Combinatorial Optimization
Fall 2005
Professor B. Guenin

Chris Almost

Contents

Office Hours: Thursday, 3:30–5:00

1 Review of LP Duality

Primal (max)	Dual (min)
$=$ constraint	unrestricted
unrestricted	$=$ constraint
$\leq$ constraint	non-negative var
non-negative var	$\geq$ constraint

Table 1: (P)–(D) conversion

1.1 Theorem (Weak Duality). *If x and y are feasible solutions for the primal and its dual, respectively, then the value of the primal objective function on x is less than or equal to the value of the dual objective function on y.*

1.2 Theorem (Strong Duality). *If the primal problem has an optimal solution x^* then the dual problem has an optimal solution y^* and the value of the primal objective function on x^* is equal to the value of the dual objective function on y^*. If the primal problem is unbounded then the dual is not feasible.*

1.3 Theorem (Complementary Slackness). *Suppose x^* and y^* are feasible for the primal and dual, repsectively. Then both x^* and y^* are optimal if and only if*

1. $x_j^* = 0$ *or the j^{th} constraint of the dual is binding, for all j*
2. $y_i^* = 0$ *or the i^{th} constraint of the dual is binding, for all i*

These conditions are known as the complementary slackness conditions.

2 Shortest Paths

2.1 Definition. A *digraph* is a pair $G = (V, E)$ where V is a set of *nodes* and E is a set of *arcs* between those nodes. A (v_1, v_k)-*dipath* is a sequence of nodes $P = (v_1, \dots, v_k)$ such that, for all $i = 1, \dots, k-1$, $v_i v_{i+1} \in E$. A dipath is *simple* if there are no repeated nodes. P is a *dicircuit* if P is a dipath and $v_1 = v_k$.

An (r,s)-dipath Q can be decomposed into a simple (r,s)-dipath P and a collection of dicircuits $\mathscr{C}$ if for every arc $uv \in E$ the number of times that uv occurs in Q is equal to the total number of times uv occurs in P and all of the dicircuits in $\mathscr{C}$. (Prove this as an exercise.)

2.1 Shortest Dipath Problem

The problem is simply stated: given a digraph $G = (V, E)$, nodes $r, s \in V$, and an assigment of costs $c \in \mathbb{R}^E$, find the the lowest cost (r,s)-dipath.

For notation, write $c(E') = \sum_{e \in E'} c_e$ if $E' \subseteq E$, and write $c(P) = c(E(P))$ if P is a dipath. Notice that if there is a negative cost dicircuit then the problem is unbounded, since the cost can be arbitrarily lowered by following the circuit many times. If there is no negative cost dicircuit then it follows that there is a lowest cost (r,s)-dipath that is simple.

Notation. Given $x \in \mathbb{R}^E$ and $v \in V$, for a node $u \in V$ we write

$$f_x^+(u) = \sum_{v \in V : uv \in E} x_{uv} \qquad \text{and} \qquad f_x^-(u) = \sum_{v \in V : vu \in E} x_{vu}$$

and finally $f_x(u) = f_x^-(u) - f_x^+(u)$.

Let P a simple (r,s)-dipath. Define $x \in \mathbb{R}^E$ as follows:

$$x_{uv} = \begin{cases} 1 & \text{if } uv \in E(P) \\ 0 & \text{otherwise} \end{cases}$$

Notice that

$$f_x(u) = \begin{cases} 0 & \text{if } u \notin V(P) \\ 0 & \text{if } u \in V(P),\, u \neq r,s \\ -1 & \text{if } u = r \\ 1 & \text{if } u = s \end{cases}$$

A linear programming relaxation of the shortest path problem is (P):

$$\text{Minimize} \qquad \textstyle\sum_{uv \in E} c_{uv} x_{uv} \qquad \text{s.t.}$$
$$f_x(u) = \begin{cases} 0 & \text{if } u \neq r,s \\ -1 & \text{if } u = r \\ 1 & \text{if } u = s \end{cases} \quad \text{for all } u \in V, \text{ and } x \geq 0$$

The dual of this LP is (D):

$$\text{Maximize} \qquad y_s - y_r \qquad \text{s.t.}$$
$$y_v - y_u \leq c_{uv} \text{ for all } u,v \in E$$

To obtain the lefthand side of the inequalities, notice that the columns of $f_x(u)$ (as a matrix acting on the collection of arcs), the column corresponding to uv has -1 in row u and $+1$ in row v.

Remark. Let $y \in \mathbb{R}^V$ be feasible for the dual. Then for any $\alpha \in \mathbb{R}$, if $y'_v = y_v - \alpha$ for all $v \in V$ then y' is feasible for the dual and has the same value for the objective function.

2.2 Lemma. *We shall call y a feasible potential if y is feasible for the dual and $y_r = 0$. If P is an (r,s)-dipath and y is a feasible potential then $c(P) \geq y_s$.*

Proof: By weak duality,

$$c(P) \geq \text{value of optimal solution for (P)} \geq \text{value of optimal solution for (D)} \geq y_s - y_r = y_s \qquad \square$$

Given $y \in \mathbb{R}^V$, call an arc uv an *equality arc* if $y_v = y_u + c_{uv}$. The complementary slackness conditions become “if $x_{uv} > 0$ then $y_v - y_u = c_{uv}$” (or uv is an equality arc).

2.3 Lemma. *If P is a simple (r,s)-dipath and y is a feasible potential then $c(P) = y_s$ if and only if all arcs of P are equality arcs.*

PROOF: Define

$$x_{uv}^P = \begin{cases} 1 & \text{if } uv \in E(P) \\ 0 & \text{otherwise} \end{cases}$$

x^P is feasible for (P) and y is feasible for (D), so

$$c(P) = c^T x^P \leq y_s - y_r = y_s$$

where the middle relation is equality if and only if the complementary slackness conditions hold. □

2.4 Lemma. *Suppose that G has no negative dicircuit. Let $r \in V$ be such that there is a dipath from r to every other node in V. For all $v \in V$ let y_v = cost of the shortest simple (r,v)-dipath. The y is a feasible potential.*

PROOF: No negative dicircuits imply that y_v = cost of the shortest (r,v)-dipath. Consider some arc $uv \in E$; we need to show that $y_u + c_{uv} \geq y_v$. There is an (r,u)-dipath P such that $c(P) = y_u$. Let Q be obtained from P by adding uv at the end. But $y_v \leq c(Q)$ since y_v is the length of the shortest (r,v)-dipath, and $c(Q) = y_u + c_{uv}$. □

2.5 Corollary. *There exist feasible potentials if and only if there are no negative dicircuits.*

PROOF: Show that if there is a negative dicircuit then (P) is unbounded as an exercise. In this case (P) is unbounded and so (D) is infeasible. □

2.2 Trees of shortest dipaths

2.6 Definition. A node $v \in V$ is said to be *reachable from r* if there exists a (r,v)-dipath. A digraph T is with node r is a *tree rooted at r* if

1. the underlying undirected graph is a tree
2. every node $v \in V \setminus \{r\}$ is reachable from r

2.7 Lemma. *Let G be a digraph and $r \in V(G)$. Then G contains a spanning tree rooted at r if and only if every node is reachable from r.*

2.8 Definition. Let G be a digraph with costs $c \in \mathbb{R}^E$ and T a spanning tree of G rooted at r. Then T is a *tree of shortest dipaths* if for all $v \in V$, the (unique) (r,v)-dipath in T is a shortest (r,v)-dipath of G.

2.9 Lemma. *Let T be a spanning tree rooted at r and let y be a feasible potential. If all arcs of T are equality arcs then T is a tree of shortest dipaths.*

PROOF: See 2.3 □

2.10 Proposition. *Let T be a tree.*

1. *If $uv \notin E(T)$ then $T + uv$ has a circuit C.*
2. *If $wz \in E(C)$ then $T + uv - wz$ is a tree.*

Dantzig's Algorithm:
INPUT: digraph $G = (V,E)$, $c \in \mathbb{R}^V$, $r \in V$
OUTPUT: negative dicircuit or a tree T of shortest dipaths

1. Find an initial spanning tree rooted at r. Denote by P_u the unique (r,u)-dipath in T.

2. For all $u \in V$, let $y_u = c(P_u)$. Try to find a non-tree arc uv such that $y_u + c_{uv} < y_v$. If no such arc exists then STOP; T is a tree of shortest dipaths. If v is a node of P_u then STOP; the dicircuit in $T + uv$ is negative. Let zv be the last arc of P_v. Set $T \leftarrow T + uz - zv$.
3. GOTO 2

The algorithm terminates because the sum of the distances to each node from r never increases, and in fact it decreases on each step of the algorithm. There are a finite number of spanning trees, so the algorithm must terminate.

