Network Design Foundation
Fall 2015
Iterative Methods in Combinatorial Optimization

Instructor: Mohammad T. Hajiaghayi
Scribe: Saba Ahmadi, Saurabh Kumar

December 10, 2015

1 Preliminaries

1.1 Linear Algebra

In this section, we review some definitions and concepts related to linear algebra which will be useful in describing the iterative methods later.

Definition 1 *The* **row rank** *of a matrix A is the maximum number of linearly independent rows in A. The* **column rank** *of a matrix A is the maximum number of linearly independent columns in A. In other words,*

$$\textit{row rank of } A = dim(span(\textit{row vectors of } A))$$
$$\textit{column rank of } A = dim(span(\textit{column vectors of } A))$$

Theorem 1 *The row rank and column rank of any given matrix are equal and therefore unambiguously referred to as the* **rank** *of the matrix.*

The general form of a linear program is as follows (this will be referenced in the definitions and lemmas given ahead):

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. }\ & Ax \geq b \qquad\qquad\qquad \ldots P \\ & x \geq 0 \end{aligned}$$

Here, P corresponds to the set of feasible solutions (the ones that satisfy both given constraints) i.e $P = \{x : Ax \geq b, x \geq 0\}$.

Definition 2 *$x \in \mathbb{R}^n$ is an* **extreme point solution** *to P if there is no non-zero $y \in \mathbb{R}^n$ s.t $x + y$ and $x - y$ belong to P. Alternately, x cannot be written as a linear combination of any two $y, z \in P$ (i.e x cannot be written as $ay + (1 - a)z$).*

Intuitively, if x is an extreme point solution to P, it is located at a corner point of P. It can be proved that a bounded LP (one with a finite optimum) always admits an extreme point solution that achieves the optimum.

Lemma 1 (Rank Lemma) *Let x be an extreme point solution of P such that $x_i > 0$ for each i. Then any maximal number of linearly independent constraints that are tight at x (i.e constraints of the form*[1] *$A_i x = b_i$) equal the number of variables in x.*

1.2 Intuition behind Iterative Methods

For NP hard problems, in order to obtain good approximation algorithms, we will be concerned with two components of iterative methods - iterative rounding and iterative relaxation.

1.2.1 Iterative Rounding

1. Formulate the NP hard problem as a linear program
2. Fix a threshold and argue the existence of an element greater than the threshold in the extreme point solution
3. Include that element in solution by rounding up
4. Modify the constraints of the problem to reflect the residual problem
5. Repeat from step 2 until no more constraints remain

1.2.2 Iterative Relaxation

1. Formulate the NP hard problem as an LP with integral extreme points
2. At each step, argue existence of integral element to include in solution or a constraint (with "low violation") to relax
3. When all constraints are relaxed, the remaining solution is integral and has low violation

The following scribe notes are based on [3] and [2].

2 The Assignment Problem

Problem Definition

Given a complete bipartite graph $G = (A \cup B, A \times B)$ where $|A| = |B| = n$ and costs c_{ij} on edge (i, j) with $i \in A$, $j \in B$. Our aim is to match each vertex in A with a distinct vertex in B in such a way that the total cost of the matching is minimized.

[1] A_i means the ith constraint in A

Approach

We describe an LP which solves the above problem and use iterative relaxation with a counting argument to obtain a solution. The LP (after standard relaxation from an ILP) can be formulated as:

$$
\begin{aligned}
\min \sum_{i,j} & c_{ij} x_{ij} & \\
\text{s.t.} \sum_{i \in A} & x_{ij} \geq 1 & \forall\ j \in B \\
\sum_{i \in B} & x_{ij} \geq 1 & \forall\ i \in A \\
& x_{ij} \geq 0 & \forall\ (i,j) \in A \times B
\end{aligned}
$$

Here, x_{ij} are indicator variables indicating whether or not edge (i,j) is part of the solution matching. c_{ij} denotes the cost of the edge (i,j).

While generally, an LP need not have an integral solution, it turns out that this LP does. This can be stated in the following theorem.

