TUM: for any square submatrix B of A, the determinant of B can be computed by first expanding the computation along the rows of B with only one nonzero entry. The resulting submatrix represents a square submatrix of A', which does not have determinant -2 or 2. Therefore, the original submatrix B of A cannot have determinant -2 or 2. This is true for any square submatrix of A, so A must be TUM.

This means that $\mathcal{F}_1$ and $\mathcal{F}_2$ cannot be collections of disjoint sets if A is to be a counterexample. Therefore, at least one of $\mathcal{F}_1$ and $\mathcal{F}_2$ must have two sets that are not disjoint. Without loss of generality, assume that $\mathcal{F}_1$ has at least two sets Z and Y such that $Y \subset Z$. Of all possible Z and Y meeting this criteria, choose the smallest Z. Then replacing Z by $Z\backslash Y$ generates another laminar family $\mathcal{F}_1'$.

Let A' be the incidence matrix for S and $\mathcal{F}' = \mathcal{F}_1' \cup \mathcal{F}_2$.

Claim 6 *A' is TUM if and only if A is TUM.*

A' is obtained from A by subtracting the column for Y from the column for Z. Hence the determinants of all submatrices are preserved. Also, A' has fewer 1's than A since $Y \neq \emptyset$. Since A was chosen as a counterexample with the smallest number of 1's, A' cannot be a counterexample, and thus A' is TUM. But this implies that A is TUM, as well. Therefore, no such counterexample A can exist. □

Let y_1^*, y_2^* be a vertex solution such that $\mathcal{F}_1$ and $\mathcal{F}_2$ are chains. Since y_1^*, y_2^* is a vertex solution, we have a subset $\mathcal{F} \subseteq \mathcal{F}_1 \cup \mathcal{F}_2$ such that (y_1^*, y_2^*) is a solution to the system of equalities

$$\sum_{\substack{U \in \mathcal{F} \\ U \ni e}} \Big(y_1(U) + y_2(U) \Big) = w(e) \qquad \forall\ e \in S.$$

Then by Lemma 5, the constraint matrix for the above system corresponds to a TUM matrix. This implies that there is an integral solution y_1, y_2 for integral w. From this we can conclude that the dual LP has an integral optimum solution whenever w is integral, and therefore the system of inequalities for the matroid intersection polytope is TDI. □

1 Matroid Union

Matroid union and matroid intersection are closely related in the sense that one can be derived from the other. However they are from different perspectives and have different applications.

To motivate matroid union theorem we state a well known theorem of Tutte and Nash-Williams on packing disjoint spanning trees in graphs.

Theorem 1 (Nash-Williams and Tutte) *An undirected multi-graph $G = (V, E)$ contains k edge-disjoint spanning trees iff for every partition P of V into ℓ sets, $V_1, V_2, \ldots, V_\ell$, the number of edges crossing the partition P is at least $k(\ell - 1)$.*

It is easy to see that the condition is necessary; if $T_1, \ldots, T_k$ are the edge-disjoint spanning trees then each T_i has to contain at least $\ell - 1$ edges across the partition P to connect them. A useful corollary of the above was observed by Gusfield. It is an easy exercise to derive this from the above theorem.

Corollary 2 *If a multi-graph $G = (V, E)$ is $2k$-edge-connected then G contains k edge-disjoint spanning trees.*

Nash-Williams proved a related theorem on covering the edge-set of a graph by forests.

Theorem 3 (Nash-Williams) *Let $G = (V, E)$ be an undirected multi-graph. Then E can be partitioned into k forests iff for each set $U \subseteq V$,*

$$|E[U]| \leq k(|U| - 1). \tag{1}$$

Again, necessity is easy to see; any forest can contain at most $|U| - 1$ edges from $E[U]$. The above two theorems were first shown via graph theoretica arguments but turn out to be special cases of the matroid union theorem, and hence are properly viewed as matroidal results. We start with a basic result of Nash-Williams that gives a clean proof of the matroid union theorem to follow.

Theorem 4 (Nash-Williams) *Let $\mathcal{M}' = (S', \mathcal{I}')$ be a maroid with rank function r'. Let $f : S' \to S$ be a function mapping S' to S. Let $\mathcal{M} = (S, \mathcal{I})$, where $\mathcal{I} = \{f(I') | I' \in \mathcal{I}'\}$. Then $\mathcal{M}$ is a matroid with rank function r, where*

$$r(U) = \min_{T \subseteq U} (|U \backslash T| + r'(f^{-1}(T))). \tag{2}$$

Proof:

We verify the three axioms.

1. $f(\emptyset) = \emptyset$ and hence $\emptyset \in \mathcal{I}$.

2. Say $A \in \mathcal{I}$ and $B \subseteq A$. Then

$$\begin{aligned} A \in \mathcal{I} \quad &\Rightarrow \quad \exists A' \in \mathcal{I}', \text{ s.t. } f(A') = A \\ &\Rightarrow \quad \forall u \in A, f^{-1}(u) \cap A' \neq \emptyset. \end{aligned}$$

   Let $B' = \{u' \in A' | f(u') \in B\}$, then $B = f(B')$ and since $B' \subseteq A', B' \in \mathcal{I}'$ and hence $B \in \mathcal{I}$.

3. Say $A, B \in \mathcal{I}$ and $|B| > |A|$. Let A' be minimal s.t. $f(A') = A$. Similarly let B' be minimal s.t. $f(B') = B$. Then

$$|f^{-1}(u) \cap A'| = 1, \forall u \in A.$$

   Similarly,

$$|f^{-1}(u) \cap B'| = 1, \forall u \in B.$$

   Therefore,

$$|A'| = |A| \text{ and } |B'| = |B|.$$

   Since

$$A', B' \in \mathcal{I}' \text{ and } |B'| > |A'|,$$

$$\Rightarrow \exists u' \in B' \backslash A', \text{ s.t. } A' + u' \in \mathcal{I}'.$$

   Then

$$A + f(u') \in \mathcal{I} \text{ and } f(u') \in B \backslash A.$$

Therefore $\mathcal{M}$ is a matroid.

We now derive the rank formula for $\mathcal{M}$. Although one can derive it from elementary methods, it is easy to obtain it from the matroid intersection theorem. Recall that if $\mathcal{M}_1 = (N, \mathcal{I}_1)$ and $\mathcal{M}_2 = (N, \mathcal{I}_2)$ are two matroids on N, then the max cardinality of a common independent set in $\mathcal{I}_1 \wedge \mathcal{I}_2$ is given by

$$\min_{X \subseteq N} r_1(X) + r_2(N \backslash X).$$

Now consider $U \subseteq S$. Let $U' = f^{-1}(U)$. We observe that $A \subseteq U$ is independent in $\mathcal{I}$ iff there is an $A' \subseteq f^{-1}(U)$ such that $|A'| = |A|$, $f(A') = A$ and A' is independent in $\mathcal{I}'$.

Define a matroid $\mathcal{M}'' = (S', \mathcal{I}'')$, where

$$\mathcal{I}'' = \{I \subseteq f^{-1}(U) \mid |I \cap f^{-1}(U)| \leq 1, u \in U\}.$$

Note that $\mathcal{M}''$ is a partition matroid. Let r'' be the rank of $\mathcal{M}''$. We leave the following claim as an exercise.

Claim 5 *$r(U)$ is the size of a maximum cardinality independent set in $\mathcal{M}' \wedge \mathcal{M}''$.*

Therefore, by the matroid intersection theorem we have that

$$r(U) = \min_{T \subseteq U'} (r'(T) + r''(U' \setminus T)) = \min_{T \subseteq U} (r'(f^{-1}(T)) + |U \setminus T|),$$

using the fact that $\mathcal{M}''$ is a partition matroid. We leave it to the reader to verify the second equality in the above. □

From the above we obtain the matroid union theorem before that was formulated by Edmonds. Let $\mathcal{M}_1 = (S_1, \mathcal{I}_1), \ldots, \mathcal{M}_k = (S_k, \mathcal{I}_k)$ be matroids. Define

$$\mathcal{M} = \mathcal{M}_1 \vee \mathcal{M}_2 \vee \cdots \vee \mathcal{M}_k = (S_1 \cup S_2 \cup \cdots \cup S_k, \mathcal{I}),$$

where

$$\mathcal{I} = \mathcal{I}_1 \vee \mathcal{I}_2 \vee \cdots \vee \mathcal{I}_k := \{I_1 \cup I_2 \cup \cdots \cup I_k | I_i \in \mathcal{I}_i, 1 \leq i \leq k\}.$$

Theorem 6 (Matroid Union) *Let $\mathcal{M}_1 = (S_1, \mathcal{I}_1), \ldots, \mathcal{M}_k = (S_k, \mathcal{I}_k)$ be matroids. Then*

$$\mathcal{M} = \mathcal{M}_1 \vee \mathcal{M}_2 \vee \cdots \vee \mathcal{M}_k \tag{3}$$

is a matroid. The rank function of $\mathcal{M}$ is given by r, where

$$r(U) = \min_{T \subseteq U} (|U \backslash T| + r_1(T \cap S_1) + \cdots + r_k(T \cap S_k)). \tag{4}$$

Proof: Let $S'_1, \ldots, S'_k$ be copies of $S_1, \ldots, S_k$, such that

$$S'_i \cap S'_j = \emptyset, i \neq j.$$

Let $\mathcal{M}'_i = (S'_i, \mathcal{I}'_i)$, where $\mathcal{I}'_i$ corresponds to $\mathcal{I}_i$. Let $S' = S'_1 \uplus S'_2 \uplus \cdots \uplus S'_k$ and define $\mathcal{M}' = (S', \mathcal{I}')$, where

$$\mathcal{I}' = \{I'_1 \cup I'_2 \cup \cdots \cup I'_k \mid I'_i \in \mathcal{I}_i\}.$$

Clearly $\mathcal{M}'$ is a matroid since it is disjoint union of matroids.

Now define $f : S' \to S$ where $S = S_1 \cup S_2 \cup \cdots \cup S_k$, and $f(s') = s$ if s' is the copy of s. Then $\mathcal{M}$ is obtained from $\mathcal{M}'$ by f and hence by Theorem 4, $\mathcal{M}$ is a matroid. The rank formula easily follows by applying the formula in Theorem 4 $\mathcal{M}'$ and $\mathcal{M}$. □

The above theorem is also referred to as the matroid *partition* theorem for the following reason. A $U \in S$ is $\mathcal{M}$ independent iff U can be partitioned into $U_1, \ldots, U_k$, such that for $1 \leq i \leq k$, U_i is independent in $\mathcal{I}_i$; note that U_i are allowed to be $\emptyset$.

We state a useful corollary.

Corollary 7 *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid and k be an integer. Then the maximum rank of the union of k independent sets of $\mathcal{M}$ is equal to*

$$\min_{U \subseteq S} (|S \setminus U| + k \cdot r(U)). \tag{5}$$

Proof: Take $\mathcal{M}'$ to be union of $\mathcal{M}_1 \vee \mathcal{M}_2 \vee \cdots \vee \mathcal{M}_k$, where $\mathcal{M}_i = \mathcal{M}$. Then the union of k independent sets in $\mathcal{M}$ is an independent set in $\mathcal{M}'$. Thus we are asking for the maximum possible rank in $\mathcal{M}'$. S achieves the maximum rank and by the previous theorem

$$\begin{aligned} r'(S) &= \min_{U \subseteq S}(|S \setminus U| + k \cdot r(S \cap U)) \quad (6)\\ &= \min_{U \subseteq S}(|S \setminus U| + k \cdot r(U)). \quad (7) \end{aligned}$$

□

We now easily derive two important theorems that were first stated by Edmonds.

Theorem 8 (Matroid base covering theorem) *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. Then S can be covered by k independent sets iff*

$$|U| \leq k \cdot r(U), \forall U \subseteq S. \quad (8)$$

Proof: S can be covered by k independent sets iff the rank of S in the union of $\mathcal{M}_1 \vee \mathcal{M}_2 \vee \cdots \vee \mathcal{M}_k$, where $\mathcal{M}_i = \mathcal{M}$, is equal to $|S|$. By Corollary 7, this is equivalent to

$$|S \setminus U| + k \cdot r(U) \geq |S|, \forall U \subseteq S$$

$$\Rightarrow k \cdot r(U) \geq |U|, \forall U \subseteq S.$$

□

Exercise 9 *Derive Nash-Williams forest-cover theorem (Theorem 3) as a corollary.*

Now we derive the matroid base packing theorem, also formulated by Edmonds.

Theorem 10 (Matroid Base Packing Theorem) *Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid. Then there are k disjoint bases in $\mathcal{M}$ iff*

$$k(r(S) - r(U)) \leq |S \setminus U|, \forall U \subseteq S. \quad (9)$$

Proof: To see necessity, consider any set $U \subseteq S$. Any base B has the property that $r(B) = r(S)$. And $r(B \cap U) \leq r(U)$. Thus

$$B \cap (S \setminus U) \geq r(S) - r(U).$$

Therefore if there are k disjoint bases then each of these bases requires $r(S) - r(U)$ distinct elements from $S \setminus U$, and hence

$$k(r(S) - r(U)) \leq |S \setminus U|.$$

For sufficiency, we take the k-fold union of $\mathcal{M}$ and there are k disjoint bases if $r'(S)$ in the union matroid $\mathcal{M}'$ satisfies the equation

$$r'(S) = k \cdot r(S)$$

in other words,

$$\min_{U \subseteq S} |S \setminus U| + k \cdot r(U) = k \cdot r(S)$$

$$\Rightarrow |S \setminus U| + k \cdot r(U) \geq k \cdot r(S)|$$

□

Exercise 11 *Derive Nash-Williams-Tutte theorem on packing spanning trees (Theorem 1) as a corollary.*

2 Algorithmic and Polyhedral Aspects

Let $\mathcal{M} = \mathcal{M}_1 \vee \mathcal{M}_2 \vee \cdots \vee \mathcal{M}_k$. Algorithmic results for $\mathcal{M}$ follow from an independence oracle or rank oracle for $\mathcal{M}$. Recall that a set $I \in \mathcal{I}$ is independent in $\mathcal{M}$ iff I an be partitioned into $I_1, I_2, \ldots, I_k$ such that for $1 \leq i \leq k$, I_i is independent in $\mathcal{I}_i$. Note that this is non-trivial to solve.

Theorem 12 *Given rank functions $r_1, \ldots, r_k$ for $\mathcal{M}_1, \ldots, \mathcal{M}_k$, as polynomial time oracles, there is a polynomial time algorithm to implement the rank function oracle r for $\mathcal{M} = \mathcal{M}_1 \vee \mathcal{M}_2 \vee \cdots \vee \mathcal{M}_k$.*

We sketch the proof of the above theorem. Recall the construction in Theorem 6 that showed $\mathcal{M}$ is a matroid. We first constructed an intermediate matroid $\mathcal{M}'$ by taking copies of $\mathcal{M}_1, \ldots, \mathcal{M}_k$ and then applied Theorem 4 to map $\mathcal{M}'$ to $\mathcal{M}$.

For the matroid $\mathcal{M}'$, one easily obtains an algorithm to implement r' from $r_1, \ldots, r_k$, i.e.

$$r'(U) = \sum_{i=1}^{k} r_i(U \cap S_i').$$

Recall that we obtained the rank function r for $\mathcal{M}$ from r' for $\mathcal{M}'$ using matroid intersection (see proof of Theorem 4). Thus, one can verify that an algorithm for matroid intersection implies an algorithm for r using algorithms for $r_1, \ldots, r_k$. There is also a direct algorithm that avoids using the matroid intersection algorithm — see [1] for details.

Polyhedrally, the base covering and packing theorems imply and are implied by the following

Theorem 13 *Given a matroid $\mathcal{M} = (S, \mathcal{I})$, the independent set polytope and base polytope of $\mathcal{M}$ have the integer decomposition property.*

Exercise 14 *Prove the above theorem using Theorem 8 and 10.*

Capacitated case and algorithmic aspects of packing and covering: The matroid union algorithm allows us to obtain algorithmic versions of the matroid base covering and base packing theorems. As a consequence, for example, there is a polynomial time algorithm that given a multi-graph $G = (V, E)$, outputs the maximum number of edge-disjoint spanning trees in G. It is also possible to solve the capacitated version of the problems in polynomial time. More precisely, let $\mathcal{M} = (S, \mathcal{I})$ and let $c : S \to \mathcal{Z}_+$ be integer capacities on the elements of S. The capacitated version of the base packing theorem is to ask for the maximum number of bases such that no element $e \in S$ is in more than $c(e)$ bases. Similarly, for the base covering theorem, one seeks a minimum number of independent sets such that each element e is in at least $c(e)$ independent sets. The capacitated case be handled by making $c(e)$ copies of each element e, however, this would give only a pseudo-polynomial time algorithm.