Dijkstra's Algorithm:
INPUT: digraph $G = (V, E)$, $c \in \mathbb{R}_+^E$, $r \in V$
OUTPUT: tree of shortest dipath rooted at r

1. Set $y_v = 0$ for each node $v \in V$.
2. Let E' be the set of equality arcs, let $G' = (V, E')$, and let R be the set of nodes reachable from r in G'. If $R = V$, GOTO 3. Increase uniformly y_u, for all $u \in V \setminus R$, until some arc in $E \setminus E'$ becomes an equality arc. GOTO 2
3. Find a spanning tree rooted at r in G' and STOP.

Dijkstra's algorithm terminates because the cardinality of R increases at every interation of the algorithm. Clearly the algorithm is correct.

3 Maximum Flows

Given a digraph $G = (V, E)$ and distinct nodes $r, s \in V$ (known as the source and sink, respectively), and $u \in (\mathbb{R}_+ \cup \{+\infty\})^E$ arc capacities.

Notation. For any $R \subseteq V$, $\delta(R) := \{uv \in E \mid u \in R, v \notin R\}$. $\delta(R)$ is the *boundery* of R. We write $\delta(v) = \delta(\{v\})$ and $\delta(\overline{v}) = \delta(V \setminus \{v\})$. Recall f_x above. We will know be writing it as $f_x(v) = x(\delta(\overline{v})) - x(\delta(v))$.

The maximum (r, s)-flow problem is

$$\begin{gathered}\text{Maximize} \qquad f_x(s) \qquad \text{s.t.} \\ f_x(v) = 0 \text{ for all } v \in V \setminus \{r, s\} \text{ and } 0 \le x \le u\end{gathered}$$

3.1 Definition. An (r, s)-*flow* is a feasible solution to the LP. An (r, s)-flow is *maximum* if it is an optimal solution to the LP.

3.1 Optimality Conditions and an Upper Bound

3.2 Definition. If $R \subseteq V$ is such that $r \in R$ and $s \notin R$ then the collection of edges $\delta(R)$ is called an (r, s)-*cut*.

3.3 Proposition. *For any (r, s)-cut $\delta(R)$ and any (r, s)-flow x,*

$$f_x(s) = x(\delta(R)) - x(\delta(\overline{R}))$$

PROOF:

$$f_x(s) = \sum_{v \in V \setminus R} f_x(v) = \sum_{v \in V \setminus R} x(\delta(\overline{v})) - x(\delta(v))$$

Consider each arc vw and let us see how many times x_{vw} appears in the above sum. If both $v,w \in R$ then x_{vw} appears zero times. If $v,w \notin R$ then x_{vw} appears with coefficient one once in the above sum (contributed by $x(\delta(\overline{w})))$ and x_{wv} appears with coefficient -1 once in the sum (contributed by $x(\delta(v)))$. If $v \notin R$ and $w \in R$ then x_{vw} appears with coefficient one and if $v \in R$ and $w \notin R$ then x_{wv} appears with coefficient -1 in the sum. □

For any (r,s)-flow x and any (r,s)-cut $\delta(R)$,

$$f_x(s) = x(d(R)) - x(\delta(\overline{R})) \leq u \leq u(\delta(R))$$

Thus if $f_x(s) = u(\delta(R))$ then the flow is maximum.

3.4 Definition. $u(\delta(R))$ is the *capacity* of the (r,s)-cut.

3.5 Theorem (Max-Flow Min-Cut). *If there exists a maximum (r,s)-flow then*

$$\max\{f_x(s) \mid x : (r,s)\text{-flow}\} = \min\{u(\delta(R)) \mid \delta(R) : (r,s)\text{-cut}\}$$

Moreover, if u is integral then there is a maximum (r,s)-flow which is integral.

3.6 Definition. Let P be an undirected path. P is *x-incrementing* if

1. for each forward arc e, $x_e < u_e$
2. for each backward arc e, $x_e > 0$.

An (r,s)-x-incrementing path is called *x-augmenting*. The *x-width* of an x-augmenting path is the minimum value in $\{u_e - x_e \mid e \text{ a forward arc of } P\} \cup \{x_e \mid e \text{ a backward arc of } P\}$. x' is obtained from x by *augmenting along P* if

$$x'_e = \begin{cases} x_e & \text{if } e \notin E(P) \\ x_e + (x\text{-width}) & \text{if } e \text{ is a forward arc of } P \\ x_e - (x\text{-width}) & \text{if } e \text{ is a backward arc of } P \end{cases}$$

If x is an (r,s)-flow then so is x' and $f_{x'}(s) = f_x(s) + x$-width.

3.7 Definition. Let x be an (r,s)-flow in $G = (V,E)$ with $u \in \mathbb{R}^E_+$. The *auxillary digraph* $G(x) = (V,E(x))$, where $vw \in E(x)$ if either $vw \in E$ and $x_{vw} < u_{vw}$ or $wv \in E$ and $x_{wv} > 0$.

x-augmenting paths in G correspond to (r,s)-dipaths in $G(x)$, as we shall see.

3.8 Lemma. *The following are equivalent.*

1. *There exists an x-augmenting path*
2. *$G(x)$ has an (r,s)-dipath*
3. *There does not exist $R \subseteq V$ with $r \in R$, $s \notin R$ such that $\delta_{G(x)}(R) = \emptyset$.*
4. *There does not an (r,s)-cut $\delta(R)$ such that $x(\delta(R)) = u(\delta(R))$ and $x(\delta(\overline{R})) = 0$.*
5. *$f_x(s) < \min\{u(\delta(R)) \mid \delta(R)$ is an (r,s)-cut$\}$.*

PROOF: Exercise. □

3.9 Corollary. *Let x be an (r,s)-flow. Then*

1. *there exists an x-augmenting path* OR
2. *there exists an (r,s)-cut $\delta(R)$ such that $f_x(s) = u(\delta(R))$.*

PROOF (MFMC THEOREM): Choose to be a maximum (r,s)-flow. Then there are no x-augmenting paths, so there is an (r,s)-cut $\delta(R)$ such that $f_x(s) = u(\delta(R))$. If u is integral then choose x which is a maximum (r,s)-flow amoung all integral (r,s)-flows. Suppose that x is not a maximum (r,s)-flow, so that there is an x-augmenting path P. Let x' be obtained from x by augmenting along P. But everything is integral, so the x-width is integral and hence so is x'. This contradicts that x was maximal amoung the integral flows. □

Augmenting Path Algorithm:

INPUT: digraph $G = (V,E)$, $r,s \in V$, $u \in \mathbb{R}^E_+ \cup \{+\infty\}$
OUTPUT: maximum (r,s)-flow x (if it exists) and minimum (r,s)-cut $\delta(R)$ $(f_x(s) = u(\delta(R)))$

1. Set $x = 0$.
2. Find $G(x)$. If there is no (r,s)-dipath in $G(x)$ then let R be the set of nodes reachable from r in $G(x)$ and STOP with x, $\delta(R)$. Otherwise find an (r,s)-dipath P of $G(x)$. Let Q be the corresponding x-augmenting path. Augment along Q and GOTO 2.

Remark. There are examples of graphs where this algorithm runs very badly. If the values of u are not rational then the algorithm may not even converge to the optimal value.

3.10 Definition. An x-augmenting path is *shortest* if it has a minimum number of arcs amoung all x-augmenting paths.

3.11 Theorem. *If we always choose a shortest augmenting path then the number of augmentations completed by the algorithm is less than or equal to* $|E||V|$.

Note that we can find a shortest (r,s)-dipath in $G(x)$ in $O(|E|)$, so we can solve MFMC problem in $O(|V||E|^2)$.

