

# **CSC2420 Fall 2012: Algorithm Design, Analysis and Theory Lecture 6**

Allan Borodin

February 12, 2015

# Announcements and today's agenda

- Announcements

The first three questions for the second problem set were posted.

- I plan to add more questions before or on the weekend.

bf My plan did not materialize; I will be adding more questions at the end of reading week and changing the due date to March 5.

- Guest Lecture: On Thursday, Feb 26, the plan is to have Professor Derek Corneil discuss “graph search and orderings that characterize various graph classes”.

- Today's agenda

- 1 Begin linear programming and rounding

- 2 LP Duality

# Integer Programming (IP) and Linear Programming (LP)

- We now introduce what is both theoretically and in practice one of the most general frameworks for solving search and optimization problems. Namely, we consider how many problems can be formulated as integer programs (IP). (Later, we will also consider other mathematical programming formulations.)
- Solving an IP is in general an NP hard problem although there are various IP problems that can be solved optimally. Moreover, in practice, many large instances of IP do get solved.
- Our initial emphasis will be on linear program (LP) relaxations of IPs. LPs can be solved optimally in polynomial time as first shown by Khachiyan's ellipsoid method [1979] and then Karmarkar's [1984] more practical interior point method. In some (many?) cases, Danzig's [1947] simplex method will outperform (in terms of time) the worst case polynomial time methods.
- **Smoothed analysis** gives an explanation for the success of simplex.
- **Open: a strongly polynomial time algorithm for solving LPs?**

# Some IP and LP concepts

## Integer Programs

An IP has the following form:

- Maximize (minimize)  $\sum_j c_j x_j$
- subject to  $(\sum_j a_{ij} x_j) R_i b_i$  for  $i = 1, \dots, m$   
and where  $R_i$  can be  $=, \geq, \leq$
- $x_j$  is an integer (or in some prescribed set of integers) for all  $j$

Here we often assume that all parameters  $\{a_{ij}, c_j, b_i\}$  are integers or rationals but in general they can be real valued.

An LP has the same form except now the last condition is realized by letting the  $x_j$  be real valued. It can be shown that if an LP has only rational parameters then we can assume that the  $\{x_j\}$  will be rational.

# Canonical LP forms

Without loss of generality, LPs can be formulated as follows:

## Standard Form for an LP

- |   |  |
|---|--|
| • Maximize $\mathbf{c} \cdot \mathbf{x}$          | Minimize $\mathbf{c} \cdot \mathbf{x}$ |
| • subject to $A \cdot \mathbf{x} \leq \mathbf{b}$ | $A \cdot \mathbf{x} \geq \mathbf{b}$   |
| • $\mathbf{x} \geq 0$                             | $\mathbf{x} \geq 0$                    |

## Slack form

- maximize/minimize  $\mathbf{c} \cdot \mathbf{x}$
- subject to  $A \cdot \mathbf{x} + \mathbf{b} = \mathbf{s}$
- $\mathbf{x} \geq 0; \mathbf{s} \geq 0$

The  $\{s_j\}$  variables are called slack variables.

## LP relaxation and rounding

- One standard way to use IP/LP formulations is to start with an IP representation of the problem and then relax the integer constraints on the  $x_j$  variables to be real (but again rational will suffice for our interests) variables.
- We start with the well known simple example for the weighted vertex cover problem. Let the input be a graph  $G = (V, E)$  with a weight function  $w : V \rightarrow \mathbb{R}^{\geq 0}$ . To simplify notation let the vertices be  $\{1, 2, \dots, n\}$ . Then we want to solve the following “natural IP representation” of the problem:
  - ▶ Minimize  $\mathbf{w} \cdot \mathbf{x}$
  - ▶ subject to  $x_i + x_j \geq 1$  for every edge  $(i, j) \in E$
  - ▶  $x_j \in \{0, 1\}$  for all  $j$ .
- The intended meaning is that  $x_j = 1$  iff vertex  $j$  is in the chosen cover. The constraint forces every edge to be covered by at least one vertex.
- Note that we could have equivalently said that the  $x_j$  just have to be non negative integers since it is clear that any optimal solution would not set any variable to have a value greater than 1.