Assuming we have a polynomial time rank oracle for $\mathcal{M}$, the following capaciatated problems can be solved in polynomial time. To solve the capacitated versions, one needs polyhedral methods; see [1] for more details.

1. fractional packing of bases, i.e., let $\mathcal{B}$ denote the set of bases of $\mathcal{M}$,

$$\max_{B \in \mathcal{B}} \lambda_B$$

$$\sum_{B \ni e} \lambda_B \leq c(e), \forall e \in S$$

$$\lambda_B \geq 0$$

2. integer packing of bases, same as above but λ_B are restricted to be integer.

3. fractional covering by independent sets, i.e.

$$\min_{I \in \mathcal{I}} \lambda_I$$

$$\sum_{I \ni e} \lambda_I \geq c(e), \forall e \in S$$

$$\lambda \geq 0$$

4. integer covering by independent sets, same as above but λ_I are constrained to be integer.

Matroid Intersection from Matroid Union: We have seen that the matroid union algorithm follows from an algorithm for matroid intersection. The converse can also be shown. To see this, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two matroids on the same ground set S. Then, one can find the maximum cardinality common independent set in $\mathcal{M}_1 \wedge \mathcal{M}_2$ be considering $\mathcal{M}_1 \vee \mathcal{M}_2^*$ where $\mathcal{M}_2^*$ is the dual of $\mathcal{M}_2$; See Problem 4 in Homework 3 for details on this.

References

[1] Alexander Schrijver, "Combinatorial Optimization: Polyhedra and Efficiency", Chapter 42, Vol B, Springer-Verlag 2003.

CS 598CSC: Combinatorial Optimization	**Lecture data: 4/8/2010**
Instructor: Chandra Chekuri	**Scribe: Bolin Ding**

1 Introduction to Submodular Set Functions and Polymatroids

Submodularity plays an important role in combinatorial optimization. Given a finite ground set S, a *set function* $f : 2^S \to \mathbb{R}$ is *submodular* if

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \quad \forall A, B \subseteq S;$$

or equivalently,

$$f(A + e) - f(A) \geq f(B + e) - f(B) \quad \forall A \subseteq B \text{ and } e \in S \setminus B.$$

Another equivalent definition is that

$$f(A + e_1) + f(A + e_2) \geq f(A) + f(A + e_1 + e_2) \quad \forall A \subseteq S \text{ and distinct } e_1, e_2 \in S \setminus A.$$

Exercise: Prove the equivalence of the above three definitions.

A set function $f : 2^S \to \mathbb{R}$ is *non-negative* if $f(A) \geq 0\ \forall A \subseteq S$. f is *symmetric* if $f(A) = f(S \setminus A)\ \forall A \subseteq S$. f is *monotone* (*non-decreasing*) if $f(A) \leq f(B)\ \forall A \subseteq B$. f is *integer-valued* if $f(A) \in \mathbb{Z}\ \forall A \subseteq S$.

1.1 Examples of submodular functions

Cut functions. Given an undirected graph $G = (V, E)$ and a 'capacity' function $c : E \to \mathbb{R}_+$ on edges, the *cut function* $f : 2^V \to \mathbb{R}_+$ is defined as $f(U) = c(\delta(U))$, *i.e.*, the sum of capacities of edges between U and $V \setminus U$. f is submodular (also non-negative and symmetric, but not monotone).

In an undirected hypergraph $G = (V, \mathcal{E})$ with capacity function $c : \mathcal{E} \to \mathbb{R}_+$, the *cut function* is defined as $f(U) = c(\delta_{\mathcal{E}}(U))$, where $\delta_{\mathcal{E}}(U) = \{e \in \mathcal{E} \mid e \cap U \neq \emptyset \text{ and } e \cap (S \setminus U) \neq \emptyset\}$.

In a directed graph $D = (V, A)$ with capacity function $c : A \to \mathbb{R}_+$, the *cut function* is defined as $f(U) = c(\delta_{\text{out}}(U))$, where $\delta_{\text{out}}(U)$ is the set of arcs leaving U.

Matroids. Let $M = (S, \mathcal{I})$ be a matroid. Then the rank function $r_M : 2^S \to \mathbb{R}_+$ is submodular (also non-negative, integer-valued, and monotone).

Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be two matroids. Then the function f given by $f(U) = r_{M_1}(U) + r_{M_2}(S \setminus U)$, for $U \subseteq S$, is submodular (also non-negative, and integer-valued). By the matroid intersection theorem, the minimum value of f is equal to the maximum cardinality of a common independent set in the two matroids.

Coverage in set system. Let $T_1, T_2, \ldots, T_n$ be subsets of a finite set T. Let $S = [n] = \{1, 2, \ldots, n\}$ be the ground set. The *coverage function* $f : 2^S \to \mathbb{R}_+$ is defined as $f(A) = |\cup_{i \in A} T_i|$.

A generalization is obtained by introducing the weights $w : T \to \mathbb{R}_+$ of elements in T, and defining the weighted coverage $f(A) = w\left(\cup_{i \in A} T_i\right)$.

Another generalization is to introduce a submodular and monotone weight-function $g : 2^T \to \mathbb{R}_+$ of subsets of T. Then the function f is defined as $f(A) = g\left(\cup_{i \in A} T_i\right)$.

All the three versions of f here are submodular (also non-negative, and monotone).

Flows to a sink. Let $D = (V, A)$ be a directed graph with an arc-capacity function $c : A \to \mathbb{R}_+$. Let a vertex $t \in V$ be the *sink*. Consider a subset $S \subseteq V \setminus \{t\}$ of vertices. Define a function $f : 2^S \to \mathbb{R}_+$ as $f(U) =$ max flow from U to t in the directed graph D with edge capacities c, for a set of 'sources' U. Then f is submodular (also non-negative and monotone).

Max element. Let S be a finite set and let $w : S \to \mathbb{R}$. Define a function $f : 2^S \to \mathbb{R}$ as $f(U) = \max\{w(u) \mid u \in U\}$ for nonempty $U \subseteq S$, and $f(\emptyset) = \min\{w(u) \mid u \in S\}$. Then f is submodular (also monotone).

Entropy and Mutual information. Let $X_1, X_2, \ldots, X_n$ be random variables over some underlying probability space, and $S = \{1, 2, \ldots, n\}$. For $A \subseteq S$, define $X_A = \{X_i \mid i \in A\}$ to be the set of random variables with indices in A. Then $f(A) = H(X_A)$, where $H(\cdot)$ is the entropy function, is submodular (also non-negative and monotone). Also, $f(A) = I(X_A; X_{S \setminus A})$, where $I(\cdot;\cdot)$ is the mutual information of two random variables, is submodular.

Exercise: Prove the submodularity of the functions introduced in this subsection.

1.2 Polymatroids

Define two polyhedra associated with a set function f on S:

$$P_f = \{x \in \mathbb{R}^S \mid x(U) \le f(U)\ \forall U \subseteq S,\ x \ge \mathbf{0}\} \text{ and } EP_f = \{x \in \mathbb{R}^S \mid x(U) \le f(U)\ \forall U \subseteq S\}.$$

If f is a submodular function, then P_f is called the *polymatroid associated with* f, and EP_f the *extended polymatroid associated with* f. A polyhedron is called an (extended) polymatroid if it is the (extended) polymatroid associated with some submodular function. Since $0 \le x_s \le f(\{s\})$ for each $s \in S$, a polymatroid is bounded, and hence is a polytope.

An observation is that P_f is non-empty iff $f \ge \mathbf{0}$, and EP_f is non-empty iff $f(\emptyset) \ge 0$.

If f is the rank function of a matroid M, then P_f is the independent set polytope of M.

A vector x in EP_f (or in P_f) is called a *base vector* of EP_f (or of P_f) if $x(S) = f(S)$. A *base vector* of f is a base vector of EP_f. The set of all base vectors of f is called the *base polytope* of EP_f or of f. It is a face of EP_f and denoted by B_f:

$$B_f = \{x \in \mathbb{R}^S \mid x(U) \le f(U)\ \forall U \subseteq S,\ x(S) = f(S)\}.$$

B_f is a polytope, since $f(\{s\}) \ge x_s = x(S) - x(S \setminus \{s\}) \ge f(S) - f(S \setminus \{s\})$ for each $s \in S$.

The following claim is about the set of tight constraints in the extended polymatroid associated with a submodular function f.

Claim 1 *Let $f : 2^S \to \mathbb{R}$ be a submodular set function. For $x \in EP_f$, define $\mathcal{F}_x = \{U \subseteq S \mid x(U) = f(U)\}$ (tight constraints). Then $\mathcal{F}_x$ is closed under taking unions and intersections.*

Proof: Consider any two sets $U, V \in \mathcal{F}_x$, we have

$$f(U \cup V) \ge x(U \cup V) = x(U) + x(V) - x(U \cap V) \ge f(U) + f(V) - f(U \cap V) \ge f(U \cup V).$$

Therefore, $x(U \cup V) = f(U \cup V)$ and $x(U \cap V) = f(U \cap V)$. □

Given a submodular set function f on S and a vector $a \in \mathbb{R}^S$, define the set function $f|a$ as

$$(f|a)(U) = \min_{T \subseteq U}(f(T) + a(U \setminus T)).$$

Claim 2 *If f is a submodular set function on S, $f|a$ is also submodular.*

Proof: Let $g = f|a$ for the simplicity of notation. For any $X, Y \subseteq S$, let $X' \subseteq X$ s.t. $g(X) = f(X') + a(X \setminus X')$, and $Y' \subseteq Y$ s.t. $g(Y) = f(Y') + a(Y \setminus Y')$. Then, from the definition of g,

$$g(X \cap Y) + g(X \cup Y) \leq \big(f(X' \cap Y') + a((X \cap Y) \setminus (X' \cap Y'))\big) + \big(f(X' \cup Y') + a((X \cup Y) \setminus (X' \cup Y'))\big).$$

From the submodularity of f,

$$f(X' \cap Y') + f(X' \cup Y') \leq f(X') + f(Y').$$

And from the modularity of a,

$$\begin{aligned} a((X \cap Y) \setminus (X' \cap Y')) + a((X \cup Y) \setminus (X' \cup Y')) &= a(X \cap Y) + a(X \cup Y) - a(X' \cap Y') - a(X' \cup Y') \\ &= a(X) + a(Y) - a(X') - a(Y'). \end{aligned}$$

Therefore, we have $g(X \cap Y) + g(X \cup Y) \leq f(X') + f(Y') + a(X \setminus X') + a(Y \setminus Y')$. □

What is $EP_{f|a}$ and $P_{f|a}$? We have the following claim.

Claim 3 *If f is a submodular set function on S and $f(\emptyset) = 0$, $EP_{f|a} = \{x \in EP_f \mid x \leq a\}$ and $P_{f|a} = \{x \in P_f \mid x \leq a\}$.*

Proof: For any $x \in EP_{f|a}$ and any $U \subseteq S$, we have that $x(U) \leq (f|a)(U) \leq f(U) + a(U \setminus U) = f(U)$ implying $x \in EP_f$, and that $x(U) \leq (f|a)(U) \leq f(\emptyset) + a(U \setminus \emptyset) = a(U)$, implying $x \leq a$.

For any $x \in EP_f$ with $x \leq a$ and any $U \subseteq S$, suppose that $(f|a)(U) = f(T) + a(U \setminus T)$. Then we have, $x(U) = x(T) + x(U \setminus T) \leq f(T) + a(U \setminus T) = (f|a)(U)$, implying $x \in EP_{f|a}$.

The proof of $P_{f|a} = \{x \in P_f \mid x \leq a\}$ is similar. □

A special case of the above claim is that when $a = \mathbf{0}$, then $(f|\mathbf{0})(U) = \min_{T \subseteq U} f(T)$ and $EP_{f|\mathbf{0}} = \{x \in EP_f \mid x \leq \mathbf{0}\}$.

2 Optimization over Polymatroids by the Greedy Algorithm

Let $f : 2^S \to \mathbb{R}$ be a submodular function and assume it is given as a value oracle. Also given a weight vector $w : S \to \mathbb{R}_+$, we consider the problem of maximizing $w \cdot x$ over EP_f.

$$\begin{aligned} &\max w \cdot x \\ &x \in EP_f. \end{aligned} \tag{1}$$

Edmonds showed that the greedy algorithm for matroids can be generalized to this setting.

We assume (or require) that $w \geq \mathbf{0}$, because otherwise, the maximum value is unbounded. W.l.o.g., we can assume that $f(\emptyset) = 0$: if $f(\emptyset) < 0$, $EP_f = \emptyset$; and if $f(\emptyset) > 0$, setting $f(\emptyset) = 0$ does not violate the submodularity.

Greedy algorithm and integrality. Consider the following greedy algorithm:

1. Order $S = \{s_1, s_2, \ldots, s_n\}$ s.t. $w(s_1) \geq \ldots \geq w(s_n)$. Let $A_i = \{s_1, \ldots, s_i\}$ for $1 \leq i \leq n$.
2. Define $A_0 = \emptyset$ and let $x'(s_i) = f(A_i) - f(A_{i-1})$, for $1 \leq i \leq n$.

Note that the greedy algorithm is a strongly polynomial-time algorithm.

To show that the greedy algorithm above is correct, consider the dual of maximizing $w \cdot x$:

$$\min \sum_{U \subseteq S} y(U) f(U) \tag{2}$$

$$\sum_{U \ni s_i} y(U) = w(s_i)$$

$$y \geq \mathbf{0}.$$

Define the dual solution: $y'(A_n) = y'(S) = w(s_n)$, $y'(A_i) = w(s_i) - w(s_{i+1})$ for $1 \leq i \leq n-1$, and $y'(U) = 0$ for all other $U \subseteq S$.

Exercise: Prove that x' and y' are feasible and y' satisfies complementary slackness w.r.t. x' in (1) and (2). Then it follows that the system of inequalities $\{x \in \mathbb{R}^S \mid x(U) \leq f(U),\ \forall U \subseteq S\}$ is totally dual integral (TDI), because the optimum of (2) is attained by the integral vector y' constructed above (if the optimum exists and is finite).

Theorem 4 *If $f : 2^S \to \mathbb{R}$ is a submodular function with $f(\emptyset) = 0$, the greedy algorithm (computing x') gives an optimum solution to* (1)*. Moreover, the system of inequalities $\{x \in \mathbb{R}^S \mid x(U) \leq f(U),\ \forall U \subseteq S\}$ is totally dual integral (TDI).*

Now consider the case of P_f. Note that P_f is non-empty iff $f \geq \mathbf{0}$. We note that if f is monotone and non-negative, then the solution x' produced by the greedy algorithm satisfies $x \geq \mathbf{0}$ and hence if feasible for P_f. So we obtains:

Corollary 5 *If f is a non-negative monotone submodular function on S with $f(\emptyset) = 0$ and let $w : S \to \mathbb{R}_+$, then the greedy algorithm also gives an optimum solution x' to* $\max\{w \cdot x \mid x \in P_f\}$*. Moreover, the system of inequalities $\{x \in \mathbb{R}_+^S \mid x(U) \leq f(U),\ \forall U \subseteq S\}$ is TDI.*

Therefore, from Theorem 4 and Corollary 5, for any integer-valued submodular function f, EP_f is an integer polyhedron, and if in addition f is non-negative and monotone, P_f is also an integer polyhedron.

One-to-one correspondence between f and EP_f. Theorem 4 also implies f can be recovered from EP_f. In other words, for any extended polymatroid P, there is a unique submodular function f satisfying $f(\emptyset) = 0$, with which P is associated with (*i.e.*, $EP_f = P$), since:

Claim 6 *Let f be a submodular function on S with $f(\emptyset) = 0$. Then $f(U) = \max\{x(U) \mid x \in EP_f\}$ for each $U \subseteq S$.*

Proof: Let $\alpha = \max\{x(U) \mid x \in EP_f\}$. $\alpha \leq f(U)$, because $x \in EP_f$. To prove $\alpha \geq f(U)$, in (1), define $w(s_i) = 1$ iff $s_i \in U$ and $w(s_i) = 0$ otherwise, consider the greedy algorithm producing x':

W.l.o.g., we can assume after Step 1 in the greedy algorithm, $U = \{s_1, s_2, \ldots, s_k\}$, and $w(s_i) = 1$ if $1 \leq i \leq k$ and $w(s_i) = 0$ otherwise. Define $x'(s_i) = f(A_i) - f(A_{i-1})$ where $A_i = \{s_1, \ldots, s_i\}$. As x' is feasible in (1) (exercise: $x' \in EP_f$), $w \cdot x' \leq \max\{w \cdot x \mid x \in EP_f\}$. From the definition of w, $w \cdot x = x(U)$, and from the selection of x', $w \cdot x' = f(A_1) - f(\emptyset) + f(A_2) - f(A_1) + \ldots + f(A_k) - f(A_{k-1}) = f(A_k) - f(\emptyset) = f(U)$. Therefore, $f(U) \leq \max\{x(U) \mid x \in EP_f\} = \alpha$. □

There is a similar one-to-one correspondence between non-empty polymatroids and non-negative monotone submodular functions f with $f(\emptyset) = 0$. We can also show that, for any such function f, $f(U) = \max\{x(U) \mid x \in P_f\}$ for each $U \subseteq S$.