PROOF: Let x be an (r,s)-flow, $v \in V$, and $d_x(v) =$ length of shortest (r,v)-dipath in $G(x)$. Suppose that P is a shortest x-augmenting path, and x' is obtained from x by augmenting on P.

Claim. $d_{x'}(v) \geq d_x(v)$ for every $v \in V$.

Suppose there is $v \in V$ such that $d_{x'}(v) < d_x(v)$. Amoung all such v choose one minimizing $d_{x'}(v)$. Let P' be a shortest (r,v)-dipath in $G(x')$. Now $d_{x'}(r) = 0 = d_x(r)$, so $v \neq r$. Let $w \in V$ such that wv is the last arc of P'. Then $d_x(w) \leq d_{x'}(w) = d_{x'}(v) - 1 < d_x(v) - 1$ so the arc wv cannot belong to $G(x)$. Let Q be the (r,s)-dipath of $G(x)$ corresponding to P. Since $wv \notin G(x)$ and $wv \in G(x)$, this implies that $vw \in G(x)$ and in particular, $vw \in Q$. But then $d_x(w) = d_x(v) + 1$ since A is a shortest path, a contradiction.

Let $x^0, x^1, \dots, x^k$ be the successive (r,s)-flows obtained by the algorithm, and let Q_i be the x^i-augmenting path and P_i be the corresponding path in $G(x^i)$.

Claim. Suppose

1. $wv \in G(x^i)$ is an arc of P_i
2. $vw \in G(x^j)$ is an arc of P_j (for $j > i$)

Then $d_{x^j}(w) \geq d_{x^i}(w) + 2$.

Indeed, $d_{x^j}(w) = d_{x^j}(v) + 1 \geq d_{x^i}(v) + 1 = d_{x^i}(w) + 2$.

Call an arc $vw \in Q_i$ *critical* if $x^{i+1}_{vw} = 0$ or u_{vw}. Every Q_i has a critical arc. Critical arcs of Q_i correspond to arcs that are reversed when going from $G(x^i)$ to $G(x^{i+1})$. It follows from the second claim that if an arc is reversed twice then the distance in $G(x)$ to its tail increases by 2. But $d_x(v) \leq |V|$, so there are at most $|E||V|$ augmentations. □

3.2 LP Interpretation and LP Duality

The MFMC problem as a linear program becomes

$$\text{Maximize} \quad f_x(s) \quad \text{s.t.}$$
$$f_x(v) = 0 \text{ for all } v \in V \setminus \{r, s\},\ x_e \le u_e \text{ for all } e \in E,\ x \ge 0$$

3.12 Exercise. We may remove all arcs $\delta(\overline{r})$ and $\delta(s)$ without changing the optimal value of the primal (P). We may assume that $\delta(\overline{r}) = 0$ and $\delta(s) = 0$.

We now find the dual problem. Let y correspond to the equality constraints and z correspond to the $\le$ constraints. There are 4 types of arcs

1. Arcs vw such that $v, w \notin \{r, s\}$. The corresponding dual constraint is $-y_v + y_w + z_{vw} \ge 0$.
2. Arcs rw such that $w \ne s$. The dual constraint is $y_w + z_{rw} \ge 0$.
3. Arcs vs such that $v \ne r$. The dual constraint is $-y_v + z_{vs} \ge 1$.
4. An arc rs, and the corresponding constraints is $z_{rs} \ge 1$.

$$\text{Minimize} \quad \textstyle\sum_{e \in E} u_e z_e \quad \text{s.t.}$$
$$-y_v + y_w + z_{vw} \ge 0, \text{ for } vw \in E \text{ with } v, w \notin \{r, s\}$$
$$y_w + z_{rw} \ge 0 \text{ where } w \ne s$$
$$-y_v + z_{vs} \ge 1 \text{ where } v \ne r$$
$$z_{rs} \ge 1 \text{ if } rs \in E$$
$$z \ge 0$$

We can simplify this by writing it as

$$\text{Minimize} \quad u^T z \quad \text{s.t.}$$
$$-y_v + y_w + z_{vw} \ge 0, \text{ for } vw \in E$$
$$z \ge 0,\ y_r = 1,\ y_s = 0$$

Let $\delta(R)$ be an (r, s)-cut. Define

$$y^*_v = \begin{cases} 1 & \text{if } v \in R \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad z^*_{vw} = \begin{cases} 1 & \text{if } vw \in \delta(R) \\ 0 & \text{otherwise} \end{cases}$$

Then it is clear that y^*, z^* is feasible for the dual and $u^T z^* = u(\delta(R))$. The MFMC theorem implies that there is an optimal solution to the dual which is integer.

3.3 Flow decomposition

3.13 Definition. A *circulation* is soltion to $f_x(v) = 0$ for all $v \in V$ and $x \ge 0$. Let C be a dicircuit and let $\alpha \ge 0$. Then αx^C is a *circuit circulation*.

3.14 Theorem. *Every circulation can be written as the sum of at most $|E|$ circuit circulations.*

PROOF: If $x = 0$ then we are done; it is the sum of no circuit circulations. We may assume that there is no arc e such that $x_e = 0$ and there are no isolated nodes. Since $f_x(v) = 0$ for all $v \in V$, every node has indegree at least 1. Therefore G has a dicircuit C (exercise). Let $\alpha = \min_{e \in C} x_e$. Then αx^C is a circuit circulation and $x - \alpha x^C$ is a circulation since $f_{x - \alpha x^C}(v) = f_x(v) - \alpha f_x(v) = 0$. By induction we are done. □

3.15 Definition. Let P be a simple (r,s)-dipath and $\alpha \geq 0$. Then αx^p is a *path flow*.

3.16 Theorem. *Every (r,s)-flow x can be written as the sume of at most $|E|$ path flows and circuit circulations.*

PROOF: Let G' be obtained from E by adding arc sr and obtain x' from x by setting $x'_{sr} = f_x(s)$. Then x' is a circulation, so it can be decomposed into circuit circulations on G'. Any of the circuits that contain sr are path flows in G, and the rest of the circuits remain circuits in G. □

3.17 Definition. Let $G = (V,E)$, $d \in \mathbb{R}^V$ be some demands, and $u \in \mathbb{R}^E$ be the capacities. A flow is a solution to $f_x(v) = d_v$ for all $v \in V$ and $0 \leq x \leq u$.

See book for how to solve this problem.

3.4 Bipartite Matchings

For this section, G is a bipartite graph with vertex partition A, B. Recall that a *matching* M is a subset of the edges such that for all $v \in V$, $|\delta(v) \cap M| \leq 1$. $\nu(G)$ is the size of that largest matching in G. Let $\overrightarrow{G}$ be the directed graph obtained from G by adding new vertices r and s and adding arcs from everything in B to s and from r to everything in A. Direct every old edge towards s. Give a capacity of 1 to each of the new edges and a capacity of $+\infty$ to each of the old edges.

3.18 Proposition. *G has a matching of size k if and only if $\overrightarrow{G}$ has an integer (r,s)-flow of capacity k.*

Recall that a *cover* X of G is a subset of the vertices such that every edge of G has at least one end in X. Notice that if M is a matching of G then $|X| \geq |M|$.

3.19 Proposition. *G has a cover of size k if and only if $\overrightarrow{G}$ has an (r,s)-cut $\delta(R)$ of capacity k.*

3.20 Theorem (König). *In a bipartite graph the size of the largest matching is equal to the size of the smallest cover.*

PROOF: Let k be the maximum flow value of $\overrightarrow{G}$. 3.18 implies that there is a matching of size k. MFMC implies there is an (r,s)-cut of capacity k, and 3.19 implies there is a cover of size k. □

There is another proof of König's Theorem the uses totally unimodular matrices.

3.5 Another Maximum Flow Algorithm

3.21 Definition. $x \in \mathbb{R}^E_+$ is a (r,s)-*preflow* if $f_x(v) \geq 0$ for all $v \in V \setminus \{r\}$ and $0 \leq x \leq u$. A node $v \in V$ is *active* if $f_x(v) > 0$. $d \in \mathbb{Z}^V_+$ is a *valid labelling (with respect to x)* if

- $d(r) = |V|$, $d(s) = 0$
- for each $vw \in E(x)$, $d(v) \leq d(w) + 1$ (3.14 in the book)

$vw \in E$ is *admissible* if v is active and $d(v) = d(w) + 1$.