## LP rounding for the natural vertex cover IP

- The “natural LP relaxation” then is to replace  $x_j \in \{0, 1\}$  by  $x_j \in [0, 1]$  or more simply  $x_j \geq 0$  for all  $j$ .
- It is clear that by allowing the variables to be arbitrary reals in  $[0, 1]$ , we are admitting more solutions than an IP optimal with variables in  $\{0, 1\}$ . Hence the LP optimal has to be at least as good as any IP solution and usually it is better.
- The goal then is to convert an optimal LP solution into an IP solution in such a way that the IP solution is not much worse than the LP optimal (and hence not much worse than an IP optimum)
- Consider an LP optimum  $\mathbf{x}^*$  and create an integral solution  $\bar{\mathbf{x}}$  as follows:  $\bar{x}_j = 1$  iff  $x_j^* \geq 1/2$  and 0 otherwise. We need to show two things:
  - 1  $\bar{\mathbf{x}}$  is a valid solution to the IP (i.e. a valid vertex cover). **Why?**
  - 2  $\sum_j w_j \bar{x}_j \leq 2 \cdot \sum_j w_j x_j^* \leq 2 \cdot IP - OPT$ ; that is, the LP relaxation results in a 2-approximation.

## The integrality gap

- Analogous to the locality gap (that we encountered in local search), for LP relaxations of an IP we can define the integrality gap (for a minimization problem) as  $\max_{\mathcal{I}} \frac{IP-OPT}{LP-OPT}$ ; that is, we take the worst case ratio over all input instances  $\mathcal{I}$  of the IP optimum to the LP optimum. (For maximization problems we take the inverse ratio.)
- Note that the integrality gap refers to a particular IP/LP relaxation of the problem just as the locality gap refers to a particular neighbourhood.
- The same concept of the integrality gap can be applied to other relaxations such as in semi definite programming (SDP).
- It should be clear that the simple IP/LP rounding we just used for the vertex cover problem shows that the integrality gap for the previously given IP/LP formulation is at most 2.
- By considering the complete graph  $K_n$  on  $n$  nodes, it is also easy to see that this integrality gap is at least  $\frac{n-1}{n/2} = 2 - \frac{1}{n}$ .



## Integrality gaps and approximation ratios

- When one proves a positive (i.e. upper) bound (say  $c$ ) on the integrality gap for a particular IP/LP then usually this is a constructive result in that some proposed rounding establishes that the resulting integral solution is within a factor  $c$  of the LP optimum and hence this is a  $c$ -approximation algorithm.
- When one proves a negative (i.e. lower) bound (say  $c'$ ) on the integrality gap then this is only a result about the given IP/LP. **In practice** we tend to see an integrality gap as strong evidence that this particular formulation will not be able to result in a better than  $c'$  approximation. Indeed I know of no natural example where we have a lower bound on an integrality gap and yet nevertheless the IP/LP formulation leads “directly” into a better approximation ratio.
- **In theory** some conditions need to be established to make this into a provable statement. For the VC example, the rounding was independent (for each variable) and “oblivious” (to the input graph). In contrast to the  $K_n$  input, the LP-OPT and IP-OPT coincide for an even length cycle. Hence this integrality gap represents a tight bound on the formulation using a graph oblivious rounding.

# Makespan for the unrelated and restricted machine models: a more sophisticated rounding

In the VC example I use the terms “(input) independent rounding” and “oblivious” rounding.)

- We now return to the makespan problem with respect to the unrelated machines model and the special case of the restricted machine model.
- Recall the unrelated machines model where a job  $j$  is represented by a tuple  $(p_{j,1}, \dots, p_{j,m})$  where  $p_{j,i}$  is the time that job  $j$  uses if scheduled on machine  $i$ .
- An important scheduling result is the Lenstra, Shmoys, Tardos (LST) [1990] IP/LP 2-approximation algorithm for the makespan problem in the unrelated machine model (when  $m$  is part of the input). They also obtain a PTAS for fixed  $m$ .