3 Ellipsoid-based Submodular Function Minimization

Let $f : 2^S \to \mathbb{R}$ be a submodular function and assume it is given as a value oracle, *i.e.*, when given $U \subseteq S$, the oracle returns $f(U)$. Our goal is to find $\min_{U \subseteq S} f(U)$. Before discussing combinatorial algorithms for this problem, we will first describe an algorithm based on the equivalence of optimization and separation (the ellipsoid-based method) in this section.

We can assume $f(\emptyset) = 0$ (by resetting $f(U) \leftarrow f(U) - f(\emptyset)$ for all $U \subseteq S$). With the greedy algorithm introduced in Section 2, we can optimize over EP_f in polynomial time (Theorem 4). So the separation problem for EP_f is solvable in polynomial time, hence also the separation problem for $P = EP_f \cap \{x \mid x \leq \mathbf{0}\}$, and therefore also the optimization problem for P.

Fact 7 *There is a polynomial-time algorithm to separate over P, and hence to optimize over P.*

Claim 8 *If $f(\emptyset) = 0$, $\max\{x(S) \mid x \in P\} = \min_{U \subseteq S} f(U)$, where $P = EP_f \cap \{x \mid x \leq \mathbf{0}\}$.*

Proof: Define $g = f|\mathbf{0}$, and then we have $g(S) = \min_{U \subseteq S} f(U)$. Since g is submodular (from Claim 2) and $P = EP_g$ (from Claim 3), thus from Claim 6, $g(S) = \max\{x(S) \mid x \in P\}$. Therefore, we have $\max\{x(S) \mid x \in P\} = \min_{U \subseteq S} f(U)$. $\square$

Fact 7 and Claim 8 imply that we can compute the value of $\min_{U \subseteq S} f(U)$ in polynomial time. We still need an algorithm to find $U^* \subseteq S$ s.t. $f(U^*) = \min_{U \subseteq S} f(U)$.

Theorem 9 *There is a polynomial-time algorithm to minimize a submodular function f given by a value oracle.*

Proof: To complete the proof, we present an algorithm to find $U^* \subseteq S$ s.t. $f(U^*) = \min_{U \subseteq S} f(U)$.

Initially, let $\alpha = \min_{U \subseteq S} f(U)$. In each iteration:

1. We find an element $s \in S$ s.t. the minimum value of f over all subsets of $S \setminus \{s\}$ is equal to α, which implies that there exists an $U^* \subseteq S$ with $f(U^*) = \alpha$ and $s \notin U^*$.

2. So we then focus on $S \setminus \{s\}$ for finding the U^*; this algorithm proceeds with setting $S \leftarrow S \setminus \{s\}$ and repeats Step 1 for finding another such s; if such an s cannot be found in some iteration, the algorithm terminates and returns the current S as U^*. $\square$

1 Submodular Functions and Convexity

Let $f : 2^S \rightarrow \mathbb{R}$ be a submodular set function. We discuss a connection between submodular functions and convexity that was shown by Lovász [3].

Given an arbitrary (not necessarily submodular) set function $f : 2^S \rightarrow \mathbb{R}$, we can view it as assigning values to the integer vectors in the hypercube $[0,1]^n$ where $n = |S|$. That is, for each $U \subseteq S$, $f(\chi(U)) = f(U)$. We say that a function $\hat{f} : [0,1]^n \rightarrow \mathbb{R}$ is an *extension* of f if $\hat{f}(\chi(U)) = f(U)$ for all $U \subseteq S$; that is $\hat{f}$ assigns a value to each point in the hypercube and agrees with f on the characterstic vectors of the subsets of S. There are several ways to define an extension and we consider one such below.

Let $S = \{1, 2, \ldots, n\}$. Consider a vector $c = (c(1), \ldots, c(n))$ in $[0,1]^n$ and let $p_1 > p_2 > \ldots > p_k$ be the distinct values in $\{c(1), c(2), \ldots, c(n)\}$. Define $q_k = p_k$ and $q_j = p_j - p_{j+1}$ for $j = 1, \ldots, k-1$. For $1 \leq j \leq k$, we let $U_j = \{i \mid c(i) \geq p_j\}$. Define $\hat{f}$ as follows:

$$\hat{f}(c) = (1 - p_1) f(\emptyset) + \sum_{j=1}^{k} q_j f(U_j)$$

As an example, if $c = (0.75, 0.3, 0.2, 0.3, 0)$ then

$$\hat{f}(c) = 0.25 \cdot f(\emptyset) + 0.45 \cdot f(\{1\}) + 0.1 \cdot f(\{1,2,4\}) + 0.2 \cdot f(\{1,2,3,4,5\})$$

In other words c is expressed as a convex combination $\chi(\emptyset) + \sum_{j=1}^{k} q_j \chi(U_j)$ of vertices of the hypercube, and $\hat{f}(c)$ is the natural interpolation. It is typically assumed that $f(\emptyset) = 0$ (one can always shift any function to achieve this) and in this case we can drop the term $(1 - p_1) f(\emptyset)$; however, it is useful to keep in mind the implicit convex decomposition.

Lemma 1 *If f is submodular then $\hat{f}(c) = \max\{cx \mid x \in EP_f\}$.*

We leave the proof of the above as an exercise. It follows by considering the properties of the Greedy algorithm for maximizing over polymatroids that was discussed in the previous lecture.

Theorem 2 (Lovász) *A set function $f : 2^S \rightarrow \mathbb{R}$ with $f(\emptyset) = 0$ is submodular iff $\hat{f}$ is convex.*

Proof: Suppose f is submodular. Let $c_1, c_2 \in [0,1]^n$ and $t \in [0,1]$ and let $c = tc_1 + (1-t)c_2$. To show that $\hat{f}$ is convex we need to show that $\hat{f}(c) \leq \hat{f}(tc_1) + \hat{f}((1-t)c_2)$. This follows easily from Lemma 1. Let $x^* \in EP_f$ be such that $\hat{f}(c) = c \cdot x^* = tc_1 \cdot x^* + (1-t)c_2 \cdot x^*$. Then $\hat{f}(tc_1) \geq tc_1 \cdot x^*$ and $\hat{f}((1-t)c_2) \geq (1-t)c_2 \cdot x^*$ and we have the desired claim.

Now suppose $\hat{f}$ is convex. Let $A, B \subseteq S$. From the definition of $\hat{f}$ we note that $\hat{f}((\chi(A) + \chi(B))/2) = \hat{f}(\chi(A \cup B)/2) + \hat{f}(\chi(A \cap B)/2)$ (the only reason to divide by 2 is to ensure that we stay in $[0,1]^n$). On the other hand, by convexity of $\hat{f}$, $\hat{f}((\chi(A)+\chi(B))/2) \leq \hat{f}(\chi(A)/2) + \hat{f}(\chi(B)/2)$. Putting together these two facts, we have $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$, and hence f is submodular. □

Corollary 3 *If f is submodular then* $\min_{U\subseteq S} f(S) = \min_{c\in[0,1]^n} \hat{f}(c)$.

Proof: Clearly $\min_{c\in[0,1]^n} \hat{f}(c) \leq \min_{U\subseteq S} f(S)$. To see the converse, let $c^* \in [0,1]^n$ achieve the minimum of $\min_{c\in[0,1]^n} \hat{f}(c)$. Then one of the sets in the convex combination of c^* in the definition of the extension achieves a value equal to $\hat{f}(c^*)$. □

The above shows that submodular function minimization can be reduced to convex optimization problem in a natural fashion. One advantage of an extension as above is that one can use it as a relaxation in optimization problems involving submodular functions and additional constraints. For example we may want to solve $\min_{U\subseteq S} f(S)$ subject to U satisfying some additional constraints that could perhaps be modeled as $x(S) \in P$ for some convex set P. Then we could solve $\min\{\hat{f}(x) \mid x \in P\}$ as a relaxation and round the solution in some fashion. There are several examples of this in the literature.

2 Combinatorial Algorithms for Submodular Function Minimization

We saw in last lecture an algorithm for solving the submodular function minimization problem (SFM): given f as a value oracle, find $\min_{U\subseteq S} f(S)$. The algorithm was based on solving a linear program via the ellipsoid method and has a strongly polynomial running time. A question of interest is whether there is a polynomial time "combinatorial" algorithm for this problem. Although there is no clear-cut and formal definition of a combinatorial algorithm, typically it is an algorithm whose operations have some combinatorial meaning in the underlying structure of the problem. Cunningham [**?**] gave a pseudo-polynomial time algorithm for this problem in 1985. It is only in 2000 that Schrijver [**?**] and independently Iwata, Fleischer and Fujishige gave polynomial time combinatorial algorithms for SFM. There have been several papers that followed these two; we mention the algorithm(s) of Iwata and Orlin [2] that have perhaps the shortest proofs. All the algorithms follow the basic outline of Cunningham's approach which was orginally developed by him for the special case of SFM that arises in the separation oracle for the matroid polytope.

Two excellent articles on this subject by Fleischer [1] and Toshev [5]. We set up the min-max result on which the algorithms are based and reader should refer to [1, 5, 4, 2] for more details.

2.1 Base Polytope and Extreme Bases via Linear Orders

Recall that $EP_f = \{x \in \mathbb{R}^S \mid x(U) \leq f(U) \forall U \subseteq S\}$. We obtain the *base polytope* by adding the constraint $x(S) = f(S)$ to EP_f.

$$B_f = \{x \in \mathbb{R}^S \mid x(U) \leq f(U)\ \forall U \subseteq S,\ x(S) = f(S)\}.$$

A vector x in B_f is called a *base vector* or simply a base of EP_f (or of f). A *base vector* of f is a base vector of EP_f. Note that B_f is a face of EP_f. B_f is a polytope, since $f(\{s\}) \geq x_s = x(S) - x(S \setminus \{s\}) \geq f(S) - f(S \setminus \{s\})$ for each $s \in S$.

An extreme point of the base polytope is called an extreme base. What are the extreme bases? We note that the greedy algorithm for $\max\{wx \mid x \in EP_f\}$ generates a base whenever $w \geq 0$ (if $w(v) < 0$ for some v then the optimum value is unbounded). In fact the greedy algorithm uses w only to sort the elements and then ignores the weights. Thus, any two weight vectors that result in the same sorted order give rise to the same base. We set up notation for this. Let $L = v_1, v_2, \ldots, v_n$ be a total order on S, in other words a permutation of S. We say $u \prec_L v$ if u comes before v in L; we use $\preceq_L$ if u, v need not be distinct. Let $L(v)$ denote $\{u \in S \mid u \preceq_L v\}$. Given a total order L the greedy algorithm produces a base vector b_L where for each $v \in S$,

$$b_L(v) = f(L(v)) - f(L(v) \setminus \{v\}).$$

Lemma 4 *For each linear order L on S the vector b_L is an extreme base. Moreover, each extreme base x there is a linear order L (could be more than one) such that $x = b_L$.*

2.2 A Min-Max Theorem

Recall that the linear programming based algorithm for SFM was based on the following theorem of Edmonds.

Theorem 5 *For a submodular function $f : 2^S \to \mathbb{R}$ with $f(\emptyset) = 0$,*

$$\min_{U \subseteq S} f(U) = \max\{x(S) \mid x \in EP_f, x \leq 0\}.$$

A related theorem that one can prove from the above is the following. For a vector $z \in \mathbb{R}^S$ and $U \subseteq S$ we define $z^-(U)$ as $\sum_{v \in U : z(v) < 0} z(v)$. Alternatively, $z^-(v) = \min\{0, z(v)\}$.

Theorem 6 *For a submodular function $f : 2^S \to \mathbb{R}$ with $f(\emptyset) = 0$,*

$$\min_{U \subseteq S} f(U) = \max\{x^-(S) \mid x \in B_f\}.$$

We give direct proof of this which underlies the algorithmic aspects.

Proof: For any $x \in \mathbb{R}^S$ and $U \subseteq S$ we have $x^-(S) \leq x(U)$. If in addition $x \in B_f$ then $x^-(S) \leq x(U) \leq f(U)$. Since this holds for any $U \subseteq S$ we have that $\min_{U \subseteq S} f(U) \geq \max\{x^-(S) \mid x \in B_f\}$.

For the converse direction, let x be an optimum solution to $\max\{x^-(S) \mid x \in B_f\}$. Let $N = \{u \in S \mid x(u) < 0\}$ and $P = \{u \in S \mid x(u) > 0\}$. We observe that for any $v \in S \setminus (N \cup P)$, $x(v) = 0$. We say that a set U is tight with respect to x if $x(U) = f(U)$. Recall that tight sets uncross, in other words the set of all tight sets are closed under intersection and union.

Claim 7 *For any $u \in N$ and $v \in P$, there exists a tight set Y_{uv} where $u \in Y_{uv}$ and $v \notin Y_{uv}$.*

Assuming the claim above we finish the proof as follows. For $u \in N$, let $Y_u = \cap_{v \in P} Y_{uv}$. We note that Y_u is tight and $Y_u \cap P = \emptyset$. Let $Z = \cup_{u \in N} Y_u$. The set Z is tight and $N \subseteq Z$ and $Z \cap P = \emptyset$. Therefore, $x^-(S) = x(Z) = f(Z)$ and we are done.

Now we prove the claim by contradiction. Suppose it is not true. Then there is a $u \in N$ and $v \in P$ such that for all A where $u \in A$ and $v \notin A$ we have $x(A) < f(A)$. Let $\epsilon = \min\{f(A) - x(A) \mid u \in A, v \notin A\}$; we have $\epsilon > 0$. Let $\epsilon' = \min(\epsilon, |x(u)|, |x(v)|)$. We obtain a new vector $x' \in B_f$ as $x' = x + \epsilon'(\chi(u) - \chi(v))$, that is we add ϵ' to $x(u)$ and subtract ϵ' from $x(v)$. The new vector x' contradicts the optimality of x since $x'^-(S) > x^-(S)$. □

The above proof suggests the following definition.

Definition 8 *Given a vector $x \in B_f$ and $u, v \in S$, the* exchange capacity *of u, v with respect to x, denoted by $\alpha(x, v, u)$, is* $\min\{f(A) - x(A) \mid u \in A, v \notin A\}$.

A corollary that follows from the proof of Theorem 6.

Corollary 9 *A vector $x \in B_f$ is optimum for* $\max\{x^-(S) \mid x \in B_f\}$ *iff* $\alpha(x, v, u) = 0$ *for all $u \in N$ and $v \in P$ where $N = \{u \in S \mid x(u) < 0\}$ and $P = \{v \in S \mid x(v) > 0\}$.*

We remark that $\max\{x^-(S) \mid x \in B_f\}$ is not a linear optimization problem. The function $x^-(S)$ is a concave function (why?) and in particular the optimum solution need not be a vertex (in other words an extreme base) of B_f. See [1] for an illustrative example in two dimensions.

2.3 Towards an Algorithm

From Corollary 9 one imagines an algorithm that starts with arbitrary $x \in B_f$ (we can pick some arbitrary linear order L on S and set $x = b_L$) and improve $x^-(S)$ by finding a pair u, v with $u \in N$ and $v \in P$ with non-negative exchange capacity and improving as suggested in the proof of Theorem 6. However, this depends on our ability to compute $\alpha(x, v, u)$ and one sees from the definition that this is another submodular function minimization problem!

A second technical difficulty, as we mentioned earlier, is that the set of optimum solutions to $\max\{x^-(S) \mid x \in B_f\}$ may not contain an extreme base.

The general approach to overcoming these problems follows the work of Cunningham. Given $x \in B_f$ we express x as a convex combination of extreme bases (vertices of B_f); in fact, using Lemma 4, it is convenient to use linear orders as the implicit representation for an extreme base. Then we write $x = \sum_{L \in \Lambda} \lambda_L b_L$ where Λ is a collection of linear orders. By Caratheodary's theorem, Λ can be chosen such that $|\Lambda| \leq |S|$ since the dimension of B_f is $|S| - 1$. Although computing the exchange capacities with respect to an arbitrary $x \in B_f$ is difficult, if x is an extreme base b_L for a linear order L, then we see below that several natural exchanges can be efficiently computed. The goal would then to obtain exchanges for $x \in B_f$ by using exchanges for the linear orders in the convex combination for x given by Λ. Different algorithms take different approaches for this. See [1, 5], in particular [5] for detailed descriptions including intuition.