Notice that an (r,s)-preflow which has no active nodes is an (r,s)-flow. We may assume that every arc goes both ways. Indeed, if $vw \in E$, if $wv \notin E$ then add wv to G and set $u_{wv} = 0$.

3.22 Definition. For every arc $vw \in E$, define $\tilde{u}_{vw} = (u_{vw} - x_{vw}) + x_{wv}$. Let $E(x) = \{vw \in E \mid \tilde{u}_{vw} > 0\}$ and $G(x) = (V, E(x))$. $G(x)$ is the *auxilliary digraph* and $\tilde{u}$ are the residual capacities.

Notation. Let $d_x(v,w)$ denote the number of arcs in the shortest (v,w)-dipath in $G(x)$. Setting $d(v) = d_x(v,s)$ for all $v \in V$ satisfies (3.14) by violates $d(r) = |V|$.

3.23 Lemma. *If x is an (r,s)-preflow and d is a valid labelling then $d_x(v,w) \geq d(v) - d(w)$.*

PROOF: Let G' be obtained from $G(x)$ be reversing all of the arcs. Suppose that every arc vw of G' has weight $c_{vw} = 1$. Then (3.14) implies that for all $wv \in G'$, $d(v) \leq d(w) + c_{vw}$. This implies that d is a feasible potential for G', which further implies that $d(v) - d(w)$ is a lower bound on the length of any (w,v)-dipath in G', and hence of the (v,w)-dipaths in $G(x)$. □

3.24 Definition. An (r,s)-cut $\delta(R)$ is *saturated* for an (r,s)-preflow x if $x(\delta(R)) = u(\delta(R))$ and $x(\delta(\overline{R})) = 0$.

3.25 Lemma. *If x is an (r,s)-preflow and d is a valid labelling then some (r,s)-cut is saturated.*

PROOF: Suppose there is a (r,s)-dipath P in $G(x)$. We may assume that P is simple, and hence it uses at most $n-1$ arcs. But then $n-1 \geq \ell(P) \geq d(r) - d(s) = n$, a contraction. Hence there are no (r,s)-dipaths in $G(x)$. Let R be the set of nodes reachable from the source in $G(x)$. Then $\delta(R)$ is an (r,s)-cut. Notice that $\delta_{G(x)}(R) = \emptyset$, which implies that for every $vw \in \delta_G(R)$, $x_{vw} = u_{vw}$, and for every $vw \in \delta(\overline{R})$, $x_{vw} = 0$. □

3.26 Corollary. *Let x be an (r,s)-preflow and d be a valid labelling. If x has no active nodes then x is a maximum (r,s)-flow and the (r,s)-cut $\delta(R)$ found above is a minimum cut.*

Push-Relabel Algorithm:

INPUT: digraph $G = (V,E)$, $r,s \in V$, $u \in (\mathbb{R}_+ \cup \{+\infty\})^E$
OUTPUT: maximum flow x

1. Set $x_e = \begin{cases} u_e & \text{if } e \text{ has tail } r \\ 0 & \text{otherwise} \end{cases}$ and set $d(r) = n$ and $d(v) = 0$ if $v \neq r$.
2. Find an active node v such that $d(v)$ is maximized; if there are no active nodes then STOP.
3. If vw is an admissible arc then let $\varepsilon = \min\{f_x(v), \tilde{u}_{vw}\}$, decrease x_{wv} by $\varepsilon' = \min\{\varepsilon, x_{wv}\}$, and increase x_{vw} by $\varepsilon - \varepsilon'$ (this is *pushing* on vw).

   If there are no admissible arcs with tail v then set $d(v) = \min\{d(w) + 1 \mid vw \in E(x)\}$.

   If v remains active GOTO 3.
4. GOTO 2.

In step (3), d remains a valid labelling since v was not the tail of an admissible arc. x remains a preflow after step (3) since no more than $f_x(v)$ flow is removed from v and more flow is added to w. Neither of these operations changes that x is a preflow.

3.27 Theorem. *The Push-Relabel Algorithm performs $O(n^2)$ relabels and $O(mn^2)$ pushes.*

PROOF: We shall first prove that, for every active node v, there is a (v,r)-dipath in $G(x)$. Indeed, $f_x(v) > 0$ implies that there is an (r,v)-dipath P in G such that $x_e > 0$ for all arcs e of P (exercise). This implies the claim. Now for any node v, $d(v) \leq 2n-1$. Indeed, only active nodes see their lable increase, so it suffices to show the result for active nodes v. By the above claim there is a (v,r)-dipath in $G(x)$, so $d_x(v,r) \leq n-1$. Recall that $d_x(v,r) \geq d(v) - d(r)$ and $d(r) = n$, so $n-1 \geq d(v) - n$, which proves the second claim. The labels never decrease, so each node is relabelled at most $2n-1$ times, which implies that there are $O(n^2)$ relabellings.

Call a push on vw *saturating* if $\tilde{u}_{vw} \leq f_x(v)$ and *non-saturating* otherwise. If a push is saturating then the arc vw is going to disappear from the new auxilliary digraph. Non-saturating implies that v will become inactive.

We claim that between two saturating pushes on the same arc vw, $d(v)$ increases by at least 2. Suppose that vw is admissible, so that $d(v) = d(w) + 1$. Since the push is assumed to be saturated, the arc disappears from $G(x)$ after the push. vw will reappear only if we push on wv. In this case wv will have to be admissible, which implies that $d'(w) = d'(v) + 1 \geq d(v) + 1 = d(w) + 2$, which proves the claim. Now $d(w) \leq 2n$, so there can be at most n saturating pushes on vw. Therefore there can be at most $O(mn)$ saturating pushes in total.

Let A be the set of active nodes and let $D = \sum_{v \in A} d(v)$. Initially $D = 0$ since $d(v) = 0$ for every node $v \neq r$. We claim that each saturating push increases D by at most $2n$, since by pushing on vw w may become active, but $d(w) \leq 2n$. Each non-saturating push decreases D, since a non-saturating push on vw implies that $d(v) > d(w)$ and v becomes inactive while w becomes active. The total amount of increase in D from relabelling is $O(n^2)$ and from saturated pushes is $O(mn^2)$. Since each non-saturated push decreases D and $D \geq 0$ (indeed, $d(v) \geq 0$ for all $v \in V$), the number of non-saturating pushes is $O(mn^2)$. Therefore the total number of pushes is at most $O(mn^2)$. □

3.6 Minimum Cost Flow

Given a graph $G = (V, E)$, demands $d \in \mathbb{R}^V$, capacities $u \in (\mathbb{R}_+ \cup \{\infty\})^E$, and costs $c \in \mathbb{R}^E$, solve the linear program

$$\begin{aligned} &\text{Minimize} \quad c^T x \quad \text{s.t.} \\ &f_x(v) = d_v \text{ for all } v \in V, \\ &-x_e \geq -u_e, \text{ and } x \geq 0 \end{aligned}$$

Special cases of this problem are the minimum (r,s)-dipath problem and the maximum (r,s)-flow problem.

3.28 Definition. Given $y \in \mathbb{R}^V$ define for all $vw \in E$ $\bar{c}_{vw} = c_{vw} + y_w - y_w$, the *reduced cost* of c_{vw}.

We say that a solution x to the primal satisfies the *optimality conditions* if there is $y \in \mathbb{R}^V$ such that $\bar{c}_e < 0$ implies that $x_e = u_e$ and $\bar{c}_e > 0$ implies that $x_e = 0$.