# The natural IP and the LP relaxation

## The IP/LP for unrelated machines makespan

- Minimize  $T$
- Subject to
  - ①  $\sum_i x_{j,i} = 1$  for every job  $j$     % schedule every job
  - ②  $\sum_j x_{j,i} p_{j,i} \leq T$  for every machine  $i$     % do not exceed makespan
  - ③  $x_{j,i} \in \{0, 1\}$     %  $x_{j,i} = 1$  iff job  $j$  scheduled on machine  $i$
- The immediate LP relaxation is to just have  $x_{j,i} \geq 0$
- Even for identical machines (where  $p_{j,i} = p_j$  for all  $i$ ), the integrality gap IG is unbounded since the input could be just one large job with say size  $T$  leading to an LP-OPT of  $T/m$  and IP-OPT = OPT =  $T$  so that the IG =  $m$ .

## Adapting the natural IP

- As in the PTAS for the identical machine makespan PTAS, we use binary search to find an appropriate approximation  $T$  for the optimal makespan.
- Given a candidate  $T$ , we remove all  $x_{ji}$  such that  $p_{j,i} > T$  and obtain a “search problem” (i.e. constant or no objective function) for finding  $x_{j,i}$  satisfying the IP constraints.
- Once we have found the optimal  $T$  for the search problem, the LST algorithm then shows how to use a non-independent rounding to obtain an integral solution yielding a 2-approximation.
- Note: We use the term “rounding” in a very general sense to mean any efficient way to convert the LP solution into an integral solution.

## Sketch of LST rounding for makespan problem

- Using slack form, LP theory can be used to show that if  $\mathcal{L}$  is a feasible LP with  $m + n$  constraints (not counting the non-negativity constraints for the variables) then  $\mathcal{L}$  has an optimal **basic solution** such that at most  $n + m$  of the variables are non-zero.
- It follows that there are at most  $m$  of the  $n$  jobs that have fractional solutions (i.e. are not assigned to a single machine).
- Jobs assigned to a single machine do not need to be rounded; i.e. if  $x_{j,i} = 1$  then schedule job  $j$  on machine  $i$ .
- Construct a bipartite graph between the  $y \leq m$  fractionally assigned jobs and the  $m$  machines.

## The rounding continued

- The goal is then to construct a matching of size  $y$ ; that, is, the matching dictates how to schedule these fractionally assigned jobs. So it “only” remains to show that this bipartite graph has a matching of size  $y$ . Note, of course, this is what makes the “rounding” non-independent .
- The existence of this matching requires more LP whereby it can be shown (LST credit Dantzig [1963]) that the connected components of the bipartite graph are either trees or trees with one added edge (and therefore causing a unique cycle).
- The resulting schedule then has makespan at most  $2T$  since each fractional job has  $p_{j,i} \leq T$  and the LP has guaranteed a makespan at most  $T$  before assigning the fractional jobs.

## The restricted machine makespan problem

- The restricted machines model is a special case of the unrelated machines problem where for every job  $j$ ,  $p_{j,i} \in \{p_j, \infty\}$ . Hence the LST 2-approximation applies.
- LST show that it is NP hard to do better than a 1.5 approximation for the restricted machines (and hence unrelated machines) problem.
- Shmoys shows that for the special case that  $p_j \in \{1, 2\}$  that the problem can be solved in polynomial time.
- There is a relatively new (somewhat strange) result due to Svensson [2011]. He shows how to approximate the value of the optimum makespan to within a factor of  $33/17 \approx 1.9413 < 2$ . This is proven constructively by a local search algorithm satisfying the approximation. However, the local search is not shown to terminate in polynomial time.
- Note that if we could determine the optimal makespan value in polynomial time, then we can also find an optimal solution in polynomial time. **However, the same cannot be said when we are only approximating the makespan value.**

# The special case of graph orientation

- Consider the special case when there are (at most) two allowable machines for each job. This is called the **graph orientation** problem.
- It turns out easier to reason about the LP rounding applied to the graph orientation problem for the given IP/LP but still the integrality gap is 2.
- A more refined IP/LP by Eveblendr, Krcal and Sgall [2008] achieves a 1.75 approximation for the graph orientation problem.
- **Even for the case when each job can only be scheduled on at most 3 machines, beating the 2-approximation remains an open problem.**



## Some concluding remarks (for now) about LP rounding

- We will hopefully return to LP/LP rounding later. There are some nice notes by Allan Jepson providing some of the geometric concepts underlying LP solutions. (Note: these slides are password protected but I will provide password in class.)
- There can be, of course, many different IP/LP formulations for a given problem. In particular, one often adds additional constraints so that the polytope of the LP solutions is smaller.
- For example, one could simply add constraints  $x_i + x_j + x_k \geq 2$  for every triangle in the graph and more generally, constraints for every odd length cycle. (These inequalities do not essentially change the integrality gap.)
- Adding such constraints corresponds to one round of what is called the LS lift and project method.
- There are a number of lift and project methods. If you are interested, then consult our local expert Toni Pitassi.