References

[1] L. Fleischer. Recent Progress in Submodular Function Minimization. OPTIMA: Mathematical Programming Society Newsletter , September 2000, no.64, 1-11. Available online at `http://www.mathprog.org/Optima-Issues/optima64.pdf`.

[2] S. Iwata and J. Orlin. A Simple Combinatorial Algorithm for Submodular Function Minimization. *Proc. of ACM-SIAM SODA*, 2009.

[3] L. Lovász. Submodular functions and convexity. Mathematical programming: the state of the art, Bonn, 235–257, 1982.

[4] A. Schrijver. *Combinatorial Optimization.* Springer-Verlag Berlin Heidelberg, 2003.

[5] Alexander Toshev. Submodular Function Minimization. Manuscript, January 2010. `http://www.seas.upenn.edu/~toshev/Site/About_Me_files/wpii-2.pdf`.

CS 598CSC: Combinatorial Optimization	Lecture date: April 20, 2009
Instructor: Chandra Chekuri	Scribe: Jing Gao

1 Spanning Trees

Let $G = (V, E)$ be an undirected graph and let $c : E \to R$ be an edge-cost function. Efficient polynomial time algorithms for computing a minimum cost spanning tree (MST) are standard. Spanning trees in G are bases in the associated graphic matroid and Kruskal's algorithm for MST is the essentially the greedy algorithm for computing a minimum cost base in a matroid. From polyhedral results on matroids we obtain corresponding results for spanning trees.

The spanning tree polytope of $G = (V, E)$ is the polytope formed by the convex hull of the characteristic vectors of spanning trees of G, and is determined by the following inequalities. We have a variable $x(e)$ for each $e \in E$ and for a set $U \subseteq V$, $E[U]$ is the set of edges with both end points in U.

$$x(E) = n - 1$$

$$x(E[U]) \leq |U| - 1 \quad U \subseteq V$$

$$x \geq 0$$

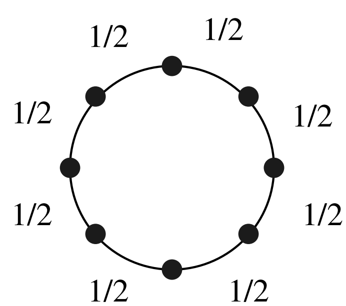
If we drop the constraint $x(E) = n - 1$, then we obtain the convex hull of the characterstic vectors of forests in G, called the forest polytope of G; note that forests are the independent sets in the graphic matroid of G.

A natural cut-based formulation for spanning trees is the following:

$$x(\delta(U)) \geq 1 \quad \forall \emptyset \subset U \subset S$$

$$x \geq 0$$

It is easy to check that every spanning tree satisfies the above constraints, but the following example shows that the constraints do not determine the spanning tree polytope. Take G to the n-cycle C_n and set $x(e) = \frac{1}{2}$ on each edge; satisfies the cut-constraints but cannot be written as a convex combination of spanning trees. In fact, it does not even satisfy the constraint that $x(E) = n - 1$.



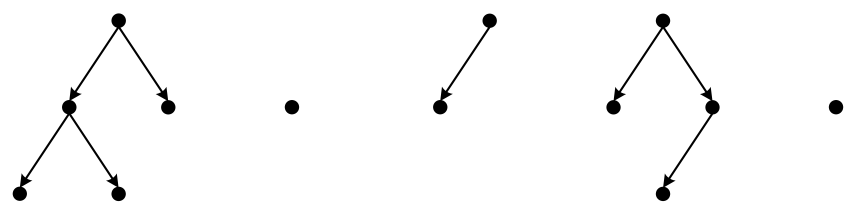
Exercise 1 *Show that even if we add the constraint $x(E) = n - 1$ to the cut-constraints, it still does not determine the spanning tree polytope.*

We have seen Tutte-Nash-Williams Theorem on maximum number of edge-disjoint spanning trees. Matroid union theorem gives polynomial-time algorithms to find a maximum number of edge-disjoint spanning trees in a given graph. We have also seen Nash-Williams forest cover theorem and again matroid union algorithm can be used to obtain the minimum number of forests that cover E.

2 Arborescences and Branchings

Let $D = (V, A)$ be a directed graph. Recall that a branching is a set of edges $A' \subseteq A$ such that

1. $\delta^{-1}_{A'}(v) \leq 1 \quad \forall v \in V$, i.e., at most one edge in A' enters any node v;
2. A' when viewed as undirected edges induces a forest on V.



An arborescence is a branching that has in-degree 1 for all nodes except one, called the root. An arborescence has a directed path from the root to each node $v \in V$.

Proposition 2 *Let $D = (V, A)$ be a directed graph and let $r \in V$. If r can reach every node $v \in V$, then there is an arborescence in D rooted at r. If G is strongly connected, then for every $v \in V$, there is an arborescence rooted at v.*

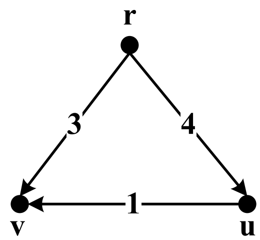
Branchings and Matroid Intersection: We saw earlier that branchings in a directed graph $D = (V, A)$ can be viewed as the common independent sets in the intersection of two matroids on A. Let $M_1 = (A, \mathcal{I}_1)$ where $\mathcal{I}_1 = \{A' \subseteq A \mid |A' \cap \delta^{-1}(v)| \leq 1 \quad \forall v \in V\}$. M_1 is a partition matroid. $M_2 = (A, \mathcal{I}_2)$ where $\mathcal{I}_2 = \{A' \subseteq A \mid A'$ when viewed as undirected edges induces a forest on $V\}$. M_2 is a graphic matroid. Thus, one easily sees that $\mathcal{I}_1 \cap \mathcal{I}_2$ is precisely the set of branchings. Moreover, for a fixed r, if we modify M_1 such that $M_1 = \{A' \subseteq A \mid |A' \cap \delta^{-1}(v)| \leq 1 \quad \forall v \in V \backslash \{r\}$ and $|A' \cap \delta^{-1}(r)| = 0\}$, then the set of arborescences rooted at r are precisely the common *bases* of $\mathcal{I}_1$ and $\mathcal{I}_2$.

Using matroid intersection results, one can solve the following problems in polynomial time:

- given $D = (V, A)$ and $w : A \to R$, find a maximum weight branching;
- given D and $c : A \to R$, find a min-cost arborescence rooted at r;
- given D and $c : A \to R$, find a max-cost arborescence rooted at r;

Polyhedral results also follow from matroid intersection. However, one can obtain direct and simple algorithms, and also polyhedral results, for arborescences. We explain them below. We first address algorithms for the optimization problems discussed above.

Combinatorial Algorithms: Let $D = (V, A)$ and $c : A \to R_+$ be a non-negative cost function on the the arcs. We wish to find a min-cost arborescence rooted at given node $r \in V$. We observe that a greedy algorithm similar to Prim's algorithm for computing an MST does not work.



For the above example, greedy method will pick (r, v) and then has to pick (r, u) for total cost of 7. However, optimal arborescence is $\{(r, u), (u, v)\}$ of cost 5.

Algorithm for min-cost r-aroborescence:

1. Let $A_0 = \{a \in A \mid c(a) = 0\}$ be the set of zero-cost arcs. If there is an r-arborescence in A_0 (of cost 0), output it as the min-cost arborescence.

2. Else, let $S_1, \ldots, S_k$ be the vertex sets of the strong connected components of $D[A_0]$. Let $\alpha_i = \min_{a \in \delta^{-1}(S_i)} c(a)$, that is α_i is the cost of the min-cost edge entering S_i.

   for $i = 1$ to k do

   for each $a \in \delta^{-1}(S_i)$

   $c'(a) = c(a) - \alpha_i$

3. Recursively compute a min-cost arborescence in $D = (V, A)$ with edge-cost function c'. Output the solution of recursive call.

First, we argue that the algorithm terminates in polynomial time. If A_0 does not contain an r-arborescence then at least one S_i has all incoming edges of strictly positive cost (why?) and hence in step 2, at least one additional arc has its cost reduced to zero; thus the size of A_0 increases and hence at most $O(m)$ recursive calls suffice. In fact, if we shrink each S_i to a single vertex, one can show that the number of vertices reduces by at least one and hence $O(n)$ recursive calls suffice. Since strong connected components can be found in $O(m)$ time, this leads to an $O(mn)$ running time.

Now we argue correctness. It is easy to see that step 1 is correct. To argue correctness of step 3, we have the following lemma.

Lemma 3 *Let $S_1, \ldots, S_k$ be vertex sets of the strong connected components of $D[A_0]$. Then there exists a min-cost arborescence A^* in D s.t. $|\delta^{-1}(S_i) \cap A^*| = 1$.*

Proof: Say A^* is an optimal arborescence and $|A^* \cap \delta^{-1}(S_i)| \geq 2$. Let $a = \arg\min_{a' \in A^* \cap \delta^{-1}(S_i)} c(a')$ be the least cost arc entering S_i in A^*. Then let $A' = (A^* \backslash \delta^{-1}(S_i)) \cup \{a\} \cup (A_0 \cap A[S_i])$; A' is the set of arcs obtained by adding to A^* all zero-cost arcs inside S_i $(A_0 \cap A[S_i])$ and removing all arcs from A^* that enter S_i other than the least cost arc a defined above. It is easy to check that A' contains an r-arborescence and moreover its cost is no more than that of A_0. Further, $|\delta^{-1}(S_i) \cap A'| = 1$

and $|\delta^{-1}(S_j) \cap A'| = |\delta^{-1}(S_j) \cap A^*|$ for all $j \neq i$. Repeating the above process for each S_i gives the desired claim. $\Box$

This leads to the following theorem.

Theorem 4 *There is an $O(nm)$ time algorithm to compute a min-cost r-arborescence in a directed graph with n nodes and m edges.*

There is an $O(m + n \log n)$-time algorithm to find a min-cost r-arboresence problem which is comparable to the standard MST algorithms.

Max-weight arborescences and Branchings. Since any arborescence has exactly $n-1$ edges, one can solve the max-weight arborescence by negating weights, adding a large positive number to make weights positive and then computing a min-weight arborescence.

One can use the max-weight arborescence algorithm to compute a max- weight branching. Note that given $w : A \rightarrow R$, we can assume $w(a) \geq 0 \ \forall a$ by removing all arcs with negative weights. We note that a max-weight branching may not be maximal even when weights are positive; this is unlike the case of matroids (in particular, a max-weight forest is a spanning tree if all weights are non-negative and the input graph is connected). To solve the max weight branching problem, we add a new vertex r and connect it to each $v \in V$ with an arc (r, v) of weight 0. Now we find a max-weight arborescence rooted at r. We leave the correctness of this algorithm as an easy exercise.

2.1 Polyhedral Aspects

One can obtain polyhedral descriptions for branchings and arborescences via matroid intersection. However, some natural and direct descriptions exist.

Let $\mathcal{P}_{r-arborescence}(D) = convexhull\{\chi(B) \mid B \text{ is a } r\text{-arborescence in } D\}$.

Theorem 5 *$\mathcal{P}_{r-arborescence}(D)$ is determined by*

$$x(a) \geq 0 \quad a \in A$$

$$x(\delta^{-1}(v)) = 1 \quad v \in V \backslash \{r\}$$
$$x(\delta^{-1}(U)) \geq 1 \quad U \subseteq V \backslash \{r\}$$

Proof: We give an iterated rounding based proof to show that the following set of inequalities

$$0 \leq x(a) \leq 1 \quad a \in A$$

$$x(\delta^{-1}(U)) \geq 1 \quad U \subseteq V \backslash \{r\}$$

is the convex hull of the characteristic vectors of arc sets that contain an r-arborescence. One can easily adapt the proof to show the theorem statement. Let x be any basic feasible solution to the above system. We claim that $\exists a \in A$ s.t. $x(a) = 0$ or $x(a) = 1$. In either case, we obtain the desired proof by induction on $|A|$. If $x(a) = 0$ we consider $D[A \setminus \{a\}]$, if $x(a) = 1$, we shrink the end points of a into a single vertex and consider the resulting graph.

We now prove that $\exists a \in A$ s.t. $x(a) \in \{0, 1\}$. Assume not, then $x(a) \in (0, 1) \ \forall a \in A$. Let $\mathcal{F} = \{U \in V \backslash \{r\} \mid x(\delta^{-1}(U)) = 1\}$ be the collection of tight sets.

Claim 6 *$\mathcal{F}$ is closed under intersections and unions.*

Claim 7 *Let $\mathcal{L}$ be a maximal laminar family in $\mathcal{F}$. Then span($\{\mathcal{X}(U) \mid U \in \mathcal{L}\}$)=span($\{\mathcal{X}(U) \mid U \in \mathcal{F}\}$).*

The above claims are based on uncrossing arguments that we have seen in several contexts. We leave the formal proofs as an exercise.

Since $\mathcal{L}$ is a laminar family on $V\backslash\{r\}$, we have

$$|\mathcal{L}| \leq 2(|V| - 1) - 1 \leq 2|V| - 3$$

Since $x(\delta^{-1}(v)) \geq 1$ for each $v \in V \setminus \{r\}$ and $x(a) \in (0, 1)$ for all $a \in A$,

$$|\delta^{-1}(v) \cap A| \geq 2 \quad \forall v \in V \setminus \{r\}$$

This implies that $|A| \geq 2|V| - 2$. However, x is a basic feasible solution and $\mathcal{L}$ determines x, and thus $|\mathcal{L}| = |A|$, a contradiction. $\square$

In fact, one can show that the system of inequalities is TDI and this implies a min-max result as well. See [1] for more details.

3 Arc-Disjoint Arborescences

A beautiful theorem of Edmonds is the following.

Theorem 8 *Let $D = (V, A)$ be a digraph and let $r \in V$. D has k arc-disjoint r-arborescences iff for each $v \in V\backslash\{r\}$ there are k arc-disjoint paths from r to v.*

Proof: If D has k arc-disjoint r-arborescences then clearly for each $v \in V\backslash\{r\}$, there are k arc-disjoint $r \to v$ paths in D, one in each of the arboresences.

We prove the converse via a proof given by Lovász using induction on k (proof adapted from [2]). Let $\mathcal{C} = \{U \subset V \mid r \in U\}$ be the collection of all proper subsets of V that contain r. Note that the condition that there are k-arc-disjoint paths from r to each v is equivalent to, by Menger's theorem,

$$|\delta^+(U)| \geq k \quad \forall U \in \mathcal{C}.$$

The idea is to start with the above condition and find an r-arborescence A_1 s.t.

$$|\delta^+(U)\backslash A_1| \geq k - 1 \qquad \forall U \in \mathcal{C}.$$

Then, by induction, we will be done. We obtain A_1 by growing an r-arborescence from r as follows. We start with $A_1 = \emptyset$ and $S = \{r\}$; S is the set of vertices reachable from r via arcs in the current set of arcs A_1. We maintain the property that $|\delta^+(U)\backslash A_1| \geq k - 1 \ \forall U \subset S, r \in U$. If we reach $S = V$, we are done.

The goal is to find an arc in $\delta^+(S)$ to add to A_1 and grow S. Call a set $X \subset V$ *critical/dangerous* if

- $X \in \mathcal{C}$ and
- $X \cup S \neq V$ and
- $|\delta^+(X) \setminus A_1| = k - 1$.

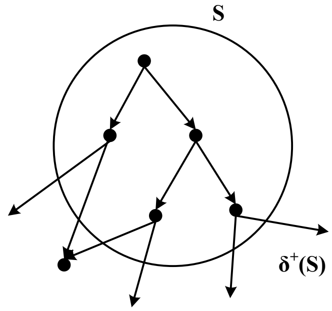


Figure 1: S and $\delta^+(S)$

We observe that if there are no critical sets, any arc $a \in \delta^+(S)$ can be used to augment A_1. If X is critical, then we cannot pick any unused arcs from $\delta^+(X)$ to grow A_1. The goal is to show that there always exists an arc $a \in \delta^+(S)$ such that a does not cross any critical set. We claim the following uncrossing property for critical sets.