Theorem 2 *Extreme points x^* of the above relaxed LP are integral.*

Proof: We make the following claim which we prove subsequently - at any extreme point for the above LP, there exists an edge with an x^* value of 1. Assuming the above claim, we then use the following algorithm to obtain our solution:

1. Solve the above fractional LP
2. If any $x^*_{ij} = 0$, remove that edge to obtain a smaller problem.
3. If any $x^*_{ij} = 1$, we take this pair as part of our final solution and delete it (both nodes and the edge) to obtain a smaller problem.
4. Repeat procedure until no edges remain.

In each iteration, because we either choose an edge with 0 or 1 weight, we pay exactly what the fractional solution would pay and therefore, at the end of the above procedure, the cost of our (integral) solution is exactly that of the (potentially) fractional one i.e our integral solution is optimal (this can be proved rigorously by induction as well). ■

Finally, we now prove the claim we made in the previous proof.

Claim 1 *At any extreme point x^*, there exists an edge (i,j) such that $x^*_{ij} = 1$ i.e there exists a 1-edge.*

Proof: At any given iteration of the above algorithm, construct a support graph $G_s = (A' \times B', E_s)$ where E_s contains only edges with non-zero values at the extreme point (and which have not been deleted in previous iterations). Let $|A'| = |B'| = n'$. If possible, suppose that there is no 1-edge.

From our assumption, for all edges $(i, j) \in E_s$, we have $0 < x_{ij} < 1$. This implies that each vertex i in A' must have at least two edges incident on it since $\sum_{j\in B} x_{ij} \geq 1$ (as per the LP constraint) and $x_{ij} < 1 \ \ \forall\, j \in B$. Therefore, the graph contains at least $2n'$ edges. In other words, The number of variables is at least $2n'$.

On the other hand, in the LP, we have $2n'$ constraints (one for each vertex in $A \cup B$). However, the sum of the constraints for all vertices in A equals the sum of the constraints for all vertices in B. More formally,

$$\sum_{i\in A}\sum_{j\in B} x_{ij} = \sum_{j\in B}\sum_{i\in A} x_{ij}$$

Therefore, the system of constraints is linearly dependent and hence the number of linearly independent constraints is at most $2n - 1$.

However, by the rank lemma, at an extreme point, the number of variables equals the number of tight linearly independent constraints. This is a contradiction. Therefore, there must exist a 1-edge in the extreme point. ■

3 Extension: Generalized Assignment

In this problem, we are given a set of jobs J and machines M, for each job $j \in J$ and machine $i \in M$ there is a processing time p_{ij} and cost c_{ij}.
The goal is to assign each job to some machine such that the total cost is minimized and no machine is scheduled for more than P time.

Approach:
We describe an LP which solves this problem and use iterative relaxation with a counting argument to obtain a solution. The LP formulation is as following:

$$\begin{aligned}
&min \sum_{i,j} c_{ij} x_{ij} && \\
&\sum_{i\in M} x_{ij} \geq 1 && \forall j \in J \\
&\sum_{j\in J} p_{ij} x_{ij} \leq P && \forall i \in M \\
&x_{ij} \geq 0 && \forall i, j
\end{aligned}$$

Here, x_{ij} are indicator variables indicating whether or not an edge (i, j) is part of the solution.

Preparation:
In the bipartite graph, we could prune edges with $p_{ij} > P$, because they can

never be used in a feasible solution of LP.
If in the optimal solution of LP for some edge e, $x_e = 1$, then include this assignment in the solution. Also delete the job and decrease makespan constraint for the machine.
If in the optimal solution of LP for some edge e, $x_e = 0$, remove the edge to obtain a smaller problem.

Relaxation:
If there is a machine with degree 1 in the support then we could remove its makespan constraint, since the single job using it fractionally cannot have $p_{ij} > P$, so final makespan of the machine is at most P.
If there is a machine with degree 2 in the support then remove its makespan constraint, now final makespan of the machine is at most $2 * P$.