## Duality: See Vazirani and Shmoys/Williamson texts, and Williamson article

- For a **primal** maximization (resp. minimization) LP in standard form, the **dual LP** is a minimization (resp. maximization) LP in standard form.
- Specifically, if the primal  $\mathcal{P}$  is:
  - ▶ Minimize  $\mathbf{c} \cdot \mathbf{x}$
  - ▶ subject to  $A_{m \times n} \cdot \mathbf{x} \geq \mathbf{b}$
  - ▶  $\mathbf{x} \geq 0$
- then the dual LP  $\mathcal{D}$  with **dual variables**  $\mathbf{y}$  is:
  - ▶ Maximize  $\mathbf{b} \cdot \mathbf{y}$
  - ▶ subject to  $A_{n \times m}^{tr} \cdot \mathbf{y} \leq \mathbf{c}$
  - ▶  $\mathbf{y} \geq 0$
- Note that the dual (resp. primal) variables are in correspondence to primal (resp. dual) constraints.
- If we consider the dual  $\mathcal{D}$  as the primal then its dual is the original primal  $\mathcal{P}$ . That is, the dual of the dual is the primal.

## An example: set cover

As already noted, the vertex cover problem is a special case of the set cover problem in which the elements are the edges and the vertices are the sets, each set (ie vertex  $v$ ) consisting of the edges adjacent to  $v$ .

### The set cover problem as an IP/LP

$$\begin{aligned} & \text{minimize } \sum_j w_j x_j \\ & \text{subject to } \sum_{j:e_i \in S_j} x_j \geq 1 \quad \text{for all } i; \text{ that is, } e_i \in U \\ & \quad x_j \in \{0, 1\} \text{ (resp. } x_j \geq 0) \end{aligned}$$

### The dual LP

$$\begin{aligned} & \text{maximize } \sum_i y_i \\ & \text{subject to } \sum_{i:e_i \in S_j} y_i \leq w_j \quad \text{for all } j \\ & \quad y_i \geq 0 \end{aligned}$$

If all the parameters in a standard form minimization (resp. maximization) problem are non negative, then the problem is called a **covering** (resp. **packing**) problem. Note that the set cover problem is a covering problem and its dual is a packing problem.

# Duality Theory Overview

- An essential aspect of duality is that a finite optimal value to either the primal or the dual determines an optimal value to both.
- The relation between these two can sometimes be easy to interpret. However, the interpretation of the dual may not always be intuitively meaningful.
- Still, duality is very useful because the duality principle states that optimization problems may be viewed from either of two perspectives and this might be useful as the solution of the dual might be much easier to calculate than the solution of the primal.
- In some cases, the dual might provide additional insight as to how to round the LP solution to an integral solution.
- Moreover, the relation between the primal  $\mathcal{P}$  and the dual  $\mathcal{D}$  will lead to [primal-Dual algorithms](#) and to the so-called [dual fitting](#) analysis.
- In what follows we will assume the primal is a minimization problem to simplify the exposition.

# Strong and Weak Duality

## Strong Duality

If  $x^*$  and  $y^*$  are (finite) optimal primal and resp. dual solutions, then  $\mathcal{D}(y^*) = \mathcal{P}(x^*)$ .

Note: Before it was known that solving LPs was in polynomial time, it was observed that strong duality proves that LP (as a decision problem) is in  $\mathbf{NP} \cap \mathbf{co-NP}$  which strongly suggested that LP was not NP-complete.

## Weak Duality for a Minimization Problem

If  $x$  and  $y$  are primal and resp. dual solutions, then  $\mathcal{D}(y) \leq \mathcal{P}(x)$ .

- Duality can be motivated by asking how one can verify that the minimum in the primal is at least some value  $z$ . To get witnesses, one can explore non-negative scaling factors (i.e. the dual variables) that can be used as multipliers in the constraints. The multipliers, however, must not violate the objective (i.e. cause any multiplies of a primal variable to exceed the coefficient in the objective) we are trying to bound.