3.29 Theorem. *A solution x to the primal is optimal if and only if the optimality conditions are satisfied.*

PROOF: The dual is

$$\begin{aligned} &\text{Maximize} \quad \textstyle\sum_{v \in V} y_v d_v - \sum_{e \in E} u_e z_e \quad \text{s.t.} \\ &-y_v + y_w - z_{vw} \leq c_{vw} \text{ for all } vw \in E, z \geq 0 \end{aligned}$$

But if $\bar{c}_{vw}$ is the reduced cost associated with some feasible y then the condition becomes $z \geq -\bar{c}_{vw}$. Clearly by choosing $z'_e = \max\{0, -\bar{c}_e\}$ we get a solution (y, z') at least as good as any solution (y, z). Complementary Slackness proves the theorem. □

3.30 Definition. Let x be a flow and W an oriented circuit. The *x-width* of W is the miniimum value in $\{u_{vw} - x_{vw} \mid vw \text{ is a forward arc of } W\} \cup \{x_{vw} \mid vw \text{ is a backward arc of } W\}$. The *cost* is

$$c(W) = \sum_{e \text{ forward}} c_e - \sum_{e \text{ backward}} c_e$$

We say that x' is *obtained by pushing flow on W* if

$$x'_e = \begin{cases} x_e + x\text{-width} & e \text{ forward arc of } W \\ x_e - x\text{-width} & e \text{ backward arc of } W \\ x_e & \text{otherwise} \end{cases}$$

If x is a flow then x' is a flow of value $c^T x + x$-width $\times\, c(W)$. Hence we are looking for circuits with positive x-width and negative cost. We will be using the same graph $G(x)$ to look for these circuits, but this time we label the edges of $G(x)$

$$c'_{vw} = \begin{cases} c_{vw} & \text{if } vw \text{ comes from } vw \in E \\ -c_{vw} & \text{if } vw \text{ comes from } wv \in E \end{cases}$$

There is a one to one correspondence between x-incrementing oriented circuits W in G with $c(W) < 0$ and negative dicircuits in $G(x)$ with costs c'. This leads to an algorithm.

Algorithm (name):
INPUT: digraph $G = (V, E)$, demands $d \in \mathbb{R}^V$, capacities $u \in (\mathbb{R}_+ \cup \{\infty\})^E$, costs $c \in \mathbb{R}^E$
OUTPUT: minimum cost flow x

1. Find an initial flow x.
2. Find $G(x)$ with costs c'. Try to find an negative dicircuit W' in $G(x)$. If none exists then STOP, otherwise push flow on the corresponding x-incrementing circuit. GOTO 2

3.31 Theorem. *Let x be a flow. x is a minimum cost flow if and only if $G(x)$ with costs c' has no negative dicircuits.*

3.7 Multi-terminal-cut Problem

Let $G = (V, E)$ be a digraph, $u \in \mathbb{R}^E_+$, and $K \subseteq V$ be terminals. Find the minimum (r, s)-cuts for all pairs (r, s) such that $r, s \in K$. Clearly we can solve this problem by solving $\binom{|K|}{2}$ flow problems, but can we do better?

Let T be a tree such that

1. $V(T)$ is a partition of V
2. Each $V_i \in V(T)$ contains exactly one terminal
3. Each edge $e \in E(T)$ has a label t_e.

A tree satisfying, for any pair (r, s) of terminals, the capacity of the minimum (r, s) cut is the minium label t_e amoung edges e in the (r, s)-path of T, is called a *flow tree*. A flow tree with the property that if an edge e^* achieves the minimum for (r, s) then a minimum (r, s)-cut is given by the partition of V corresponding to the 2 trees obtained by deleting the edge e^* from T is called a *cut tree*.

Gomory-Hu Algorithm:
INPUT: digraph $G = (V, E)$, $u \in \mathbb{R}^E_+$, $K \subseteq V$ terminals
OUTPUT: flow tree T

1. Choose $r, s \in K$. Find a minimum (r, s)-cut $\delta(R)$ and let $S = V \setminus R$. R and S become vertices of the tree T with label $u(\delta(R))$.
2. Pick a vertex R of T that contains two terminal nodes $p, q \in K$. If there is no such vertex then STOP. Suppose $T \setminus \{R\}$ is the collection of subtrees $\{T_1, \ldots, T_r\}$. Let G_R be the graph obtained from G by contracting every node in T_i to a single node v_i.
3. Find a minimum capacity (p, q)-cut $\delta(X)$ in G_R. Let $P = R \cap X$ and $Q = R \setminus X$ be vertices of the tree replacing R. If $v_i \in X$ then join P to T_i with the edge that used to be between R and T_i, otherwise if $v_S \notin X$ then join Q to T_i. GOTO 2

See the text for a proof of this algorithm.

4 Matchings

4.1 The Tutte-Berg Formula

4.1 Definition. Let M be a mathcing of $G = (V, E)$ and let $v \in V$. If v is an end of an edge in M then v is *M-covered*, otherwise v is *M-exposed*. Let $\text{def}(G)$ be the minimum number of M-exposed vertices over all matchings M of G.

Recall that $\nu(G)$ is the size of the largest matching in G. Then $\text{def}(G) = |V| - 2\nu(G)$.

4.2 Definition. Given a matching M of G, P is an *M-alternating path* if its edges are alternatively in M and not in M. If both ends of P are M-exposed then P is said to be *M-augmenting*.

4.3 Theorem. *A matching M is maximum if and only if there is no M-augmenting path.*

PROOF: Suppose that there is an M-augmenting path P. Let $N = M \triangle E(P)$. Then N is a matching and $|N| = |M| + 1$, so M is not maximum. Suppose that M is not maximum. Then there is a matching N with $|N| > |M|$. Let $H = (V, M \triangle N)$, a subgraph of G. Then H has maximum degree at most 2, so it consists of isolated vertices, circuits, and paths. For every circuit C or H, $|C \cap M| = |C \cap N|$, and every path P in H is both M and N alternating. But N has more edges than M, so there is at least one path P such that $|P \cap M| < |P \cap N|$. Therefore the ends of P are M-exposed, so P is an M-augmenting path in G. □

We would like to find an upper bound on $\nu(G)$. Recall that if W is a cover fo G then $\nu(G) \leq |W|$, but this is not tight unless G is bipartite (*cf.* König's Theorem). Indeed, if $G = C_{2k+1}$ then $\nu(G) = k$ while the size of the smallest cover is $k + 1$.

Let $G = (V, E)$, $A \subseteq V$, and M a matching of G. Let $H_1, \dots, H_k$ be the components of $G \setminus A$ with an odd number of vertices. For each H_i, $i = 1, \dots, k$, either some vertex of $V(H_i)$ is M-exposed or M contains an edge with ends in both $V(H_i)$ and A. Since the second possibility can occur at most $|A|$ times, the number of M-exposed vertices is at least $k - |A|$.

Notation. Let $\text{oc}(G)$ denote the number of components of G with an odd number of vertices.

The discussion above proves $\text{def}(G) \geq \text{oc}(G \setminus A) - |A|$ for any $A \subseteq V$.

4.4 Theorem (Tutte-Berge Formula).

$$\text{def}(G) = \max_{A \subseteq V} \{\text{oc}(G \setminus A) - |A|\}$$

or equivalently,

$$\nu(G) = \min_{A \subseteq G} \{\tfrac{1}{2}(|V| - \text{oc}(G \setminus A) - |A|)\}$$

4.5 Definition. Let $G = (V, E)$ have an odd circuit C. Then $G \times C$ is the graph obtained by contracting all edges of C and removing all loops. The vertex of $G \times C$ corresponding to C is denoted by C as well.

4.6 Proposition. *Let C be any odd circuit of G. Then $\text{def}(G) \leq \text{def}(G \times C)$.*

PROOF: Find a maximum matching M for $G \times C$. Extend M to a matching of G by adding $\lfloor \frac{|C|}{2} \rfloor$ edges in C. If there was an edge of M incident with C in $G \times C$ then that edge in G is made the be incident with the remaining vertex of C. □

Call an odd circuit *tight* if the inequality in 4.6 is an equality. Call a vertex $v \in V$ *essential* if $\text{def}(G \setminus \{v\}) > \text{def}(G)$. v is *inessential* otherwise. Notice that if for $A \subseteq V$, we have $\text{def}(G) = \text{oc}(G \setminus A) - |A|$ then all vertices of A are essential.