Claim 9 *If X, Y are critical and $X \cup Y \neq V$, then $X \cap Y$ and $X \cup Y$ are critical.*

Proof: Let $G' = G[A \backslash A_1]$, then $|\delta^+_{G'}(X)| = k-1$ and $|\delta_{G'}(Y)| = k-1$. We have, by submodularity of the cut function $|\delta^+_G()|$,

$$|\delta_{G'}(X)| + |\delta_{G'}(Y)| \geq |\delta_{G'}(X \cup Y)| + |\delta_{G'}(X \cap Y)|.$$

Since $r \in X \cap Y$ and $r \in X \cup Y$ and $X \cup Y \neq V$, we have that $|\delta_{G'}(X \cap Y)| \geq k-1$ and $|\delta_{G'}(X \cup Y)| \geq k-1$. Since X, Y are critical,

$$k - 1 + k - 1 \geq |\delta_{G'}(X \cap Y)| + |\delta_{G'}(X \cup Y)| \Rightarrow \delta_{G'}(X \cap Y) = \delta_{G'}(X \cup Y) = k - 1.$$

□

Let X be a inclusion-wise maximal critical set.

Claim 10 *There exists an arc $(u, v) \in A$ s.t. $u \in S \setminus X$ and $v \in V \setminus (S \cup X)$.*

Proof: Note that $A_1 \cap \delta^+(S) = \emptyset$ since S, by defintion, is a set of reachable nodes in A_1. Since $|\delta^+(S \cup X)| \geq k$ (by assumption) and $|\delta^+_{G-A_1}(X)| = k-1$ (since X is critical), we have an arc as desired. □

Now let $A_1' = A_1 + (u, v)$. The claim is that for all $U \in \mathcal{C}$, $|\delta^+(U) \setminus A_1'| \geq k-1$. Suppose not. Then let Y be such that $|\delta^+(Y) \backslash A_1'| < k-1$ but this implies that $|\delta^+(Y) \backslash A_1| = k-1$, that is Y is

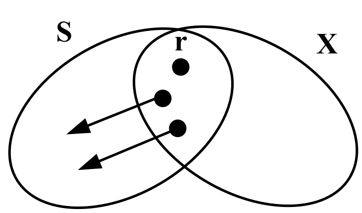


Figure 2: All arcs from $A_1 \cap \delta^+(S)$ go from X to S

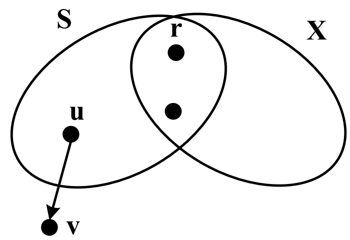


Figure 3: Claim 10

critical and $(u, v) \in \delta^+(Y)$. But consider Y, X both critical and $Y \cup X \neq V$ since $v \notin Y$, $v \notin X$. Therefore $X \cup Y$ is critical, which contradicts maximality of X. $\square$

We note that the above theorem shows the integer decomposition property for the arboresence polytope discussed earlier. The proof can be converted into a polynomial time algorithm to find a maximum number of arc-disjoint r-arborescences in a given digraph. First we let k be the $\min_{v \in V \setminus \{r\}} \lambda_D(r, v)$ where $\lambda_D(r, v)$ is the arc-connectivity between r and v. The theorem guarantees k r-aroboresences. In the above proof, the main issue is to find in each iteration an arc to augment A_1 with. We note that given an arc a, we can check if $A_1 + a$ satisfies the invariant by checking the min-cut value from r to each node v, in the graph $D' = D[A \setminus (A_1 + a)]$. The proof guarantees the existence of an arc a that can be used to augment A_1 and hence one of the m arcs in D will succeed. It is easy to see that this leads to a polynomial time algorithm. There is also a polynomial time algorithm for the capacitated case. See [1] for details.

Edmonds derived the arc-disjoint r-arborescences theorem from a more general theorem on disjoint branchings. We refer the reader to [1].

References

[1] A. Schrijver. *Combinatorial Optimization.* Springer-Verlag Berlin Heidelberg, 2003.

[2] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms.* Springer-Verlag, Fourth Edition, 2008.

CS 598CSC: Approximation Algorithms **Lecture date: April 22, 2009**
Instructor: Chandra Chekuri **Scribe: Zhenhui Li**

We describe two well known theorems in combinatorial optimization. We prove the theorems using submodular flows later.

1 Graph Orientation

Definition 1 *Let $G = (V, E)$ be an undirected graph. For $u, v \in V$, we denote by $\lambda_G(u, v)$ the edge-connectivity between u and v in G, that is, the maximum number of edge-disjoint paths between u and v. Similarly for a directed graph $D = (V, A)$, $\lambda_D(u, v)$ is the maximum number of arc-disjoint paths from u to v.*

Note that for an undirected graph G, $\lambda_G(u, v) = \lambda_G(v, u)$ but it may not be the case that $\lambda_D(u, v) = \lambda_D(v, u)$ in a directed graph D.

Definition 2 *G is k-edge-connected if $\lambda_G(u, v) \geq k$ $\forall u, v \in V$. Similarly, D is k-arc-connected if $\lambda_D(u, v) \geq k$ $\forall u, v \in V$.*

Proposition 1 *G is k-edge-connected iff $|\delta(S)| \geq k$ $\forall S \subset V$. D is k-arc-connected iff $|\delta^+(S)| \geq k$ $\forall S \subset V$.*

Proof: By Menger's theorem. □

Definition 3 *$D = (V, A)$ is an orientation of $G = (V, E)$ if D is obtained from G by orienting each edge $uv \in E$ as an arc (u, v) or (v, u).*

Theorem 2 (Robbins 1939) *G can be oriented to obtain a strongly-connected directed graph iff G is 2-edge-connected.*

Proof: "⇒" Suppose $D = (V, A)$ is a strongly connected graph obtained as an orientation of $G = (V, E)$. Then, since $\forall S \subset V$, $|\delta^+_D(S)| \geq 1$ and $|\delta^-_D(S)| \geq 1$, we have $|\delta_G(S)| \geq 2$. Therefore, G is 2-edge-connected.

"⇐" Let G be a 2-edge-connected graph. Then G has an ear-decomposition. In other words, G is either a cycle C or G is obtained from a 2-edge-connected graph G' by adding an ear P (a path) connecting two not-necessarily distinct vertices $u, v \in V$.

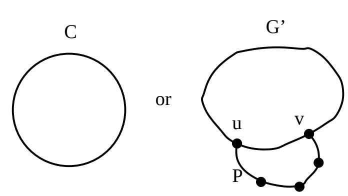
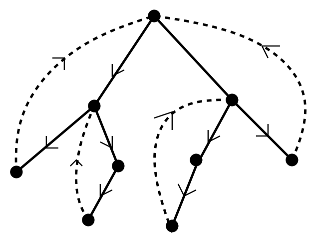


Figure 1: G is either a cycle C or is G' plus an ear P.

If $G = C$, orient it to obtain a directed cycle which is strongly-connected. Otherwise, inductively, G' has an orientation that is strongly-connected. Extend the orientation of G' to G by orienting

P from u to v (or v to u). It is easy to check that this orientation results in strongly-connected graph. $\square$

An alternative proof is as follows. Do a depth-first-search (DFS) of G starting at some node r. One obtains a DFS tree T. Orient all edges of T away from r to obtain an arborescence. Every other edge is a back-edge, that is if $uv \in E(G)\backslash E(T)$, then, either u is the ancestor of v in T or v is an ancestor of u in T. Orient uv from the descendant to the ancestor. We leave it as an exercise to argue that this is a strongly-connected orientation of G iff G is 2-edge-connected. Note that this is an easy linear time algorithm to obtain the orientation.



dashed edges are back edges

Figure 2: Orientation of a 2-edge-connected graph via a DFS tree.

Nash-Williams proved the following non-trivial extension.

Theorem 3 (Nash-Williams) *If G is $2k$-edge-connected, then it has an orientation that is k-arc-connected.*

In fact, he proved the following deep result, of which the above is a corollary.

Theorem 4 (Nash-Williams) *G has an orientation D in which $\lambda_D(u,v) \geq \lfloor \lambda_G(u,v)/2 \rfloor$ for all $u, v \in V$.*

The proof of the above theorem is difficult - see [1]. We will prove the easier version using submodular flows later.

2 Directed Cuts and Lucchesi-Younger Theorem

Definition 4 *Let $D = (V, A)$ be a directed graph. We say that $C \subset A$ is a directed cut if $\exists S \subset V$ such that $\delta^+(S) = \emptyset$ and $C = \delta^-(S)$.*

If D has a directed cut then D is *not* strongly-connected.

Definition 5 *A* dijoin *(also called a directed cut cover) in $D = (V, A)$ is a set of arcs in A that intersect each directed cut of D.*

It is not difficult to see that the following are equivalent:

- $B \subseteq A$ is a dijoin.

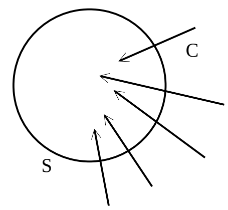


Figure 3: A directed cut $C = \delta^-(S)$.

- shrinking each arc in B results in a strongly-connected graph.
- adding all reverse arcs of B to D results in a strongly-connected graph.

Given $B \subseteq A$, it is therefore, easy to check if B is a dijoin; simply add the reverse arcs of B to D and check if the resulting digraph is strongly connected or not.

Definition 6 *A digraph D is weakly-connected if the underlying undirected graph is connected.*

Theorem 5 (Lucchesi-Younger) *Let $D = (V, A)$ be a weakly-connected digraph. Then the minimum size of a dijoin is equal to the maximum number of disjoint directed cuts.*

A dijoin intersects every directed cut so its size is at least the the maximum number of disjoint directe cuts. The above theorem is yet another example of a min-max result. We will prove this later using submodular flows. One can derive easily a weighted version of the theorem.

Corollary 6 *Let $D = (V, A)$ be a digraph with $\ell : A \to \mathbb{Z}_+$. Then the minimum length of a dijoin is equal to the maximum number of directed cuts such that each arc a is in at most $\ell(a)$ of them (in other words a maximum packing of directed cuts in ℓ).*

Proof: If $\ell(a) = 0$, contract it. Otherwise replace a by a path of length $\ell(a)$. Now apply the Lucchesi-Younger theorem to the modified graph. $\Box$

As one expects, a min-max result also leads to a polynomial time algorithm to compute a minimum weight dijoin and a maximum packing of directed cuts.

Woodall conjectured the following, which is still open. Some special cases have been solved [1].

Conjecture 1 (Woodall) *For every directed graph, the minimum size of a directed cut equals to the maximum number of disjoint dijoins.*

We describe an implication of Lucchesi-Younger theorem.

Definition 7 *Given a directed graph $D = (V, A)$, $A' \subseteq A$ is called a feedback arc set if $D[A \setminus A']$ is acyclic, that is, A' intersects each directed cycle of D.*

Computing a minimum cardinality feedback arc set is NP-hard. Now suppose D is a plane directed graph (i.e., a directed graph that is embedded in the plane). Then one defines its dual graph D^* as follows. For each arc (w, x) of D, we have a dual arc $(y, z) \in D^*$ that crosses (w, x) from "left" to "right". See example below.

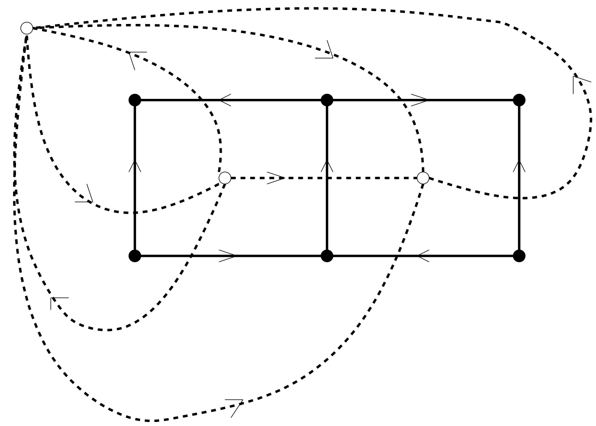


Figure 4: A planar digraph and its dual.

Proposition 7 *The directed cycles of D correspond to directed cuts in D^* and vice versa.*

Thus, a feedback arc set of D corresponds to a dijoin in D^*. Via Lucchesi-Younger theorem, we have the following corollary.

Corollary 8 *For a planar directed graph, the minimum size of a feedback arc set is equal to the maximum number of arc-disjoint directed cycles.*

Using the algorithm to compute a minimum weight dijoin, we can compute a minimum weight feedback arc set of a planar digraph in polynomial time.

3 Polymatroid Intersection

Recall the definition of total dual integrality of a system of inequalities.

Definition 8 *A rational system of inequalities $Ax \leq b$ is TDI if for all integral c, $\min\{yb \mid y \geq 0, yA = c\}$ is attained by an integral vector y^* whenever the optimum exists and is finite.*

Definition 9 *A rational system of inequalities $Ax \leq b$ is box-TDI if the system $d \leq x \leq c$, $Ax \leq b$ is TDI for each $d, c \in \mathcal{R}^n$.*

In particular, we have the following. If $Ax \leq b$ is box-TDI, then the polyhedron $\{x \mid Ax \leq b, d \leq \ell \leq u\}$ is an integer polyhedron whenever b, ℓ, u are integer vectors.

Recall that if $f : 2^S \to \mathcal{R}$ is a submodular function, EP_f is the extended polymatroid defined as

$$\{x \in \mathcal{R}^S | x(U) \leq f(U), U \subseteq S\}$$

We showed that the system of inequalities $x(U) \leq f(U), U \subseteq S$ is TDI. In fact, one can show that the system is also box-TDI. Polymatroids generalize matroids. One can also consider polymatroid intersection which generalizes matroid intersection.

Let f_1, f_2 be two submodular functions on S. Then the polyhedron $EP_{f_1} \cap EP_{f_2}$ described by

$$\begin{aligned} x(U) &\le f_1(U) \qquad U \subseteq S \\ x(U) &\le f_2(U) \qquad U \subseteq S \end{aligned}$$

is an integer polyhedron whenever f_1 and f_2 are integer valued. We sketch a proof of the following theorem.

Theorem 9 (Edmonds) *Let f_1, f_2 be two submodular set functions on the ground set S. The system of inequalities*

$$\begin{aligned} x(U) &\le f_1(U) \qquad U \subseteq S \\ x(U) &\le f_2(U) \qquad U \subseteq S \end{aligned}$$

is box-TDI.

Proof: (Sketch) The proof is similar to that of matroid intersection. Consider primal-dual pair below

$$\begin{aligned} &\max wx \\ x(U) &\le f_1(U) \qquad U \subseteq S \\ x(U) &\le f_2(U) \qquad U \subseteq S \\ &\ell \le x \le u \end{aligned}$$

$$\begin{gathered} \min \textstyle\sum_{U \subseteq S}(f_1(U)y_1(U) + f_2(U)y_2(U)) + \sum_{a \in S} u(a)z_1(a) - \sum_{a \in S} \ell(a)z_2(a) \\ \textstyle\sum_{a \in U}(y_1(U) + y_2(U)) + z_1(a) - z_2(a) = w(a), a \in S \\ y \ge 0, z_1, z_2 \ge 0 \end{gathered}$$

Claim 10 *There exists an optimal dual solution such that $\mathcal{F}_1 = \{U \mid y_1(U) > 0\}$ and $\mathcal{F}_2 = \{U \mid y_2(U) > 0\}$ are chains.*

The proof of the above claim is similar to that in matroid intersection. Consider $\mathcal{F}_1 = \{U \mid y_1(U) > 0\}$. If it is not a chain, there exist $A, B \in \mathcal{F}_1$ such that $A \not\subset B$ and $B \not\subset A$. We change y_1 by adding ϵ to $y_1(A \cup B)$ and $y_1(A \cap B)$ and subtracting ϵ from $y_1(A)$ and $y_1(B)$. One observes that the feasibility of the solution is maintained and that the objective function can only decrease since f_1 is submodular. Thus, we can uncross repeatedly to ensure that $\mathcal{F}_1$ is a chain, similarly $\mathcal{F}_2$.

Let y_1, y_2, z_1, z_2 be an optimal dual solution such that $\mathcal{F}_1$ and $\mathcal{F}_2$ are chains. Consider $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ and the $S \times \mathcal{F}$ incidence matrix M. As we saw earlier in the proof for matroid intersection, M is TUM. We then have y_1, y_2, z_1, z_2 are determined by a system $[\; y_1 \quad y_2 \quad z_1 \quad z_2 \;][\; M \quad I \quad -I \;] = w$, where w is integer and M is TUM. Since $[\; M \quad I \quad -I \;]$ is TUM, there exists integer optimum solution. $\square$

Note that, one can separate over $EP_{f_1} \cap EP_{f_2}$ via submodular function minimization and hence one can optimize $EP_{f_1} \cap EP_{f_2}$ in polynomial time via the ellipsoid method. Strongly polynomial time algorithm can also be derived. See [1] for details.