## Motivating duality

Consider the motivating example in V. Vazirani's text:

Primal

$$\text{minimize } 7x_1 + x_2 + 5x_3$$

subject to

- (1)  $x_1 - x_2 + 3x_3 \geq 10$
- (2)  $5x_1 + 2x_2 - x_3 \geq 6$
- $x_1, x_2, x_3 \geq 0$

Dual

$$\text{maximize } 10y_1 + 6y_2$$

subject to

- $y_1 + 5y_2 \leq 7$
- $-2y_1 + 2y_2 \leq 1$
- $3y_1 - y_2 \leq 5$
- $y_1, y_2 \geq 0$

Adding (1) and (2) and comparing the coefficient for each  $x_i$ , we have:

$$7x_1 + x_2 + 5x_3 \geq (x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3) \geq 10 + 6 = 16$$

Better yet,

$$7x_1 + x_2 + 5x_3 \geq 2(x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3) \geq 26$$

For an upper bound, setting  $(x_1, x_2, x_3) = (7/4, 0, 11/4)$

$$7x_1 + x_2 + 5x_3 = 7 \cdot (7/4) + 1 \cdot 0 + 5 \cdot (11/4) = 26$$

This proves that the optimal value for the primal and dual (with solution  $(y_1, y_2) = (2, 1)$ ) must be 26.

# Easy to prove weak duality

## The proof for weak duality

$$\begin{aligned}\mathbf{b} \cdot \mathbf{y} &= \sum_{j=1}^m b_j y_j \\ &\leq \sum_{j=1}^m \left( \sum_{i=1}^n A_{ji} x_i \right) y_j \\ &\leq \sum_{i=1}^n \sum_{j=1}^m (A_{ji} y_j) x_i \\ &\leq \sum_{i=1}^n c_i x_i = \mathbf{c} \cdot \mathbf{x}\end{aligned}$$

## Max flow-min Cut in terms of duality

- While the max flow problem can be naturally formulated as a LP, the natural formulation for min cut is as an IP. However, for this IP, it can be shown that the *extreme point solutions* (i.e. the vertices of the polyhedron defined by the constraints) are all integral  $\{0,1\}$  in each coordinate. Moreover, there is a precise sense in which max flow and min cut can be viewed as dual problems. This is described nicely in Vazarani (section 12.2).
- In order to formulate max flow in standard LP form we reformulate the problem so that all flows (i.e. the LP variables) are non-negative. And to state the objective as a simple linear function (of the flows) we add an edge of infinite capacity from the terminal  $t$  to the source  $s$  and hence define a circulation problem.

### The max flow LP

maximize  $f_{t,s}$

subject to  $f_{i,j} \leq c_{i,j}$  for all  $(i,j) \in E$

$$\sum_{j:(j,i) \in E} f_{j,i} - \sum_{j:(i,j) \in E} f_{i,j} \leq 0 \quad \text{for all } i \in V$$

$$f_{i,j} \geq 0 \quad \text{for all } (i,j) \in E$$



## Max flow-min cut duality continued

For the primal edge capacity constraints, introduce dual (“distance”) variables  $d_{i,j}$  and for the vertex flow conservation constraints, introduce dual (“potential”) variables  $p_i$ .

### The fractional min cut dual

$$\begin{aligned} & \text{minimize } \sum_{(i,j) \in E} c_{i,j} d_{i,j} \\ & \text{subject to } \quad d_{i,j} - p_i + p_j \geq 0 \\ & \quad \quad \quad p_s - p_t \geq 1 \\ & \quad \quad \quad d_{i,j} \geq 0; p_i \geq 0 \end{aligned}$$

- Now consider the IP restriction :  $d_{i,j}, p_i \in \{0, 1\}$  and let  $\{(d_{i,j}^*, p_i^*)\}$  be an integral optimum.
- The  $\{0, 1\}$  restriction and second constraint forces  $p_s^* = 1; p_t^* = 0$ .
- The IP optimum then defines a cut  $(S, T)$  with  $S = \{i | p_i^* = 1\}$  and  $T = \{i | p_i^* = 0\}$ .
- Suppose  $(i, j)$  is in the cut, then  $p_i^* = 1, p_j^* = 0$  which by the first constraint forces  $d_{i,j} = 1$ .
- The optimal  $\{0, 1\}$  IP solution (of the dual) defines a min cut.