Lemma 2 *For any extreme point solution x' to fractional LP, either there is a makespan constraint to relax, or there is an edge like e, such that $x_e \in \{0, 1\}$.*

Proof: Suppose for contradiction there is no edge like e, such that $x_e \in \{0, 1\}$. Also there is no machine with degree 1 or 2. Thus degree of jobs is at least 2 and degree of machines is at least 3.
Thus $|E| \geq \frac{2|J|+3|M|}{2}$. But number of tight constraints is at most $|J| + |M|$. Thus number of edges is greater than number of tight constraints which is a contradiction of the rank lemma.
■

Apply the lemma repeatedly by either relaxing a makespan constraint or deleting an edge which has value 0 or 1, then resolve the residual problem. Finally the solution will have optimal cost and makespan at most $2 * P$.

4 Minimum Spanning Trees

Spanning Tree Polyhedron

$$min \sum_{e \in E} c_e x_e$$

$$s.t \sum_{e \in E(V)} x_e = |V| - 1 \qquad \text{Any tree has } n-1 \text{ edges.}$$

$$s.t \sum_{e \in E(S)} x_e \leq |S| - 1 \qquad \forall S \subset V, \text{ Subtour Elimination}$$

$$x_e \geq 0 \qquad \forall e \in E$$

$E(S)$ shows set of edges with both endpoints in S.

Uncrossing Technique For a set $F \subset E$, let $\chi(F)$ denote the characteristic vector in $R^{|E|}$ that has a 1 corresponding to each edge $e \in F$ and 0 otherwise.

Proposition 1 *For $X, Y \subset V$,*
$\chi(E(X)) + \chi(E(Y)) \leq \chi(E(X \cup Y)) + \chi(E(X \cap Y))$

Proof: Observe that,
$\chi(E(X)) + \chi(E(Y)) = \chi(E(X \cup Y)) + \chi(E(X \cap Y)) - \chi(E(X \setminus Y, Y \setminus X))$
So the proposition follows.
■

Let $F = \{S \subset V | \chi(E(S)) = |S| - 1\}$ be the family of tight constraints.

Lemma 3 *If $S, T \in F$ and $S \cap T \neq \emptyset$, then both $S \cap T$ and $S \cup T$ are in F. Also, $\chi(E(S)) + \chi(E(T)) = \chi(E(S \cap T)) + \chi(E(S \cup T))$.*

Proof:

$\lvert S\rvert - 1 + \lvert T\rvert - 1 = \chi(E(S)) + \chi(E(T))$	$S, T \in F$
$\leq \chi(E(S \cap T)) + \chi(E(S \cup T))$	Proposition. 1
$\leq \lvert S \cap T\rvert - 1 + \lvert S \cup T\rvert - 1$	according to the LP costraints for $S \cap T, S \cup T$
$= \lvert S\rvert - 1 + \lvert T\rvert - 1$	$\lvert S \cap T\rvert + \lvert S \cup T\rvert = \lvert S\rvert + \lvert T\rvert$

Now we could conclude that $\chi(E(S \cap T)) + \chi(E(S \cup T)) = |S \cap T| - 1 + |S \cup T| - 1$. Thus $\chi(E(S \cap T)) = |S \cap T| - 1$ and $\chi(E(S \cup T)) = |S \cup T| - 1$, which means that both $S \cap T$ and $S \cup T$ are in F. Also $\chi(E(S)) + \chi(E(T)) = \chi(E(S \cap T)) + \chi(E(S \cup T))$.
■
Following lemma is based on uncrossing arguments.

Lemma 4 *Independent set of tight constraints defining an extreme point can be chosen s.t. corresponding subset of vertices form a Laminar family L. [Cornuejols et. al. '88]*

Iterative Algorithm: While G is not a singleton, solve LP to obtain extreme point x^*. Remove all edges s.t ${x^*}_e = 0$.
Contract all edges s.t ${x^*}_e = 1$.

Lemma 5 *Support E of any extreme point x^* of the LP has an edge with x^*-value 1.*

Proof: There are $|L|$ tight constraints in the form $\sum_{e \in E(S)} x_e = |S| - 1$, thus according to the rank lemma $|E| = |L|$.
Assign one token per edge in E. For each edge e, collect x_e fractional token from the smallest tight set containing both endpoints of e, then we show some extra leftover tokens which implies that $|E| > |L|$ giving us a contradiction of rank lemma.
Let S be any set in L, 2 cases might happen:
1- S is a leaf node, then: $\chi(S) = |S| - 1 \geq 1$.