4 Submodularity on Restricted Families of Sets

So far we have seen submodular functions on a ground set S. That is $f : 2^S \to R$ and $\forall A, B \subseteq S$,

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$$

In several applications, one needs to work with restricted families of subsets. Given a finite set S, a family of sets $\mathcal{C} \subseteq 2^S$ is

- a *lattice family* if $\forall A, B \in \mathcal{C}$, $A \cap B \in \mathcal{C}$ and $A \cup B \in \mathcal{C}$.
- an *intersecting family* if $\forall A, B \in \mathcal{C}$ and $A \cap B \neq \emptyset$, we have $A \cap B \in \mathcal{C}$ and $A \cup B \in \mathcal{C}$.
- a *crossing family* if $A, B \in \mathcal{C}$ and $A \cap B \neq \emptyset$ and $A \cup B \neq S$, we have $A \cap B \in \mathcal{C}$ and $A \cup B \in \mathcal{C}$.

For each of the above families, a function f is submodular on the family if

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

whenever $A \cap B$, $A \cup B$ are guaranteed to be in family for A, B. Function f is called intersection submodular and crossing submodular if $\mathcal{C}$ is intersecting and crossing family respectively.

We give some examples of interesting families that arise from directed graphs. Let $D = (V, A)$ be a directed graph.

Example 1 *$\mathcal{C} = 2^V \setminus \{\emptyset, V\}$ is a crossing family.*

Example 2 *Fix $s, t \in V$, $\mathcal{C} = \{U \mid s \in U, t \notin U\}$ is lattice, intersecting, and crossing family.*

Example 3 *$\mathcal{C} = \{U \subset V \mid U$ induces a directed cut i.e. $\delta^+(U) = \emptyset$ and $\emptyset \subset U \subset V\}$ is a crossing family.*

For the above example, we sketch the proof that $\mathcal{C}$ is a crossing family. If $A, B \in \mathcal{C}$ and $A \cap B \neq \emptyset$ and $A \cup B \neq V$, then by submodularity of δ^+, $|\delta^+(A \cup B)| + |\delta^+(A \cap B)| \leq |\delta^+(A)| + |\delta^+(B)|$. Therefore we have $\delta^+(A \cup B) = \emptyset$ and $\delta^+(A \cap B) = \emptyset$ and more over $A \cap B$ and $A \cup B$ are non-empty. Hence they both belong to $\mathcal{C}$ as desired.

Various polyhedra associates with submodular functions and the above special families are known to be well-behaved.

For lattice families the system

$$x(U) \quad \leq \quad f(U), U \in \mathcal{C}$$

is box-TDI. Also, the following system is also box-TDI

$$\begin{aligned} x(U) &\leq f_1(U), U \in \mathcal{C}_1 \\ x(U) &\leq f_2(U), U \in \mathcal{C}_2 \end{aligned}$$

where $\mathcal{C}_1$ and $\mathcal{C}_2$ are lattice families and f_1 and f_2 are submodular on $\mathcal{C}_\infty$ and $\mathcal{C}_2$ respectively. The above facts also hold for intersecting families and intersecting submodular functions.

For crossing family $\mathcal{C}$, the system

$$x(U) \leq f(U)$$

is not necessarily TDI. However, the system

$$\begin{aligned} x(U) &\leq f(U), U \in \mathcal{C} \\ x(S) &= k \end{aligned}$$

where $k \in R$ is box-TDI. Also, the system

$$\begin{aligned} x(U) &\leq f_1(U), U \in \mathcal{C}_1 \\ x(U) &\leq f_2(U), U \in \mathcal{C}_2 \\ x(S) &= k \end{aligned}$$

is box-TDI for crossing families $\mathcal{C}_1$ and $\mathcal{C}_2$ with f_1 and f_2 crossing supermodular on $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

Although the polyhedra are well-behaved, the separation problem for them is not easy since one needs to solve submodular function minimization over a restricted family $\mathcal{C}$. It does not suffice to have a value oracle for f on sets in $\mathcal{C}$; one needs additional information on the representation of $\mathcal{C}$. We refer the reader to [1] for more details.

References

[1] A. Schrijver. *Combinatorial Optimization.* Springer-Verlag Berlin Heidelberg, 2003.

[2] Lecture notes from Michel Goemans's class on Combinatorial Optimization. http://www-math.mit.edu/ goemans/18997-CO/co-lec18.ps

CS 598CSC: Combinatorial Optimization	**Lecture date: April 27th, 2010**
Instructor: Chandra Chekuri	**Scribe: Peixiang Zhao**

1 Submodular Flows

Network flows are a fundamental object and tool in combinatorial optimization. We have also seen submodular functions and their role in matroids, matroid intersection, polymatroids and polymatroid intersection. Edmonds and Giles developed the framework of submodular flows to find a common generalization of network flow and polymatroid intersection. Here is the model.

Definition 1 (Crossing Family) *Let $D = (V, A)$ be a directed graph and let $\mathscr{C} \subseteq 2^V$ be a family of subsets of V. $\mathscr{C}$ is called a **crossing family** if: $A, B \in \mathscr{C}$, $A \cap B \neq \emptyset$, $A \cup B \neq V \Rightarrow A \cap B \in \mathscr{C}$ and $A \cup B \in \mathscr{C}$.*

Definition 2 (Crossing Submodular) *Let $\mathscr{C}$ be a crossing family. A function $f : \mathscr{C} \to \mathbb{R}$ is called **crossing submodular** on $\mathscr{C}$ if it satisfies: $A, B \in \mathscr{C}$, $A \cap B \neq \emptyset$, $A \cup B \neq V \Rightarrow f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.*

Definition 3 (Submodular Flow) *Let $D = (V, A)$ be a digraph, $\mathscr{C}$ be a crossing family, and f be a crossing submodular function on $\mathscr{C}$. A vector $x \in \mathbb{R}^A$ is called a **submodular flow** if*

$$x(\delta^-(u)) - x(\delta^+(u)) \leq f(u), \qquad \forall u \in \mathscr{C} \tag{1}$$

Theorem 4 (Edmonds-Giles, 1977**)** *The system of inequalities shown in Equation 1 is box-TDI where $\mathscr{C}$ is a crossing family on V and f is crossing submodular on $\mathscr{C}$.*

Proof: We consider the primal-dual pair of LPs below where $w : A \to \mathbb{Z}_+$ and ℓ, u are integer vectors.

$$\begin{aligned} & \max \sum wx \\ \text{s.t.} \quad & x(\delta^-(U)) - x(\delta^+(U)) \leq f(U) \qquad U \in \mathscr{C} \\ & \ell \leq x \leq u \end{aligned}$$

and

$$\begin{aligned} & \min \sum_{U \in \mathscr{C}} f(U) y(U) + \sum_{a \in A} u(a) z_1(a) - \sum_{a \in A} \ell(a) z_2(a) \\ \text{s.t.} \quad & \sum_{U : U \in \mathscr{C}, a \in \delta^-(U)} y(U) - \sum_{U : U \in \mathscr{C}, a \in \delta^+(U)} y(U) + z_1(a) - z_2(a) = w(a) \qquad a \in A \\ & y, z_1, z_2 \geq 0 \end{aligned}$$

A family of sets $\mathcal{F} \subseteq 2^V$ is cross-free if for all $A, B \in \mathcal{F}$ the following holds:

$$A \subseteq B \text{ or } B \subseteq A \text{ or } A \cap B = \emptyset \text{ or } A \cup B = V.$$

Claim 5 *There exists an optimum solution y, z_1, z_2 such that $\mathscr{F} = \{U \in \mathscr{C} \mid y(U) > 0\}$ is cross-free.*

Proof: Suppose $\mathscr{F}$ is not cross-free. Then let $A, B \in \mathscr{F}$, such that $y(A) > 0$ and $y(B) > 0$ and $A \cap B \neq \emptyset$ and $A \cup B \neq V$. Then add $\epsilon > 0$ to $y(A \cup B)$, $y(A \cap B)$ and subtract $\epsilon > 0$ from $y(A)$ and $y(B)$. By submodularity of f, the objective function increases or remains same. We claim that alterting y in this fashion maintains dual feasibility; we leave this as an exercise.

By repeated uncrossing we can make $\mathscr{F}$ cross-free. Formally one needs to consider a potential function. For example, among all optimal solutions pick one that minimizes

$$\sum_{U \in \mathscr{C}} y(U)|U||V \backslash U|$$

□

Theorem 6 *Let $\mathscr{F}$ be a cross-free family on 2^V. Let M be an $|A| \times |\mathscr{F}|$ matrix where*

$$M_{a,U} = \begin{cases} 1 & \text{if } a \in \delta^-(U) \\ -1 & \text{if } a \in \delta^+(U) \\ 0 & \text{otherwise} \end{cases}$$

Then M is TUM.

The proof of the above theorem proceeds by showing that M is a network matrix. See Schrijver Theorem 13.21 for details [1].

By the above one sees that the non-negative components of y, z_1, z_2 are determined by $[M, I, -I]$ and integer vector w where M is TUM. From this we infers that there exists an integer optimum solution to the dual. □

Corollary 7 *The polyhedron P determined by*

$$x(\delta^-(U)) - x(\delta^+(U)) \leq f(U) \qquad U \in \mathscr{C}$$

$$l \leq x \leq u$$

is an integer polyhedron whenever f is integer valued and l, u are integer vectors.

One can show that optimality on P can be done in strongly polynomial time if one has a value oracle for f. This can be done via a reduction to polymatroid intersection. We refer to Schrijver, Chapter 60 for more details [1].

2 Applications

Submodular flows are a very general framework as they combine graphs and submodular functions. We gave several applications below.

2.1 Circulations

Given a directed graph $D = (V, A)$, $x : A \to \mathbb{R}$ is a circulation if

$$x(\delta^-(v)) - x(\delta^+(v)) = 0, \qquad \forall v \in V$$

This can be modeled as a special case of submodular flow by setting $\mathscr{C} = \{\{v\} \mid v \in V\}$ and $f = 0$. We get the inequalities

$$x(\delta^-(v)) - x(\delta^+(v)) \leq 0, \qquad v \in V.$$

One can check that the above inequalities imply that for any $\emptyset \subset U \subset V$ the inequality $x(\delta^-(U) - x(\delta^+(U) \leq 0$ holds by adding up the inequalities for each $v \in U$. Combining this with the inequality $x(\delta^-(V \setminus U) - x(\delta^+(V \setminus U) \leq 0$, we have $x(\delta^-(U) - x(\delta^+(U) = 0$ for all $\emptyset \subset U \subset V$ and in particular for each $v \in V$. The box-TDI result of submodular flow implies the basic results on circulations and flows including Hoffman's circulation theorem and the max-flow min-cut theorem.

2.2 Polymatroid Intersection

We saw earlier that the system

$$x(U) \leq f_1(U) \qquad U \subseteq S$$
$$x(U) \leq f_2(U) \qquad U \subseteq S$$

is box-TDI whenever f_1, f_2 are submodular functions on S. We can derive this from submodular flows as follows. Define S' and S'' are copies of S. Let $V = S' \uplus S''$ and define $\mathscr{C} = \{U' \mid U \subseteq S\} \cup \{S' \cup U'' \mid U \subseteq S\}$, where U' and U'' denote the sets of copies of elements of U in S' and S'' [1].

Claim 8 *$\mathscr{C}$ is a crossing family.*

Exercise 9 *Prove Claim 8.*

We further define $f : \mathscr{C} \to \mathbb{R}_+$ by

$$f(U') = f_1(U) \qquad U \subseteq S$$
$$f(V \backslash U'') = f_2(U) \qquad U \subseteq S$$
$$f(S') = \min\{f_1(S), f_2(S)\}$$

Claim 10 *f is crossing submodular on $\mathscr{C}$.*

Exercise 11 *Prove Claim 10.*

Now define $G = (V, A)$ where $A = \{(s'', s') | s \in S\}$, as shown in Figure 1. The submodular flow polyhedron is

$$x(\delta^-(Z)) - x(\delta^+(Z)) \leq f(Z) \qquad Z \in \mathscr{C}$$

If $Z = U'$ where $U \subseteq S$, then we get $x(U) \leq f_1(U)$. And if $Z = V \backslash U''$, as shown in Figure 2, then we get $x(U) \leq f_2(U), U \subseteq S$. Thus, we recover the polymatroid intersection constraints. Since the submodular flow constraint inequalities are box-TDI, it implies that the polymatroid intersection constraints are also box-TDI.

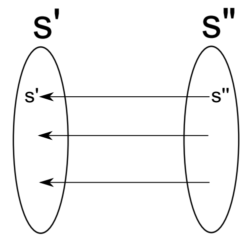


Figure 1: A Directed Graph Defined on S

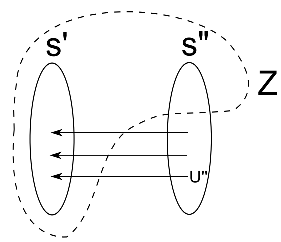


Figure 2: $Z = V \backslash U''$

2.3 Nash-Williams Graph Orientation Theorem

We stated the following theorem of Nas-Williams in the previous lecture.

Theorem 12 *If G is $2k$-edge-connected then it has an orientation D such that D is k-arc-connected.*

Frank(1980) [2] showed that Theorem 12 can be derived from submodular flows. Consider an arbitrary orientation D of G. Now if D is k-arc-connected we are done. Otherwise we consider the problem of reversing the orientation of some arcs of D such that the resulting graph is k-arc-connected. We set it up as follows.

Let $D = (V, A)$, define a variable $x(a), a \in A$ where $x(a) = 1$ if we reverse the orientation of a. Otherwise $x(a) = 0$. For a set $U \subset V$ we want k arcs coming in after applying the switch of orientation, i.e., we want

$$x(\delta^-(u)) - x(\delta^+(u)) \leq |\delta^-(u)| - k \qquad \forall\, \emptyset \subset U \subset V$$

Note that $\mathscr{C} = \{U \mid \emptyset \subset U \subset V\}$ is a crossing family and $f(U) = |\delta_D^-(u)| - k$ is crossing submodular. Hence by Edmonds-Giles theorem, the polyhedron determined by the inequalities

$$\begin{array}{rcll} x(\delta^-(u)) - x(\delta^+(u)) & \leq & |\delta^-(u)| - k & \quad \emptyset \subset U \subset V \\ x(a) & \in & [0,1] & \quad a \in A \end{array}$$

is an integer polyhedron. Moreover, the polyhedron is non-empty since $x(a) = 1/2, \forall a \in A$ satisfies all the constraints. To see this, let $\emptyset \subset U \subset V$, and let $h = \delta_D^-(U)$ and $\ell = \delta_D^+(U)$, then we have $h + \ell \geq 2k$ since G is $2k$-edge-connected. Then by setting $x(a) = 1/2, \forall a \in A$, for U we need

$$\frac{h}{2} - \frac{l}{2} \leq h - k \Rightarrow \frac{h+l}{2} \leq k$$

which is true. Thus there is an integer vector x in the polyhedron for D if G is $2k$-edge-connected. By reversing the arcs $A' = \{a | x(a) = 1\}$ in D we obtain a k-arc-connected orientation of G.

2.4 Lucchesi-Younger theorem

Theorem 13 (Lucchesi-Younger) *In any weakly-connected digraph, The size of the minimum cardinality dijoin equals the maximum number of disjoint directed cuts.*

Proof: Let $D = (V, A)$ be a directed graph and let $\mathscr{C} = \{U \mid \emptyset \subset U \subset V, |\delta^+(U)| = 0\}$, i.e., $U \in \mathscr{C}$ iff U induces a directed cut. We had seen that $\mathscr{C}$ is a crossing family. Let $f : \mathscr{C} \to \mathbb{R}$ be $f(U) = -1, \forall U \in \mathscr{C}$, clearly f is crossing submodular. Then by Edmonds-Giles theorem the following set of inequalities is TDI.

$$\begin{array}{rcll} x(\delta^-(U)) - x(\delta^+(U)) & \leq & -1 & \quad U \in \mathscr{C} \\ x & \leq & 0 & \end{array}$$

We note that $\delta^+(U) = \emptyset$ for each $U \in \mathscr{C}$. We can rewrite the above polyhedron as the one below by replacing $-x$ by x.

$$\begin{array}{rcll} x(\delta^-(U)) & \geq & 1 & \quad U \in \mathscr{C} \\ x & \geq & 0 & \end{array}$$

Note that the above is a "natural" LP relaxation for finding a set of arcs that cover all directed cuts. The above polyhedron is integral, and hence

$$\begin{array}{rcll} \min \sum_{a \in A} x(a) & & & \\ x(\delta^-(U)) & \geq & 1 & \quad U \in \mathscr{C} \\ x & \geq & 0 & \end{array}$$

gives the size of a minimum cardinality dijoin.