## Solving the $f$ -frequency set cover by a primal dual algorithm

- In the  $f$ -frequency set cover problem, each element is contained in at most  $f$  sets.
- Clearly, the vertex cover problem is an instance of the 2-frequency set cover.
- As in the vertex cover LP rounding, we can similarly solve the  $f$ -frequency cover problem by obtaining an optimal solution  $\{x_j^*\}$  to the (primal) LP and then rounding to obtain  $\bar{x}_j = 1$  iff  $x_j^* \geq \frac{1}{f}$ . This is, as noted before, a conceptually simple method but requires solving the LP.
- We know that for a minimization problem, any dual solution is a lower bound on any primal solution. One possible goal in a primal dual method for a minimization problem will be to maintain a fractional feasible dual solution and continue to try improve the dual solution. As dual constraints become tight we then set the corresponding primal variables.

# Primal dual for $f$ -frequency set cover continued

## Suggestive lemma

Claim: Let  $\{y_i^*\}$  be an optimal solution to the dual LP and let  $\mathcal{C}' = \{S_j \mid \sum_{e_i \in S_j} y_i^* = w_j\}$ . Then  $\mathcal{C}'$  is a cover.

This suggests the following algorithm:

## Primal dual algorithm for set cover

Set  $y_i = 0$  for all  $i$

$\mathcal{C}' := \emptyset$

**While** there exists an  $e_i$  not covered by  $\mathcal{C}'$

    Increase the dual variables  $y_i$  until there is some  $j$  :

$e_i \in S_j$  and  $\sum_{\{k: e_i \in S_j\}} y_k = w_j$

$\mathcal{C}' := \mathcal{C}' \cup \{S_j\}$

**End While**

## Theorem: Approximation bound for primal dual algorithm

The cover formed by tight constraints in the dual solution provides an  $f$  approximation for the  $f$ -frequency set cover problem.

## Comments on the primal dual algorithm

- What is being shown is that the integral primal solution is within a factor of  $f$  of the dual solution which implies that the primal dual algorithm is an  $f$ -approximation algorithm for the  $f$ -frequency set cover problem.
- In fact, what is being shown is that the integrality gap of this IP/LP formulation for  $f$ -frequency set cover problem is at most  $f$ .
- In terms of implementation we would calculate the minimum  $\epsilon$  needed to make some constraint tight so as to choose which primal variable to set. This  $\epsilon$  could be 0 if a previous iteration had more than one constraint that becomes tight simultaneously. This  $\epsilon$  would then be subtracted from  $w_j$  for  $j$  such that  $e_j \in S_j$ .

## More comments on primal dual algorithms

- We have just seen an example of a basic form of the primal dual method for a minimization problem. Namely, we start with an infeasible integral primal solution and feasible (fractional) dual. (For a covering primal problem and dual packing problem, the initial dual solution can be the all zero solution.) Unsatisfied primal constraints suggest which dual constraints might be tightened and when one or more dual constraints become tight this determines which primal variable(s) to set.
- Some primal dual algorithms extend this basic form by using a second (reverse delete) stage to achieve minimality.
- **NOTE** In the primal dual method we are not solving any LPs. Primal dual algorithms are viewed as “combinatorial algorithms” and in some cases they might even suggest an explicit greedy algorithm.

# A primal dual algorithm with reverse delete : the weighted vertex feedback problem

## The vertex feedback problem

Given a graph  $G = (V, E)$ , a feedback vertex set (FVS)  $F$  is a subset of vertices whose removal will make the resulting graph acyclic. That is, if  $S = V - F$ , then  $G[S] = (S, E[S])$  is acyclic where  $G[S]$  is the graph induced by  $S$ .

- The (weighted) feedback vertex set problem is to compute a minimum size (weight) feedback vertex set.
- The problem (i.e. in its decision version) was one of Karp's original NP complete problems. It has application to circuit design and constraint satisfaction problems. It is as hard as vertex cover.
- An obvious IP for this problem would have the constraints  $\sum_{v \in C} x_v \geq 1$  for every cycle  $C$  in the graph. Not only is this possibly an exponential size IP (which might not be a problem), it is known that the integrality gap is  $\Theta(\log |V|)$ .