4.7 Lemma. *Let $v_1v_2 \in E$ and suppose that v_1 and v_2 are inessential. Then there is a tight odd circuit C containing v_1v_2 and C is inessential in $G \times C$.*

PROOF: For $i = 1, 2$, $\text{def}(G\setminus\{v_i\}) = \text{def}(G)$, so there is a maximum matching M_i such that v_i is M_i-exposed. But M_i maximum implies that v_{3-i} is M_i-covered (since v_1v_2 is an edge). Let $H = (V, M_1 \triangle M_2)$. Again, the components of H are points, paths, and circuits. Let P be the path-component of H that contains v_1. Then $\deg_H(v_1) = 1$, so P is a simple path starting at v_1. Let w be the other end of P. If w is M_2 covered, then P would be M_1-augmenting, a contradiction since M_1 is maximum. Therefore w is M_1-covered and P has even length. If $w \neq v_2$ then v_1v_2P is an M_2-augmenting path, another contraction. Therefore $w = v_2$. It follows that $C = v_1v_2P$ is an odd circuit in G. Let $M' = M_1 \setminus E(C)$, so that M' is a matching of $G \times C$. The number of M'-exposed vertices of $G \times C$ is equal to the number of M_1-exposed vertices of G, which is $\text{def}(G)$. Therefore $\text{def}(G \times C) = \text{def}(G)$ by 4.6, so C is tight. Furthermore, M' shows that C is an inessential vertex in $G \times C$. □

PROOF (OF 4.4): We need to show that there is $A \subseteq V$ such that $\text{def}(G) = \text{oc}(G \setminus A) - |A|$. Suppose there is a counterexample G, where G has the fewest vertices amoung all counterexamples. We may assume that $V \neq \emptyset$. Suppose that G has an essential vertex $v \in V$. By minimality, there is $A \subseteq V \setminus \{v\}$ such that

$$\begin{aligned}\text{def}(G) + 1 &= \text{def}(G \setminus \{v\}) \\ &= \text{oc}((G \setminus \{v\}) \setminus A) - |A| \\ &= \text{oc}(G \setminus (A \cup \{v\})) - |A| \\ &= \text{oc}(G \setminus (A \cup \{v\})) - |A \cup \{v\}| + 1\end{aligned}$$

But this implies that $\text{def}(G) = \text{oc}(G \setminus (A \cup \{v\})) - |A \cup \{v\}|$, contradicting that G was a counterexample.

Therefore G has no essential vertices, so we may assume that there is an edge v_1v_2 such that v_1 and v_2 are inessential (if $E = \emptyset$ then choose $A = \emptyset$ for contradiction). Then there is a tight odd circuit C such that C is inessential in $G \times C$. By minimality there is $A \subseteq V \setminus (V(C) \cup \{C\})$ such that $\text{def}(G) = \text{def}(G \times C) = \text{oc}(G \times C \setminus A) - |A|$. But since $C \notin A$ and unshrinking an odd cycle does not change the parity of any of the components of $G \times C \setminus A$, it follows that $\text{oc}(G \times C \setminus A) = \text{oc}(G \setminus A)$. Again, this contradicts that G was a counterexample, so there is no counterexample. □

4.8 Definition. A matching M is *perfect* if every vertex is M-covered.

4.9 Theorem (Tutte). *G has a perfect matching if and only if for every $A \subseteq V$, $oc(G \setminus A) \leq |A|$.*

4.2 Finding Perfect Matchings

4.10 Definition. Let M be a matching of G. A tree T with root r is *M-alternating* if

- r is M-exposed
- all other vertices of T are M-covered
- all (r, v)-paths in T are M-alternating

$A(T)$ is the set of vertices of T that are at an odd distance in T from r and $B(T)$ is the set of vertices of T at an even distance in T from r.

Suppose we have an edge $vw \in E$ such that $v \in B(T)$ and $w \notin V(T)$ such that w is M-covered. Then $wz \in M$ for some $z \in V$, and $z \notin V(T)$. Therefore we can extend T to $T + vw + wz$.

Suppose we have an edge $vw \in E$ such that $v \in B(T)$ and $w \notin V(T)$ such that w is M-exposed. Then (r, w) is an M-augmenting path in $T + vw$, so use the path to augment M.

4.11 Definition. An M-alternating tree T is frustrated if all edges with ends in $B(T)$ have an edge in $A(T)$.

4.12 Proposition. *If T is frustrated then G has no perfect matching.*

PROOF: In $G \setminus A(T)$, vertices in $B(T)$ are odd components. But $|A(T)| < |B(T)| \leq \text{oc}(G \setminus A(T))$, so there is no perfect matching by Tutte's Theorem. □

Algorithm for bipartite graphs:

INPUT: graph G

OUTPUT: perfect matching M of G, or proof that there is no perfect matching

1. Set $M = \varnothing$ and $T = (r, \varnothing)$ (for any $r \in V$).
2. Suppose there is $vw \in E$ such that $v \in B(T)$ and $w \notin A(T)$. If w is M-covered, extend T. Otherwise augment M and STOP if M is a perfect matching. Start a new tree $T = (r, \varnothing)$, where r is any M-exposed vertex. GOTO 2

   If there is no such edge vw then T is frustrated, so STOP, there is no perfect matching.

In the non-bipartite case, if G' is obtained from G by shrinking a sequence of odd cycles then G' has two types of vertices: *originial vertices* (vertices of G) and *pseudo vertices* (correponding to odd circuits). Given $vw, \in E$ with $v, w \in B(T)$, there is a unique circuit C of odd length in $T + vw$. If we shrink C then the resulting tree T' is also M-alternating and $C \in B(T')$. This leads to the general algorithm.

Algorithm for general graphs:

INPUT: graph G

OUTPUT: perfect matching M of G, or proof that there is no perfect matching

1. Set $M = \varnothing$ and $T = (r, \varnothing)$ (for any $r \in V$).
2. Suppose there is $vw \in E$ such that $v \in B(T)$.

   (a) If $w \in B(T)$ then shrink the unique odd circuit in $T + vw$.

   (b) If $w \notin B(T)$ and w is M-covered then extend T.

   (c) If $w \notin B(T)$ and w is M-exposed then augment M, unshrink all odd circuits, and expand M. STOP if M is perfect, otherwise start a new tree $T = (r, \varnothing)$, where r is M-exposed.

   GOTO 2

   If there is no such edge vw then T is frustrated, so STOP, there is no perfect matching.

To find a maximum cardinality matching, try to find a perfect matching in G. If we get a frustrated tree T with root v_1 in G' then delete $V(T_1)$ and get G_1. Repeat the process and construct disjoint trees $T_1, \dots, T_k$. Let $A = \bigcup_{i=1}^{k} A(T_i)$. Then $G \setminus A$ has k exposed vertices — one in each tree — and we have $\text{oc}(G \setminus A) - |A| = k$. This method gives an algorithmic proof of the Tutte-Berg formula.

4.3 Weighted matchings

Given $G = (V, E)$ and weights $c \in \mathbb{R}^E$, we would like to find a perfect matching M of G with $c(M)$ is minimized. First, consider the LP

$$\text{Minimize} \quad \textstyle\sum_{e \in E} c_e x_e \quad \text{s.t.}$$
$$x(\delta(v)) = 1 \text{ for every } v \in V, \text{ and } x \geq 0.$$

This is an LP relaxation since a $\{0,1\}$ solution corresponds to a solution to the minimum weight perfect matching. For non-bipartite graphs, this LP is not tight.

4.13 Theorem (Birkhoff). *If G is bipartite then there is an optimal solution to this LP which is integer.*

PROOF: The dual LP is

$$\begin{array}{c}\text{Maximize} \quad \sum_{v\in V} y_v \quad \text{s.t.}\\ y_v + y_w \le c_{vw} \text{ for all } vw \in E\end{array}$$

Complementary slackness implies that if $x_{vw} > 0$ then $y_v + y_w = c_{vw}$. Define the reduced costs $\overline{c}_{vw} = c_{vw} - y_v - y_w$. If M is a perfect matching of G and if there is $y \in \mathbb{R}^V$ such that $\overline{c} \ge 0$ and $c_e = 0$ for every $e \in M$ then M is minimum. (See text.) □

5 Matroids

5.1 Introduction

Missed the lecture defining matroids.