Consider the dual

$$\begin{array}{lll} & \max \sum_{U \in \mathscr{C}} y(U) & \\ \text{s.t.} & \sum_{U : a \in \delta^-(U), U \in \mathscr{C}} y(U) \leq 1 & \quad a \in A \\ & y \geq 0 & \end{array}$$

The dual is an integer polyhedron since the primal inequality system is TDI and the objective function is an integer vector. It is easy to see that the optimum value of the dual is a maximum packing of arc-disjoint directed cuts. Therefore by strong duality we obtain the Lucchesi-Younger theorem. $\Box$

References

[1] Lex Schrijver, "Combinatorial Optimization: Polyhedra and Efficiency, Vol. B", *Springer-Verlag*, 2003.

[2] Lecture notes from Michael Goemans's class on Combinatorial Optimization, http://www-math.mit.edu/~goemans/18997-CO/co-lec18.ps, 2004.

CS 598CSC: Combinatorial Optimization	Lecture date: April 29, 2010
Instructor: Chandra Chekuri	Scribe: Lu Su

1 Multiflows

The maxflow-mincut theorem of Ford and Fulkerson generalizes Menger's theorem and is a fundamental result in combinatorial optimization with many applications.

Theorem 1 *In a digraph $D = (V, A)$ with arc capacity function $c : A \to \mathbb{R}_+$, the maximum s-t flow value is equal to the minimum s-t capacity cut value. Moreover, if c is integer valued, then there is an integer valued maximum flow.*

In particular, the maximum number of s-t arc-disjoint paths in a digraph is equal to the minimum number of arcs whose removal disconnects s from t (Menger's theorem). When applied to undirected graphs we obtain the edge-disjoint and node-disjoint path version of Menger's Thoerem.

In many applications in networks we are interested in multiflows, also referred to as multicommodity flows. $s - t$ flows are also referred to as single-commodity flows.

A *multiflow instance* in a directed graph consists of a directed "supply" graph $D = (V, A)$ with non-negative arc capacities $c : A \to \mathbb{R}_+$ and a demand graph $H = (T, R)$ with $T \subseteq V$ called terminals, and non-negative demand requirements $d : R \to \mathbb{R}_+$. The arcs in R are referred to as nets. The demand graph can also be specified as a set of ordered pairs $(s_1, t_1), \ldots, (s_k, t_k)$ with $d_i \in \mathbb{R}_+$ denoting the demand for (s_i, t_i). This is referred to as the k-commodity flow problem.

A *multiflow instance* in an undirected graph consists of an undirected supply graph $G = (V, E)$ and an undirected demand graph $H = (T, R)$. The demand graph can be specified by a collection of unordered pairs $s_1t_1, \ldots, s_kt_k$.

Given a multiflow instance in a directed graph $D = (V, A)$ with demand graph $H = (T, R)$, a *multiflow* is a collection of flows, $f_r, r \in R$ where f_r is an s_r-t_r flow and $r = (s_r, t_r)$. A multiflow satisfies the capacity constraints of the supply graph if for each arc $a \in A$,

$$\sum_{r \in R} f_r(a) \leq c(a). \tag{1}$$

The multiflow satisfies the demands if for each $r = (s_r, t_r) \in R$, the f_r flow from s_r to t_r is at least $d(r)$.

For undirected graphs we need a bit more care. We say that $f : E \to \mathbb{R}_+$ is an $s - t$ flow if there is an orientation $D = (V, A)$ of $G = (V, E)$ such that $f' : A \to \mathbb{R}_+$ defined by the orientation and $f : E \to \mathbb{R}_+$ is an $s - t$ flow in D. Thus $f_r,\ r \in R$ where $f_r : E \to \mathbb{R}_+$ is a multiflow if each f_r is an s_r-t_r flow. It satisfies the capacity constraints if $\forall e \in E$,

$$\sum_{r \in R} f_r(e) \leq c(e). \tag{2}$$

We say a multiflow is *integral* if each of the flows is integer valued; that is $f_r(a)$ is an integer for each arc a and each $r \in R$. Similarly half-integral (i.e., each flow on an arc is an integer multiple of 1/2).

Proposition 2 *Given a multiflow instance in a directed graph, there is a polynomial time algorithm to check if there exists a multiflow that satisfies the capacities of the supply graph and the demand requirements of the demand graph.*

Proof: Can be solved by expressing the problem as a linear program. Variables $f_r(a)$ $r \in R,\ a \in A$. Write standard flow conservation constraints that ensures $f_r : A \to \mathbb{R}_+$ is a flow for each r (flow conservation at each node other than the source and destination of r). We add the following set of constraints to ensure capacity constraints of the supply graph are respected.

$$\sum_{r \in R} f_r(a) \leq c(a) \quad a \in A. \tag{3}$$

Finally, we add constraints that the value of f_r (leaving the source of r) should be at least $d(r)$. □

Proposition 3 *Given an undirected multiflow instance, there is a polynomial time algorithm to check if there is a feasible multiflow that satisfies the supply graph capacities and the demand requirements.*

Proof: We reduce it to the directed flow case as follows. Given $G = (V, E)$ obtain a digraph $D = (V, A)$ by dividing each edge e into two arcs $\overrightarrow{e}$ and $\overleftarrow{e}$. Now we have variable $f_r(a),\ a \in A,\ r \in R$, and write constraints that ensure that $f_r : A \to \mathbb{R}_+$ is a flow of value $d(r)$ from s_r to t_r where $r = s_r t_r$. The capacity constraint ensures that the total flow on both $\overrightarrow{e}$ and $\overleftarrow{e}$ is at most $c(e)$, i.e.,

$$\sum_{r \in R} (f_r(\overrightarrow{e}) + f_r(\overleftarrow{e})) \leq c(e),\ e \in E. \tag{4}$$

□

LP duality gives the following useful necessary and sufficient condition; it is some times referred to as the Japanese theorem.

Theorem 4 *A multiflow instance in directed graph is feasible iff*

$$\sum_{i=1}^{k} d_i \ell(s_i, t_i) \leq \sum_{a \in A} c(a)\ell(a) \tag{5}$$

for all length functions $\ell : A \to \mathbb{R}_+$.

Here $\ell(s_i, t_i)$ is the shortest path distance from s_i to t_i with arc lengths $\ell(a)$. For undirected graph the characterization is similar

$$\sum_{i=1}^{k} d_i \ell(s_i, t_i) \leq \sum_{e \in E} c(e)\ell(e) \tag{6}$$

for all $\ell : E \to \mathbb{R}_+$.

Proof: Consider the path formulation we prove it for undirected graphs. Let P_i be the set of $s_i t_i$ path in G. Let $f_i : P_i \to \mathbb{R}_+$ be an assignment of flow values to paths in P_i. We want feasibility of

$$\sum_{p \in P_i} f_i(p) \geq d_i \quad i = 1 \text{ to } k \tag{7}$$

$$\sum_{i=1}^{k} \sum_{p \in P_i : e \in p} f_i(p) \leq c(e) \quad e \in E \tag{8}$$

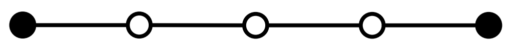
$$f_i(p) \geq 0, \quad p \in P_i, \quad 1 \leq i \leq k. \tag{9}$$

We apply Farkas lemma. Recal that $Ax \leq b,\ x \geq 0$ has a solution iff $yb \geq 0$ for each row vector $y \geq 0$ with $yA \geq 0$. We leave it as an exercise to derive the statement from Farkas lemma applied to the above system of inequalities. □

It is useful to interpret the necessity of the condition. Suppose for some $\ell : E \to \mathbb{R}_+$

$$\sum_{i=1}^{k} d_i l(s_i, t_i) > \sum_{e \in E} c(e)\ell(e) \tag{10}$$

we show that there is no feasible multiflow. For simplicity assume ℓ is integer valued. Then replace an edge e with length $\ell(e)$ by a path of length $\ell(e)$



and place capacity $c(e)$ on each edge. Suppose there is a feasible flow. For each (s_i, t_i), each flow path length is of length at least $l(s_i, t_i) \Rightarrow$ total capacity used up by flow for (s_i, t_i) is $\geq d_i \ell(s_i, t_i)$. But total capacity available is $\sum_e c(e)\ell(e)$ (after expansion). Hence if $\sum_{i=1}^{k} d_i \ell(s_i, t_i) > \sum_{e \in E} c(e)\ell(e)$, there cannot be a feasible multiflow.

To show that a multiflow instance is not feasible it is sufficient to give an appropriate arc length function that violates the necessary condition above.

2 Integer Multiflow and Disjoint Paths

When all capacities are 1 and all demands are 1 the problem of checking if there exists an integer multiflow is the same as asking if there exist arc-disjoint path (edge-disjoint path if graph is undirected) connecting the demand pairs.

The *edge-disjoint paths problem (EDP)* is the following decision problem: given supply graph $D = (V, A)$ (or $G = (V, E)$) and a demand graph $H = (T, R)$, are there arc/edge-disjoint paths connecting the pairs in R?

Theorem 5 (Fortune-Hopcroft-Wyllie 1980) *EDP in directed graphs is NP-complete even for two pairs.*

Theorem 6 *EDP in undirected graphs is NP-complete when $|R|$ is part of the input, even when $|R|$ consists of three sets of parallel edges.*

A deep, difficult and fundamental result of Robertson and Seymour is that EDP in undirected graphs is polynomial-time solvable when $|R|$ is fixed. In fact they prove that the vertex-disjoint path problem (the pairs need to be connected by vertex disjoint paths) is also tractable.

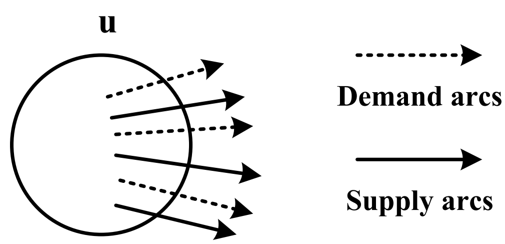
Theorem 7 (Robertson-Seymour) *The vertex-disjoint path problem is polynomial-time solvable if the number of demand pairs is a fixed constant.*

The above theorem relies on the work of Robertson and Seymour on graph minors.

3 Cut Condition—Sparsest Cuts and Flow-Cut Gaps

A necessary condition for the existence of a feasible multiflow for a given instance is the so called cut-condition. In directed graphs it is

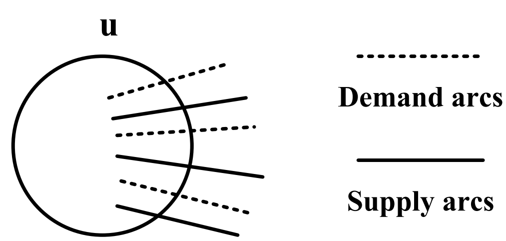
$$c(\delta_D^+(U)) \geq d(\delta_H^+(U)) \quad \forall U \subset V \tag{11}$$



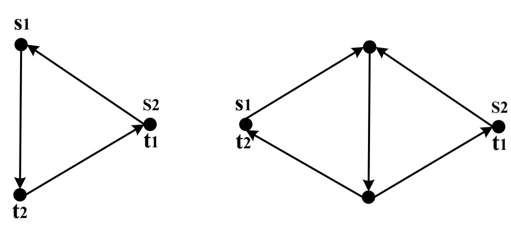
where $c(\delta_D^+(U))$ is capacity of all arcs leaving U, and $d(\delta_H^+(U))$ is the demand of all demand arcs leaving U. It is easy to see that this condition is necessary. Formally one sees that this condition is necessary by considering the length function $\ell : A \to \mathbb{R}_+$ where $\ell(a) = 1$ if $a \in \delta_D^+(U)$ and $\ell(a) = 0$.

For undirected graphs the cut condition states

$$c(\delta_G^+(U)) \geq d(\delta_H^+(U)) \quad \forall U \subset V \tag{12}$$

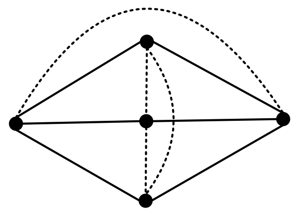


Cut condition is not sufficient in general. Consider the following examples in directed graphs



Cut condition is true for each case but no feasible multiflows exists as can be seen by considering the length function $\ell(a) = 1$ for each arc a.

For undirected graphs the following example is well known. Supply graph is $K_{2,3}$, a series-parallel graph. Again, cut-condition is satisfied but $\ell(e) = 1$ for each e shows no feasible multiflow exists.



4 Sufficiency of Cut Condition

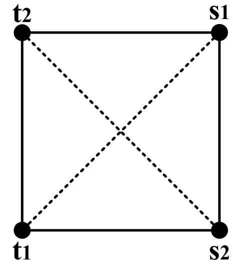
Given that the cut condition is not sufficient for feasible flow it is natural to consider cases where it is indeed sufficient. First consider directed graphs. Suppose we have demand pairs of the form $(s, t_1), (s, t_2), \ldots, (s, t_k)$, i.e., all pairs share a common source. Then it is easy to see that cut condition implies feasible multiflow by reduction to the single-commodity flow case by connecting $t_1, t_2, \ldots, t_k$ to a common sink t. Similarly if the demand pairs are of the form $(s_1, t), (s_2, t), \ldots, (s_k, t)$ with a common source.

It turns out that these are the only interesting cases for which cut condition suffices. See Theorem 70.3 in Schrijver Book.

For undirected graphs several non-trivial and interesting cases where the cut-condition is sufficient are known. We list a few below:

- Hu's 2-commodity theorem shows that if there are only two pairs s_1t_1 and s_2t_2 then cut-condition is sufficient.
- Okamura-Seymour theorem states that if G is a planar graph and T is vertex set of a single face then cut condition is sufficient. Note that the theorem implies that for capacitated ring supply graphs the cut condition is sufficient.
- Okamura's theorem generalizes Okamura-Seymour Theorem. If G is planar and there are two faces F_1 and F_2 such that each $st \in R$ has both $s,\ t$ on one of the faces then cut condition is sufficient.
- Seymour's Theorem shows that if $G + H$ is planar then cut condition is sufficient.

For all of the above cases one has the following stronger result. If $G + H$ is Eulerian then the flow is guaranteed to be integral. To see that the Eulerian condition is necessary for integral flow in each of the above cases, consider the example below where the only feasible multiflow is a half-integral.



CS 598CSC: Combinatorial Optimization	**Lecture date: May 4th, 2010**
Instructor: Chandra Chekuri	**Scribe: Dong Ye**

1 Okamura-Seymour Theorem

Theorem 1 *Let $G = (V, E)$ be a plane graph and let $H = (T, R)$ be a demand graph where T is the set of vertices of a single face of G. Then if G satisfies the cut condition for H and $G + H$ is eulerian, there is an integral multiflow for H in G.*

The proof is via induction on $2|E| - |R|$. Note that if G satisfies the cut condition for H, then $|R| \leq |E|$ (why?).

There are several "standard" induction steps and observations that are used in this and other proofs and we go over them one by one. For this purpose we assume G, H satisfy the conditions of the theorem and is a counter example with $2|E(G)| - |R|$ minimal[1].

Claim 2 *No demand edge r is parallel to a supply edge e.*

Proof: If r is parallel to e then $G - e$, $H - r$ satisfy the conditions of the theorem and by induction $H - r$ has an integral multiflow in $G - e$. We can route r via e. Thus H has an integral multiflow in G. □

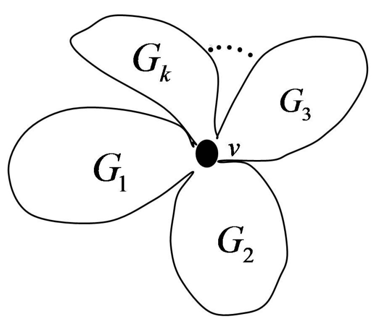
Definition 3 *A set $S \subset V$ is said to be tight if $|\delta_G(S)| = |\delta_H(S)|$.*

Claim 4 *For every demand edge $r \in R$ there is a tight cut S s.t. $r \in \delta_H(S)$.*

Proof: If r is not in any tight set, then adding two copies of r to H maintains cut condition and the Eulerian condition. By induction (note that the induction is on $2|E| - |R|$) the new instance is routable. □

Claim 5 *G is 2-node connected.*

Proof: Suppose not and let v be a cut vertex of G. Let $G_1, G_2, ..., G_k$ be the graphs obtained by combining v with the components of $G - v$.



Suppose there is a demand edge $r = (s, t)$ s.t. $s \neq v, t \neq v$ and $s \in G_i$ and $t \in G_j, i \neq j$. Then we can replace (s, t) by two edges (s, v) and (v, t). The claim is that this new instance satisfies the cut condition - we leave the formal proof as an exercise. Clearly Euler condition is maintained. The new instance is routable by induction which implies that the original instance is also routable.