2- S has children $c_1, c_2, ..., c_k$ for some $k \geq 0$, we have that:
$\chi(S) - \sum_{1 \leq i \leq k} \chi(c_i) = |S| - \sum_{1 \leq i \leq k} |c_i| + k - 1 = \chi(A)$
where $A = E(S) \setminus (\cup_i E(c_i))$. If $A = 0$, then $\chi(S) = \sum_{1 \leq i \leq k} \chi(c_i)$, then these tight constraints in L are not independent which is a contradiction. Also $\chi(A)$ is an integer and is at least one, giving S at least one token.
Since for all edges like e, x_e is fractional, now every edge has $1 - x_e$ leftover, which is a contradiction.
■

5 Minimum Degree Bounded Spanning Trees

5.1 Problem Definition

Given a graph $G = (V, E)$, edge costs c_e for edge $e \in E$ and an upper bound B_v on the degree for each vertex $v \in V$. Our aim is to find a minimum spanning tree which satisfies the degree bounds.

5.2 Approach

Here, we consider a more general case where degree bounds are specified for a subset $W \subseteq V$. For $S \subseteq V$, let $E(S)$ be the set of edges with both endpoints in S and let $\delta(S)$ be the set of edges with exactly one endpoint in S. Also, for $E' \subseteq E$, define as shorthand $x(E') = \sum_{e \in E'} x_e$. We formulate the relaxed LP for this problem as follows:

$$\min \sum_{e \in E} c_e x_e \tag{5.1}$$

$$\text{s.t. } x(E(V)) = |V| - 1 \tag{5.2}$$

$$x(E(S)) \leq |S| - 1 \qquad \forall\ S \subset V, S \neq \varnothing \tag{5.3}$$

$$x(\delta(v)) \leq B_v \qquad \forall\ v \in W \tag{5.4}$$

$$x_e \geq 0 \qquad \forall\ e \in E \tag{5.5}$$

The constraint formulation of this problem is related to the traveling salesman problem and subtour elimination. A spanning tree of vertices in set S has at most $|S| - 1$ edges (any more and we would have a cycle). Constraint (5.2) eliminates any subtours (cycles) that could have formed in the overall MST. Constraint (5.3) eliminates any subtour that could have formed in $E(S)$ for all proper subsets S of V. Constraint (5.4) is our degree bound (Note that $\delta(v)$ denotes the set of edges incident on v).

In the above LP, the number of constraints is exponential (due to constraint set (5.3)), but we can devise a polynomial time separating oracle for testing feasibility using min cut. This allows us to solve the LP in polynomial time. Another approach is to recast the above LP using a polynomial number of constraints which is also possible. Either way, we can solve the above LP in linear time.

For the following discussion, remember that an extreme point solution is the unique solution defined by n lineary independent tight inequalities, where n is the number of variables in the linear program. Also, an extreme point solution is characterized by tight inequalities whose corresponding sets form a laminar family.

Lemma 6 *Let x^* be an extreme point solution to the above LP with $x_e > 0$ for each $e \in E$. For a set $F \subseteq E$, let $\chi(F)$ denote the characteristic vector in $\mathbb{R}^{|E|}$ that has a 1 for each edge $e \in F$ and 0 otherwise. Then, there exists a set $T \subseteq W$ and a laminar family $\mathcal{L}$ such that:*

1. *$x(\delta(v)) = B_v$ for each $v \in T$ and $x(E(S)) = |S| - 1$ for each $S \in \mathcal{L}$*
2. *The characteristic vectors in $\{\chi(E(S)) : S \in \mathcal{L}\} \cup \{\chi(\delta(v)) : v \in T\}$ are linearly independent.*
3. *$|\mathcal{L}| + |T| = |E|$*

1 Approximation Algorithm There exists an additive one approximation to the above LP (this algorithm and its analysis is based on [2]). In it, we do not perform any rounding at all, we simply relax vertex degree constraints one by one to obtain the final solution. The algorithm is:

1. While $W \neq \varnothing$ do
   (a) Solve the LP to find an optimal extreme point solution x^* (note that W may have been updated from a previous iteration).
   (b) Remove all edges e such that $x^*_e = 0$. Let E be the support of x^*.
   (c) **(Relaxation)** If there exists a vertex $v \in W$ such that $\deg_E(v) \leq B_v + 1$, remove v from W (remove that vertex's degree bound) i.e set $W \leftarrow W - \{v\}$.
2. Return E

Lemma 7 *The above algorithm returns a spanning tree*

Proof: At each step we remove a vertex bound constraint (we prove that we can do this in the subsequent lemmas). Once the degree constraints are removed, the LP is identical to the one for the standard MST problem which is known to be integral. Therefore the algorithm returns a tree. ■

Lemma 8 *Let x^* be the an extreme point solution to the above LP with $x_e > 0$. Let $\mathcal{L}$ and $T \subseteq W$ be the tight set constraints and the tight degree constraints defining x (based on the Lemma 6). If $T \neq \varnothing$, there exists a vertex $v \in W$ with* $\deg_E(v) \leq B_v + 1$.

Proof: We can use a fractional token counting argument to prove this lemma. If possible, suppose that $T \neq \varnothing$ and for each $v \in W$, we have that $\deg_E(v) > B_v + 2$. Assign one token to each edge in E. We perform the following token redistribution: each edge $e \in E$ gives:

1. x_e fraction of its token to the smallest set in the laminar family which contains both the endpoints of e.

2. $(1-x_e)/2$ fraction of its token to each endpoint for the degree constraints.

Now, let S be any set in $\mathcal{L}$ with children $S_1, S_2, ...S_k$ for $k \geq 0$. Then, since our laminar family is tight, we must have that $x(E(S)) = |S| - 1$ and $x(E(S_i)) = |S_i| - 1$ for each i. If we subtract these expressions, we obtain,

$$x(E(S)) - \sum_{i=1}^{k} x(E(S_i)) = |S| - \sum_{i=1}^{k} |S_i| + k - 1$$

Let $A = E(S) - \bigcup_i E(S_i)$. In other words, A is the set of edges where both endpoints are in S but not in any child of S. Therefore, we have:

$$x(A) = x(E(S)) - \sum_{i=1}^{k} x(E(S_i)) = |S| - \sum_{i=1}^{k} |S_i| + k_1 \tag{5.6}$$

Now, by the way we defined the redistribution, we are giving x_e fraction of a token to the smallest set in the laminar family containing both endpoints of e. In other words, S obtains token fraction x_e for each edge in A (since edges A are not contained in any smaller subset) i.e S obtains $x(A)$ token (by definition of $x(A)$). Since from the above $x(A)$ is integral (Equation 5.6), it is at least one, and so each set in the laminar family gets at least one token. Note that if $A = \varnothing$, then $\chi(E(S)) = \sum_i \chi(E(S_i))$ which contradicts linear independence of the sets of these constraints in $\mathcal{L}$.

For a vertex $v \in T$, by our redistribution scheme, it receives fraction $(1 - x_e)/2$ token for each edge e incident on it. This gives a total value of:

$$\begin{aligned}
\sum_{e\in\delta(v)} \frac{1-x_e}{2} &= \sum_{e\in\delta(v)} \frac{1}{2} - \sum_{e\in\delta(v)} \frac{x_e}{2} \\
&= \frac{\deg_E(v)}{2} - \frac{x(v)}{2} \\
&= \frac{\deg_E(V) - B_v}{2} && \ldots \text{ since the degree constraints are tight for } T \\
&\geq 1 && \ldots \text{ initial assumption: } \deg_E(v) > B_v + 2 \text{ for } v \in T
\end{aligned}$$

Therefore, each vertex gets a token value of at least 1.

So far, we have proved the existence of $|\mathcal{L}|$ tokens (from each set in the laminar family) and an additional $|T|$ tokens (from each vertex in T). If we can show that some tokens are left over, we will have contradicted the statement that $|E| = |T| + |\mathcal{K}|$.

If $V \notin \mathcal{L}$, then there exists an edge e which is not contained in any set of the laminar family. Therefore, this edge's token x_e fraction has not yet been counted (since previously, we only counted edges fully contained in a minimal set). Hence, our total token count has now exceeded $|\mathcal{L}|+|T|$ and therefore $|E|$. Contradiction.