Let $G = (V, E)$ be a graph. $M = M(G)$ is defined as follows. $S = V$ and $I = \{J \subseteq V \mid \text{there is a matching which covers } J\}$.

5.1 Proposition. *M is a matroid.*

PROOF: We need to show that all bases of $A \subseteq S$ have the same cardinality. Suppose we have bases J_1 and J_2 of A and $|J_1| < |J_2|$. Then there is a matching M_1 that covers J_1 and a matching M_2 that covers J_2. Let $G' = (V, M_1 \triangle M_2)$, so that G consists of alternating circuits and alternating paths. Since $|J_1| < |J_2|$, there exists an alternating path P in G', and one end of P is in A and not covered by M_1, and the other end is either not in A or not covered by M_1. Then $M_1 \triangle P$ covers more vertices of A, which contradicts that J_1 is a basis. □

Greedy Algorithm:

INPUT: matroid $M = (S, I)$, weights $c \in \mathbb{R}^S$

OUTPUT: basis J of maximum weight

1. Set $J = \emptyset$.
2. While there is $e \in S \setminus J$ such that $J \cup \{e\} \in I$, choose such an e with c_e maximum and set $J \leftarrow J \cup \{e\}$.

5.2 Theorem. *For any matroid $M = (S, I)$ and $c \in \mathbb{R}^S$, the Greedy algorithm finds a maximum weight basis.*

PROOF: Clearly the Greedy algorithm finds a basis J. Suppose there is a basis J' such that $c(J') > c(J)$. Let $J = \{a_1, \ldots, a_m\}$, where the elements are in the order in which they we added by the Greedy algorithm. Let $J' = \{b_1, \ldots, b_m\}$, where $c_{b_1} \ge \cdots \ge c_{b_m}$. Let k be the smallest index such that $c_{b_k} > c_{a_k}$. Since we did not add any of $b_1, \ldots, b_k$ at this stage, either $b_i \in \{a_1, \ldots, a_{k-1}\}$ or $b_i \cup \{a_1, \ldots, a_{k-1}\} \notin I$, for all $i = 1, \ldots, k$. Therefore $\{a_1, \ldots, a_{k-1}\}$ is a basis of the set $\{a_1, \ldots, a_{k-1}, b_1, \ldots, b_k\}$. But $\{b_1, \ldots, b_k\}$ is also a basis for this set, and this is a contradiction since all bases must have the same cardinality. □

To find a maximum weight indepedent set, use the same algorithm, but only consider elements of weight at least 0.

5.2 Matroid Polytopes

Let (S, I) be a matroid with rank function r. Let $J \in I$ and let x^J be the characteristic vector of J. Then for all $A \subseteq S$, $x^J(A) \sum_{e\in A} x^J_e = |A \cap J| \le r(A)$. Consider the LP

$$\begin{array}{c} \text{Maximize} \quad \sum_{e\in S} c_e x_e \quad \text{s.t.} \\ x(A) \le r(A) \text{ for all } A \subseteq S,\ x \ge 0 \end{array}$$

The dual is

$$\begin{array}{c} \text{Minimize} \quad \sum_{A\subseteq S} y_A r(A) \quad \text{s.t.} \\ \sum_{A\subseteq S, e\in A} y_A \ge c_e \text{ for all } e \in S \\ y_A \ge 0 \text{ for all } A \subseteq S. \end{array}$$

The complementary slackness conditions are:

1. $x_e > 0$ implies $\sum_{A\subseteq S, e\in A} y_A = c_e$ for all $e \in S$
2. $y_A > 0$ implies $x(A) = r(A)$ for all $A \subseteq S$

5.3 Proposition. *This LP relaxation for the problem of finding the maximum weight independent set is tight.*

PROOF: Let J be a maximum weight independent set obtained by the Greedy algorithm. Order the elements of S by decreasing cost, *i.e.* $S = \{e_1 \ge \cdots \ge e_m > 0 \ge e_{m+1} \ge \cdots \ge e_n\}$. Set $S_i = \{e_1, \dots, e_i\}$. Let

$$y_A = \begin{cases} c_{e_i} - c_{e_{i+1}} & \text{if } A = S_i,\ 1 \le i \le m-1 \\ c_{e_m} & \text{if } A = S_m \\ 0 & \text{otherwise} \end{cases}$$

Clearly $y \ge 0$. We need to show that $\sum_{A\subseteq S, e_j \in S} y_A \ge c_{e_j}$. If $j > m$ then for every A with $e_j \in A$, $y_A = 0$ and the inequality holds. If $j \le m$ then

$$\sum_{A\subseteq S, e_j\in S} y_A = \sum_{i=j}^{m} y_{S_i} = \sum_{i=j}^{m-1} y_{S_i} + y_{S_m} = c_{e_j}$$

so y is feasible for the dual. The complementary slackness conditions give $x_{e_j} > 0$ implies $\sum_{A\subseteq S, e_j\in S} y_A = c_{e_j}$. If $j < m$ then the dual constraint is satisfied with equality. If $j > m$ then $x_{e_j} = 0$. See text. □

5.3 Matriod Intersection

Given matroids $M_1 = (S, I_1)$ and $M_2 = (S, I_2)$ with rank functions r_1 and r_2, find a maximum cardinality common independent set. Let $J \in I_1 \cap I_2$ and consider any $A \subseteq S$. $|J| = |J \cap A| + |J \cap \bar{A}|$, so $J \cap A \in I_1$ implies $|J \cap A| \le r_1(A)$ and $J \cap \bar{A} \in I_2$ implies $|J \cap \bar{A}| \le r_2(A)$. Therefore $|J| \le r_1(A) + r_2(\bar{A})$ for every $A \subseteq S$. Is there a pair (J, A) such that equality holds?

5.4 Theorem (Matroid Intersection). $\max\{|J| : J \in I_1 \cap I_2\} = \min\{r_1(A) + r_2(\bar{A}) : A \subseteq S\}$

PROOF: A functions $r : 2^S \to \mathbb{R}$ is said to be *submodular* if, for every $X, Y \subseteq S$, we have $r(X) + r(Y) \ge r(X \cup Y) + r(X \cap Y)$. The rank function on any matroid is submodular since for any $X, Y \subseteq S$, of B is a basis of $X \cap Y$ then we may extend it to a basis Q of $X \cup Y$.

$$r(X) + r(Y) \ge |Q \cap X| + |Q \cap Y| = |Q \cap (X \cup Y)| + |Q \cap (X \cap Y)| = |Q| + |B| = r(X \cup Y) + r(X \cap Y)$$

In fact, $r : 2^S \to \mathbb{R}$ defines the rank function of a matroid if

- $r(\varnothing) = 0$
- $r(A \cap \{e\}) = r(A)$ or $r(A) + 1$ for every $e \in S$
- r is submodular

Let $M = (S, I)$ be a matroid with rank function r and let $B \subseteq S$. $M \backslash B = (S \backslash B, I')$, where $I' = \{J \subseteq S \backslash B \mid J \in I\}$, is the matroid obtained by *deleting* B. It has the same rank function as M. (For a graphic matroid $M(G)$ and any $B \subseteq E(G)$, $M(G \setminus B) = M(G) \setminus B$, so this explains where the name comes from).

Let J_B be a basis for B. Define $M' = (S \setminus B, I')$, where $I' = \{J \subseteq S \setminus B \mid J \cup J_B \in I\}$. Then M' is a matroid, usually denoted M/B, called the *contraction* of B, with rank function $r'(A) = r(A \cup B) - r(B)$. Notice that r' really is a rank function of a matroid, so the definition of M/B is actually independent of the chosen basis of J_B. (Again, $M(G/B) = M(G)/B$ for a graphic matroid $M(G)$.) Check that M' is actually a matroid:

M0 $\varnothing \cup J_B \in I$, so $\varnothing \in I'$

M1 $J_1 \subseteq J_2 \in I'$, so $J_1 \cup J_B \subseteq J_1 \cup J_B \in I$. Thus $J_1 \in I'$.