If no such demand edge exits then all demand edges have both end points in G_i for some i. Then let H_i be the demand graph induced on G_i. We can verify that each G_i, H_i satisfy the cut condition and the Euler condition. By induction each H_i is routable in G_i which implies that H is routable in G. $\square$

Definition 6 *A set $\emptyset \subset S \subset V$ is central if $G[S]$ and $G[V \setminus S]$ are connected.*

Lemma 7 *Let G be a connected graph. Then G, H satisfy the cut condition if and only if the cut condition is satisfied for each central set S.*

Proof: Clearly, if G, H satisfy the cut condition for all sets then it is satisfied for the central sets. Suppose the cut condition is satisfied for all central sets but there is some non-central set S' such that $|\delta_G(S)| < |\delta_H(S)|$. Choose S' to be minimal among all such sets. We obtain a contradiction as follows. Let $S_1, S_2, \ldots, S_k$ be the connected components in $G \setminus \delta_G(S')$; since S' is not central, $k \geq 3$. Moreover each S_i is completely contained in S' or in $V \setminus S'$. We claim that some S_j violates the cut-condition, whose proof we leave as an exercise. Moreover, by minimality in the choice of S', S_j is central, contradicting the assumption. $\square$

One can prove the following corollary by a similar argument.

Corollary 8 *Let G, H satisfy the cut condition. If S' is a tight set and S' is not central, then there is some connected component S contained in S' or in $V \setminus S'$ such that S is a tight central set.*

Uncrossing:

Lemma 9 *Let G, H satisfy cut-condition, Let A, B be two tight sets such that $A \cap B \neq \emptyset$ and $A \cup B \neq V$. If $|\delta_H(A)| + |\delta_H(B)| \leq |\delta_H(A \cap B)| + |\delta_H(A \cup B)|$, then $A \cap B$ and $A \cup B$ are tight. If $|\delta_H(A)| + |\delta_H(B)| \leq |\delta_H(A - B)| + |\delta_H(B - A)|$, then $A - B$ and $A - B$ are tight.*

Proof: By submodularity and symmetry of the cut function $|\delta_G| : 2^V \to \mathbb{R}_+$, we have

$$|\delta_G(A)| + |\delta_G(B)| \geq |\delta_G(A \cap B)| + |\delta_G(A \cup B)|$$

and also

$$\delta_G(A)| + |\delta_G(B)| \geq |\delta_G(A - B)| + |\delta_G(B - A)|.$$

Now if

$$|\delta_H(A)| + |\delta_H(B)| \leq |\delta_H(A \cap B)| + |\delta_H(A \cup B)|$$

then we have

$$|\delta_G(A \cap B)| + |\delta_G(A \cup B)| \geq |\delta_H(A \cap B)| + |\delta_H(A \cup B)| \geq |\delta_H(A)| + |\delta_H(B)| = |\delta_G(A)| + |\delta_G(B)|$$

where the first inequality follows from the cut-condition, the second from our assumption and the third from the tightness of A and B. It follows that

$$|\delta_G(A \cap B)| = |\delta_H(A \cap B)|$$

and

$$|\delta_G(A \cup B)| = |\delta_H(A \cup B)|.$$

The other claim is similar. $\square$

Corollary 10 *If A, B are tight sets and $\delta_H(A-B, B-A) = \emptyset$ then $A \cap B$ and $A \cup B$ are tight.*

Proof: We note that

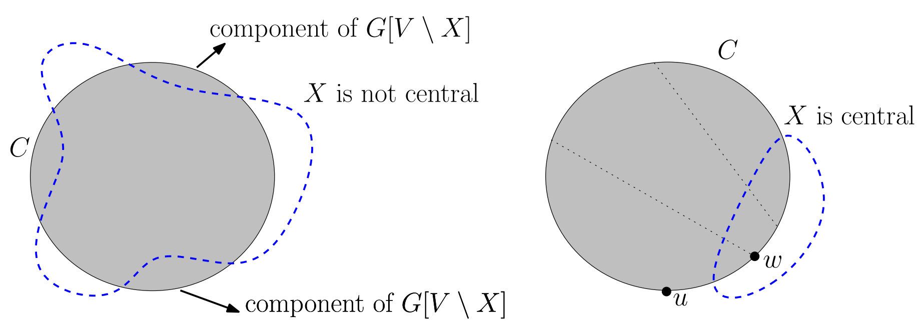
$$|\delta_H(A)| + |\delta_H(B)| = |\delta_H(A \cap B)| + |\delta_H(A \cup B)| + 2|\delta_H(A-B, B-A)|.$$

Thus, if $\delta_H(A-B, B-A) = \emptyset$ we have $|\delta_H(A)| + |\delta_H(B)| = |\delta_H(A \cap B)| + |\delta_H(A \cup B)|$ and we apply the previous lemma. □

Proof: Now we come to the proof of the Okamura-Seymour theorem. Recall that G, H is a counter example with $2|E| - |R|$ minimal. Then we have established that:

1. G is 2-connected.
2. every demand edge is in a tight cut.
3. no supply edge is parallel to a demand edge.

Without loss of generality we assume that the all the demands are incident to the outer/unbounded face of G. Since G is 2-connected the outer face is a cycle C. Let $X \subset V$ be a tight set; a tight set exists since each demand edge is in some tight set. Then if $X \cap C$ is not a contiguous segment, X is not a central set as can be seen informally by the picture below; $G[V \setminus X]$ would have two or more connected components.



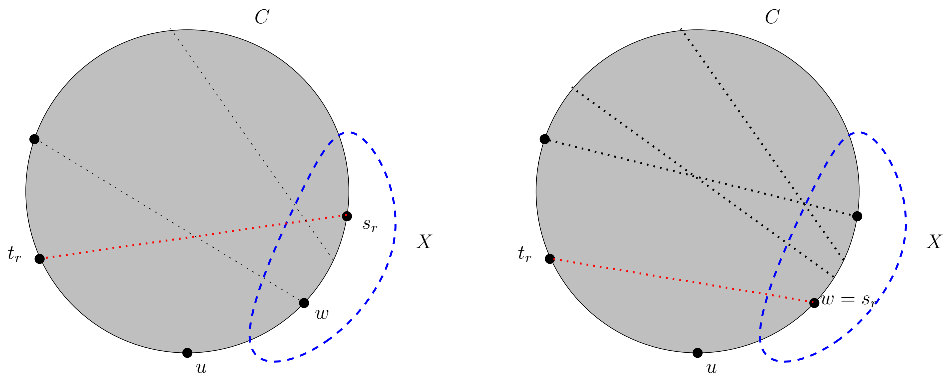
From Corollary 8 we can assume the existence of a tight set X such that $X \cap C$ is a contiguous segment. Choose such a tight set with $X \cap C$ minimal.

Let uw be one of the two edges of the cycle C that crosses X; let $w \in X$ and $u \notin X$. Since X is tight, $\delta_R(X) \neq \emptyset$. For each $r \in \delta_R(X)$, let s_r, t_r be the endpoints of r with $s_r \in X \cap C$ and $t_r \notin X \cap C$. Choose $r \in \delta_R(X)$ such that t_r is closest (in distance along the cycle C) to u in $C - X$. Note that r is not parallel to uw. So if $s_r = w$ then $t_r \neq u$ and if $t_r = u$ then $s_r \neq w$. Let $v \in \{u, w\} \setminus \{s_r, t_r\}$, v exists by above; for simplicity choose $v = w$ if $s_r \neq w$.

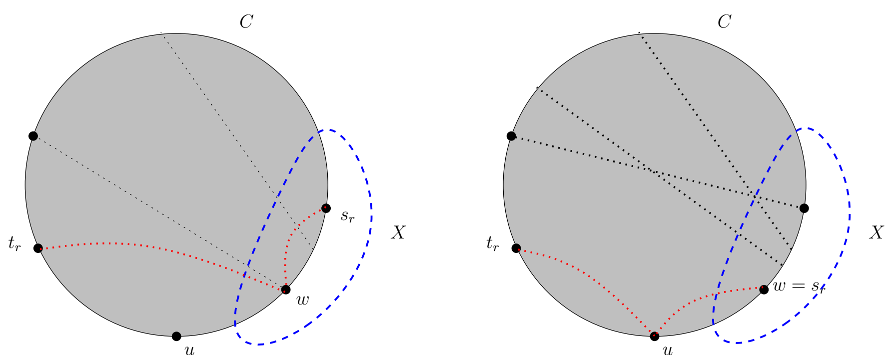
Let $R' = (R \setminus \{s_r, t_r\}) \cup \{s_r v, v t_r\}$. That is, we replace the demand edge $s_r t_r$ by two new demand edges $s_r v$ and $v t_r$ as shown in the figure.

Claim 11 *G satisfies cut condition for R' and $E + R'$ induces an Eulerian graph.*

Assuming claim, we are done because $2|E| - |R'| < 2|E| - |R|$ and by induction R' has an integral multiflow in G, and R has an integer multiflow if R' does.



In the picture on the left $v = w$ and on the right $v = u$.



Replacing $s_r t_r$ by new demands $s_r v$ and $v t_r$.

Trivial to see $E + R'$ induces an Eulerian graph. Suppose G does not satisfy the cut condition for the demand set R'. Let Y be a cut that violates the cut condition for R'. For this to happen Y must be a tight set for R in G; this is the reason why replacing $s_r t_r$ by $s_r v$ and $v t_r$ violates the cut condition for Y for R'. By complementing Y if necessary we can assume that $v \in Y, s_r, t_r \notin Y$. Further, by Corollary 8, we can assume Y is central and hence $Y \cap C$ is a contiguous segment of C.

By choice of r there is no demand r' between $Y - X$ and $X - Y$. If there was, then $t_{r'}$ would be closer to u than t_r. We have X, Y tight and

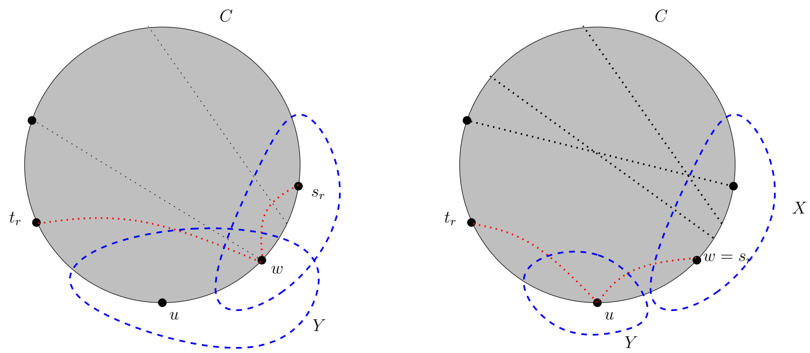
$$\delta_R[X - Y, Y - X] = \emptyset.$$

We consider two cases. First, suppose $X \cap Y \neq \emptyset$ (this is guaranteed if $v = w$). Then from Corollary 10, $X \cap Y$ and $X \cup Y$ are tight since $X \cap Y \neq \emptyset$ by assumption and $X \cup Y \neq V$ (since $t_r \in V \setminus (X \cup Y)$). $X - Y \neq \emptyset$, since $s_r \in X - Y$. Since $X \cap Y$ is a tight set and $X \cap Y \neq X$, it contradicts the choice of X as the tight set with $X \cap C$ minimal. If $X \cap Y = \emptyset$ then $v = u$ and $u \in Y$; again $X \cup Y \neq V$. Note that the edge uw joins X and Y. In this case we claim that $X \cup Y$ does not satisfy the cut condition which is a contradiction. To see this note that

$$|\delta_G(X \cup Y)| \leq |\delta_G(X)| + |\delta_G(Y)| - 2$$

since uw connects X to Y. However,

$$|\delta_H(X \cup Y)| = |\delta_H(X)| + |\delta_H(Y)| = |\delta_G(X)| + |\delta_G(Y)|$$



Tight set Y in the two cases.

where the first inequality follows since $X \cap Y = \emptyset$ and there are no demand edges between $X - Y$ and $Y - X$. The second inequality follows from the tightness of X and Y. $\square$

2 Sparse Cuts, Concurrent Multicomodity Flow and Flow-Cut Gaps

In traditional combinatorial optimization, the focus has been on understanding and characterizing those cases where cut condition implies existence of fractional/integral multiflow. However, as we saw, even in very restrictive settings, cut condition is not sufficient. A theoretical CS/algorithms perspective has been to quantify the "gap" between flow and cut. More precisely, suppose G satisfies the cut condition for H. Is it true that there is a feasible multiflow in G that routes λd_i for each pair $s_i t_i$ where λ is some constant in $(0, 1)$?

There are two reasons for considering the above. First, it is a mathematically interesting question. Second, and this was the initial motivation from a computer science/algorithmic point of view, is to obtain approximation algorithms for finding "sparse" cuts in graphs; these have many applications in science and engineering. The following is known.

Theorem 12 *Given a multiflow instance, it is co-NP complete to check if the cut-condition is satisfied for the instance.*

Definition 13 *Given a multiflow instance the maximum concurrent flow for the given instance is the maximum $\lambda \geq 0$ such that there is a feasible multiflow if all demand values are multiplied by λ.*

Proposition 14 *There is a polynomial time algorithm that, given a multiflow instance, computes the maximum concurrent flow.*

Proof: Write a linear program:

$$\begin{array}{rl} & \max \lambda \\ \text{flow for each} & s_i t_i \geq \lambda d_i \end{array}$$

Flow satisfies capacity constraints. We leave the details to the reader. $\square$

Definition 15 *Given a multiflow instance on G, H, the sparsity of a cut $U \subset V$ is*

$$sparsity(U) := \frac{c\left(\delta_G(U)\right)}{d\left(\delta_H(U)\right)}.$$

A sparsest cut is $U \subset V$ such that $sparsity(U) \leq sparsity(U')$ for all $U' \subset V$. We refer to $\min_{U \subset V} sparsity(U)$ as the min-sparsity of the given multiflow instance.

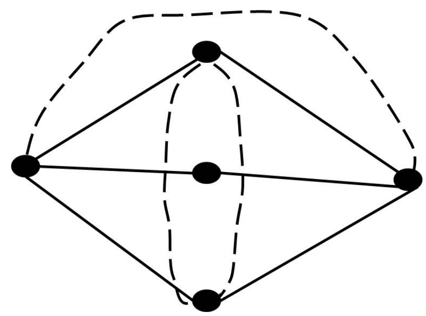
Observation 16 *(G, H) satisfies the cut condition implies sparsity$(U) \geq 1$ for all $\mathcal{U} \subset V$.*

Proposition 17 *In many multiflow instance, if λ^* is the max concurrent flow then*

$$\lambda^* \leq sparsity(U), \forall U \subset V.$$

The ratio $\frac{min-sparity}{\lambda^*}$ is the flow cut gap for the given instance.

For example,



with capacities and demands equal to 1, the flow-cut gap is $\frac{4}{3}$. Min-sparsity for the above instance is 1 while $\lambda^* = \frac{3}{4}$. In general, we are interested in quantifying flow-cut gaps for classes of instances rather than a particular instance.

In the sequel, we think of G and H as "topological" graphs in that they are not endowed with capacities and demands. A multiflow instance on G, H is defined by $c : E \to \mathbb{R}_+$ and $d : R \to \mathbb{R}_+$. Note that by setting $c(e) = 0$ or $d(r) = 0$, we can "eliminate" some edges. We define $\alpha(G, H)$, the flow-cut gap for G, H, as the supremum over all instances on G, H defined by capacities $c : E \to \mathbb{R}_+$ and $d : R \to \mathbb{R}_+$. We can then define for a graph G:

$$\alpha(G) = \sup_{\substack{H=(T,R) \\ T \subseteq V}} (G, H).$$

Some results that we mentioned on the sufficiency of cut condition for feasible flow can be restated as follows: $\alpha(G, H) = 1$ if $|R| = 2$ (Hu's theorem), $\alpha(G, H) = 1$ if G is planar and T is the vertex set of a face of G (Okamura-Seymour theorem), and so on. What can we say about $\alpha(G)$ for an arbitrary graph?

Theorem 18 (Linial-London-Rabinovich, Aumann-Rabani) *$\alpha(G) = O(\log n)$ where $n = |V|$ and in particular $\alpha(G, H) = O(\log |R|)$ i.e. the flow-cut gap is $O(\log k)$ for k-commodity flow. Moreover there exist graphs G, H for which $\alpha(G, H) = \Omega(\log |R|)$, in particular there exist graphs G for which $\alpha(G) = \Omega(\log n)$.*

Conjecture 19 *$\alpha(G) = O(1)$ if G is a planar graph.*

Theorem 20 (Rao) *$\alpha(G) = O(\sqrt{\log n})$ for a planar graph G.*

References

[1] Lex Schrijver, "Combinatorial Optimization: Polyhedra and Efficiency", Chapter 74, Vol. C, Springer-Verlag, 2003.