Therefore, henceforth, we assume that $V \in \mathcal{L}$. Also, since e is a tight set of two vertices, we have $e \in span(\mathcal{L})$ for each e such that $x_e = 1$ (since therefore, the set corresponding to this edge has one token).

In a similar vein, if there is a vertex $v \in W - T$, then it has an unnecessary token and this again gives us a contradiction similar to the previous. Furthermore, if there exists a vertex $v \in V - T$, then each edge e incident on it either has $x_e < 1$, meaning that v has a fraction $(1 - x_e)/2$ which is extra (giving the contradiction), or $x_e = 1$.

We have,

$$\begin{aligned} 2\chi(E(V)) &= \sum_{v \in V} \chi(\delta(v)) \\ &= \sum_{v \in T} \chi(\delta(v)) + \sum_{v \in V-T} \chi(\delta(v)) \\ &= \sum_{v \in T} \chi(\delta(v)) + \sum_{v \in V-T} \sum_{e \in \delta(v)} \chi(e) \end{aligned}$$

By our previous assumption, $V \in \mathcal{L}$ and $e \in span(\mathcal{L})$ for each edge $e \in \delta(v)$ for $v \in V - T$. We also have $T \neq \varnothing$ and so the above implies the linear independence of the tight constraints in T and $\mathcal{L}$. Contradiction.

■

6 SNDP

We consider a general version of the problem as follows: we are given a graph $G = (V, E)$ where each edge e has a non-negative cost c_e and a *requirement function* $f : 2^V \to \mathcal{Z}$. Our aim is to find a minimum cost subgraph such that for each $S \subseteq V$, the number of edges crossing the cut defined by S is greater than or equal to the requirement of S.

We can write the LP for this problem as:

$$\begin{aligned} \text{Min } & \sum_{e \in E} c_e x_e \\ \text{s.t. } & x(\delta(S)) \geq f(S) && \ldots \forall S \subseteq V \\ & 1 \geq x_e \geq 0 \end{aligned}$$

Now, we introduce two types of functions - skew supermodular functions and strongly submodular functions.

Definition 3 (Skew Supermodular Function) *A function g is skew supermodular iff at least one of the following holds:*

$$g(A \cup B) + g(A \cap B) \geq g(A) + g(B)$$

$$g(A-B)+g(B-A) \geq g(A)+g(B)$$

Definition 4 (Strongly Submodular Function) *A function h is strongly submodular iff* both *of the following hold:*

$$g(A \cup B)+g(A \cap B) \leq g(A)+g(B)$$
$$g(A-B)+g(B-A) \leq g(A)+g(B)$$

Note that if g is skew supermodular and h is strongly submodular, then $g-h$ is also skew supermodular.

It can be shown that the function $x(\delta(S))$ (which denotes the sum of the x_e variables for each edge e crossing the cut i.e the edge boundary) is strongly submodular.

6.1 2 Approximation for SNDP

Jain gave 2 approximation for SNDP when the requirement function f is weakly supermodular as follows:

1. Initialize $F = \varnothing$, $f' = f$
2. While $f' \neq \varnothing$
   (a) Solve the above LP and obtain an extreme point x^*
   (b) Remove any edge e for which $x_e = 0$
   (c) If there exists an edge e such that $x_e \geq 1/2$, add e to F and for each subset $S \subseteq V$, set $f'(S) = f'(S) - |e \cap \delta(S)|$
3. Return $H = (V, F)$

Note that in step 2, the condition $f' \neq \varnothing$ means that there exists at least one set $S \subseteq V$ such that $f'(S) \neq 0$. In step 2.c, we will prove later that an edge e with $x_e \geq 1/2$ can always be found. Once found, we add e to F and reduce the requirement for each subset S for which e is a crossing edge.

Lemma 9 *We can choose an independent set of tight constraints uniquely defining x^* in such a way that the subsets corresponding to each tight constraint form a laminar family (Note that a tight constraint corresponds to a set S satisfying $x(\delta(S)) = f(S)$).*

Theorem 3 (Jain [1]) *Any extreme point of the above LP relaxation has an edge e with $x_e \geq 1/2$.*

Proof: The key idea in the proof is that the sets corresponding to the independent tight constraints form a laminar family $\mathcal{L}$. We use a fractional token redistribution argument.