## An alternative IP/LP for the FVS problem

- Chudak et al [1998] provide primal dual interpretations for the 2-approximation algorithms due to Becker and Geiger [1994] and Bafna, Berman, Fujito [1995]. In the primal dual interpretations, both algorithms use almost the same IP representation and method for raising dual variables.
- The basic fact underlying the IP representations is the following:

### Fact

Let  $d(v)$  be the degree of  $v$ ,  $b(S) = |E[S]| - |S| + 1$  and  $\tau(S)$  = the size of a minimal feedback set for  $G[S]$ . Then if  $F$  is any FVS, and  $E[S] \neq \emptyset$  then

- 1  $\sum_{v \in F} [d_S(v) - 1] \geq b(S)$  for all  $S \subseteq V$  and hence
- 2  $\sum_{v \in F} d_S(v) \geq b(S) + \tau(S)$

## Primal dual for FVS continued

The IP/LP and the resulting primal dual algorithm is a little easier to state for the Berger and Geiger algorithm but the analysis is perhaps a little simpler for the Bafna et al. algorithm. Here is the formulation for the Berger and Geiger algorithm:

### Primal for Berger and Geiger algorithm

$\mathcal{P}$ : minimize  $\sum_{v \in V} w_v x_v$   
subject to  $\sum_{v \in S} d_S(v) x_v \geq b(S) + \tau(S)$  for all  $S \subseteq V$  with  $E[S] \neq \emptyset$   
IP:  $x_v \in \{0, 1\}$  LP:  $x_v \geq 0$

### The dual

$\mathcal{D}$ : maximize  $\sum_S (b(S) + \tau(S)) y_S$   
subject to  $\sum_{S: v \in S} d_S(v) y_S \leq w_v$  for all  $v \in V$   
 $y_S \geq 0$  for all  $S \subseteq V$  with  $E[S] \neq \emptyset$

**Note:** These are exponential size LPs but that will not be a problem.



## Primal dual for Berger and Geiger

$y_v = 0$  for all  $v$ ;  $\ell := 0$ ;  $F := \emptyset$

$V' := V$ ;  $E' := E$

**While**  $F$  is not a FVS for  $(V', E')$

$\ell := \ell + 1$

recursively remove all isolated vertices and degree 1 vertices and incident edges from  $(V', E')$

$S := V'$  In the Bafna et al algorithm  $S$  is not always set to  $V'$

Increase  $y_S$  until  $\exists v_\ell \in S: \sum_{T: v_\ell \in T} d_T(S) v_T = w_{v_\ell}$

$F := F \cup \{v_\ell\}$

Remove  $v_\ell$  from  $V'$  and all incident edges from  $E'$

**End While**

**For**  $j = \ell..1$  % This is the reverse delete phase

**If**  $F - \{v_j\}$  is an FVS then  $F := F - \{v_j\}$

**End If**

**End For**

# Comments on the primal dual for Berger and Geiger algorithm

- The algorithm as originally stated shows how to efficiently find a  $v_\ell$  so as to make the the dual constraint tight; namely let  $v_\ell = \operatorname{argmin}_{v \in S} w_v / d_S(v_\ell)$  and let  $\epsilon = w_{v_\ell} / d_S(v_\ell)$ . Then  $\epsilon d_S(u)$  is subtracted from  $w_u$  for all  $u \in S$ .
- It is easy to verify that any FVS is a solution to the primal and conversely any IP solution is an FVS.
- It is immediate that the  $F$  computed is an (integral) FVS since the **While** condition forces this.
- The analysis shows that for the dual LP constructs a feasible fractional  $\{y_S\}$  solution satisfying:  
$$\sum_{v \in F} w_v \leq 2 \sum_S (b(S) + \tau(S)) - 2 \sum_S y_S \leq 2 \sum_S (b(S) + \tau(S))$$
- Therefore, the primal dual algorithm is a 2-approximation algorithm.
- The integrality gap is then at most 2 and this is known to be tight. It is also interesting to note that the dual objective function cannot be efficiently evaluated since  $\tau(S)$  is the optimal FVS value for  $G[S]$ .