M2 Let $A \subseteq S \backslash B$ and J' be a basis of A in M'. Then $J' \cup J_B$ is a basis of $A \cup B$ in M. (Clearly $J' \cup J_B$ is independent. If were not maximal then there would be $e \in S$ such that $\{e\} \cup J' \cup J_B$. But J_B is a basis of B, so $e \notin B$, and J' is a basis of A in M', so $e \notin A$, contradiction.) Therefore we have $r(A \cup B) = |J' \cup J_B| = |J'| + |J_B| = |J'| + r(B)$. This formula for $|J'|$ does not depend on J'.

Now, on to the proof of the theorem. Let $M_1 = (S, I_1)$ and $M_2 = (S, I_2)$, with rank functions r_1 and r_2. Let $k = \min\{r_1(A) + r_2(\overline{A}) \mid A \subseteq S\}$. We must show that there is $J \in I_1 \cap I_2$ such that $|J| = k$. If $S = \varnothing$ then $k = 0$ and we may take $J = \varnothing$. We may assume that there is $e \in S$ such that $\{e\} \in I_1 \cap I_2$. (If not then

$$S = \{e \in S \mid r_1(\{e\}) = 0\} \cup \{e \in S \mid r_2(\{e\}) = 0\}$$

and $k = 0$, so choose $J = \varnothing$.) Let $M_1' = M_1 \setminus e$ with rank function r_1', and $M_2' = M_2 \setminus e$ with rank function r_2'. By induction there is $J' \in I_1' \cap I_2'$ and $A \subseteq S \setminus \{e\}$ such that

$$|J'| = r_1'(A) + r_2'((S \setminus \{e\}) \setminus A) = r_1(A) + r_2((S \setminus A) \setminus \{e\})$$

If $|J'| = k$ then we are done, otherwise $r_1(A) + r_2((S \setminus A) \setminus \{e\}) \leq k - 1$.

Now let $M_1' = M_1/e$ and $M_2' = M_2/e$. By induction there is $J' \in I_1' \cap I_2'$ and $B \subseteq S \setminus \{e\}$ such that

$$|J'| = r_1'(B) + r_2'((S \setminus \{e\}) \setminus B) = r_1(B \cup \{e\}) + r_2(S \setminus B) - 2$$

since $\{e\} \in I_1 \cap I_2$. Notice that $J' \cup \{e\} \in I_1 \cap I_2$ by definition of contraction. If $|J' \cup \{e\}| = k$ we are done, otherwise we may assume that $|J'| \leq k - 2$, or $r_1(B \cup \{e\}) + r_2(S \setminus B) \leq k$. Adding the inequalities we have obtained for A and B,

$$\begin{aligned} 2k - 1 &\geq r_1(A) + r_1(B \cup \{e\}) + r_2((S \setminus A) \setminus \{e\}) + r_2(S \setminus B) \\ &\geq r_1(A \cap B) + r_1(A \cup B \cup \{e\}) + r_2(S \setminus (A \cap B)) + r_2(S \setminus (A \cup B \cup \{e\}) \\ &\geq k + k \end{aligned}$$

This contradiction proves the theorem. □

5.5 Definition. Let Π be a partition $\{X_1, \ldots, x_k\}$ of S. A *partition matroid* is $M = (S, I)$ where $I = \{J \subseteq S \mid |J \cap X_i| \leq 1, i = 1, \ldots, k\}$. M is a matriod with rank function $r(A) = |\{i \mid A \cap X_i \neq \varnothing\}|$.

5.6 Example. Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. For each vertex set V_i, we get a partition matrix M_i on the edges of G by taking $\Pi_i = \{\delta(v) \mid v \in V_i\}$. Notice that every $J \in I_1 \cap I_2$ is a matching of G. Let $A \subseteq E$ and consider that

$$\{v \in V_1 \mid A \cap \delta(v) \neq \varnothing\} \cup \{v \in V_2 \mid \overline{A} \cap \delta(v) \neq \varnothing\}$$

is a vertex cover of G. By the Matroid Intersection Theorem there is a matching $J \in I_1 \cap I_2$ and $A \subseteq E$ such that $|J| = r_1(A) + r_2(\overline{A})$. This is König's Theorem.

5.7 Example (Arborescence). Let $G=(V,E)$ be a digraph, $M_1=(E,I_1)$ be the graph matroid of G (omitting the directions on the edges), and $M_2=(E,I_2)$ be the partition matroid given by $\Pi=\{\delta(\overline{v}) \mid v\in V\}$. Let $T\in I_1\cap I_2$, so T is a (undirected) tree, and at most one arc of T enters every vertex of G. T is an *arborescence*. The Matrix Intersection Theorem tells us somethings about arborescences.

A Totally Unimodular Matrics

A.1 Theorem (Cramer's Rule). *Let A be an $n\times n$ non-singular matrix and let x be a solution to $Ax=b$. Then*

$$x_i=\frac{\det(A\leftarrow_i b)}{\det(A)}$$

where $A\leftarrow_i b$ is the matrix obtained from A by replacing column i with b.

A.2 Proposition. *Let A be an integer $n\times n$ non-singular matrix. Then* $\det(A)=\pm1$ *if and only if $A^{-1}b$ is integer for all integer vectors b.*

PROOF: Suppose that $\det(A)=\pm1$. Then for any integer vector b, Cramer's Rule implies that the entries of $A^{-1}b$ are of the form $\pm\det(A\leftarrow_i b)$. Conversely, suppose that $A^{-1}b$ is integer for every integer vector b. Then $A^{-1}e_i$ is integer, so all entries of A^{-1} are integer. Hence $\det(A^{-1})$ is integer, and $\det(A)\det(A^{-1})=1$, so $\det(A)=\det(A^{-1})=\pm1$. □

A.3 Definition. A matrix A is *totally unimodular* (TU) if all its square submatrices have determinant 0, 1, or -1. As a consequence, A must be a $\{0,\pm1\}$ matrix.

A linear programming problem can always be formulated as

$$\text{Maximize} \quad c^Tx \quad \text{s.t.} \\ Ax=b,\ x\geq 0$$

where A is an $m\times n$ matrix. We may assume that the rows of A are linearly independent. Given a subset $J\subseteq\{1,\dots,n\}$, define A_J as the column submatrix of A indexed by J.

$B\subseteq\{1,\dots,n\}$ is a *basis* if A_B is non-singular. Let B be a basis and $N=\{1,\dots,n\}\setminus B$. We can write $A=[A_B\mid A_N]$, $c^T=[c_B^T\mid c_N^T]$, and $x^T=[x_B^T\mid x_N^T]$ A feasible solution x^* is *basic* (for a basis B) if $x_N^*=0$. In this case $Ax^*=b$, so $x_B^*=A_B^{-1}b$.

A.4 Theorem (Basic Solution). *If an LP problem has a solution then it has an optimal solution which is basic.*

A.5 Proposition. *If an LP problem has an optimal solution where A is TU and b is integer then it has an optimal solution which is integer.*

PROOF: Any basic optimal solution is integer. □

A.6 Proposition. *Let A be a* $\{0,\pm1\}$ *matrix, where each column has at most one* $+1$ *entry and at most one* -1 *entry. Then A is TU.*

PROOF: By induction on the size of the $k\times k$ submatrix N under consideration. If $k=1$ then $\det(N)=0,\pm1$, as all entries of A are 0, 1, or -1. Suppose $k>0$ and $\det(N)=0,\pm1$ for all smaller k. One of the following cases must occur:

1. N has a zero column, so $\det(N)=0$.

2. all columns of N have exactly one $+1$ and one -1, so the sum of all the rows is zero and $\det(N) = 0$.
3. Some column of N has exactly one non-zero entry.

In the final case, we may assume without loss of generality that the non-zero entry occurs in the upper left corner. Then $\det(N) = N_{1,1} \det(N') = 0, \pm 1$ by the determinant formula. □

A.7 Definition. Let $G = (V, E)$ be a digraph. The *incidence matrix* A of G is defined to have rows corresponding to the nodes of G and columns corresponding to the arcs of G. An entry

$$A_{ve} = \begin{cases} +1 & \text{if } v \text{ is the head of } e \\ -1 & \text{if } v \text{ is the tail of } e \\ 0 & \text{otherwise} \end{cases}$$

These ideas can be used to give another proof of the MFMC theorem.