If possible, assume that for all $e \in E$, $x_e < 1/2$.

Our distribution scheme is as follows:

1. Assign 1 token to each edge $(u, v) \in E$
2. Assign token fraction x_e to the smallest set containing u
3. Assign token fraction x_e to the smallest set containing v
4. Assign the remaining $1-2x_e$ token to the smallest set which contains both u and v

From our assumption $x_e < 1/2$ and so none of the token fractions above are negative.

First we show that every set in the $\mathcal{L}$ receives at least one token.

Consider a set S with the children R_1, R_2, ..., R_k in the laminar family. Note that each set corresponds to a tight constraint. We define three sets of edges A, B and C. Let S' denote the set of vertices in S that are not in any child of S. A contains edges with one vertex in S' and the other vertex outside S. B contains edges with one edge in a child and the other in S'. C contains edges with one vertex in a child and the other in another child.

Then, we have the following:

$$x(\delta(S)) - \sum_j x(\delta(R_j)) = f(S) - \sum_j f(R_j) \qquad \ldots \text{ sets have tight constraints}$$

$$x(\delta(S)) - \sum_j x(\delta(R_j)) = x(A) - x(B) - 2x(C) \qquad \ldots \text{ by a counting argument}$$

Note that we must have $x(\delta(S)) - \sum_j x(\delta(R_j)) \neq 0$ since otherwise, the constraints would be linearly dependent. Therefore, $x(A) - x(B) - 2x(C) \neq 0$ and must be integral (since it equals $f(S) - \sum_j f(R_j)$ which is integral). Therefore, $x(A) - x(B) - 2x(C)$ is at least 1.

Now, the tokens assigned to S are given by:

$$\begin{aligned}
&\sum_{e \in A} x_e + \sum_{e \in B} (1 - x_e) + \sum_{e \in C} (1 - 2x_e) \\
=& x(A) + |B| - x(B) + |C| - 2x(C) \\
=& |B| + |C| + x(A) - x(B) - 2x(C) \\
\geq& |B| + |C| + 1
\end{aligned}$$

So each set in $\mathcal{L}$ receives at least one token. Therefore the number of tokens is at least $|\mathcal{L}|$.

Consider a maximal set M in $\mathcal{L}$. It has at least one edge (u, v) leaving it. No set in $\mathcal{L}$ contains one of the vertices of the edge. So a fraction $1 - 2x_{uv}$ is still unassigned. Therefore, the total tokens $> |\mathcal{L}|$. But by the rank lemma, we must have $|E| = |\mathcal{L}|$ and initially one token was assigned per edge for a total of $|E|$ tokens. Contradiction. ∎

Therefore an edge with $x_e \geq 1/2$ can always be found. This gives us a 2 approximation to this problem.

SNDP Suppose we are given a graph $G = (V, E)$ where each edge e has cost c_e. We are also given a set of source and destination pairs $\{(s_i, t_i)\}$. Our aim is to find a minimum cost subgraph with r_i edge disjoint paths between s_i and t_i.

We can formulate the LP for this problem as follows:

$$\begin{aligned} \text{Min } & \sum_{e \in E} c_e x_e \\ \text{s.t. } & x(\delta(S)) \geq r_i \qquad \ldots \forall S \subseteq V \text{ separating } s_i \text{ and } t_i \\ & x_e \geq 0 \end{aligned}$$

where as usual, x_e indicates whether edge e is part of the final solution. $\delta(S)$ is the set of edges with exactly one end in S (i.e the edges crossing the graph cut induced by the partition S and $V - S$).

Suppose we define a function $f : 2^V \to \mathcal{Z}$ such that:

$$f(S) = \max_{(s_i, t_i) \text{ cut by } S} r_i$$

Then we can show that f is supermodular. So Jain's algorithm above gives a 2-approximation to SNDP.

References

[1] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[2] Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.

[3] R Ravi. Iterative methods in combinatorial optimization.