

Lecture 13

Lecturer: Debmalya Panigrahi

Scribe: Allen Xiao

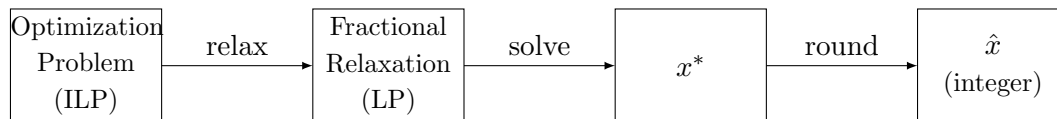
# 1 Overview

In this lecture, we discuss another class of methods for approximating linear programs called *LP rounding*.

# 2 LP Relaxation and Rounding

Recall that many combinatorial problems of interest can be encoded as integer linear programs. Solving integer linear programs is in general NP-hard, so we nearly always *relax* the integrality requirement into a linear constraint like nonnegativity during our analysis.

Our previous algorithms for solving these problems never solved the relaxed program explicitly (e.g. using simplex). In LP rounding, we will directly round the fractional LP solution to generate an integral combinatorial solution. In most cases, this rounding will incur some loss on the solution value, so the results from LP rounding are often approximate.



The approximation analysis works in a similar way to dual fitting. This time, we control the gap between the fractional optimal solution and our rounded solution to bound the gap between the rounded solution and the integer optimal.

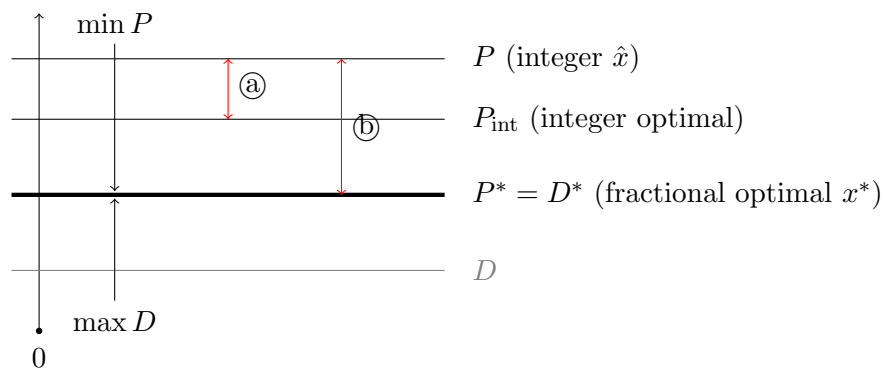


Figure 1: LP rounding solves for  $x^*$ , the fractional optimal solution, and rounds it to an integer feasible solution  $\hat{x}$ . The approximation ratio is the ratio between  $\hat{x}$  and the integer optimal solution (a). We bound this using the ratio between  $\hat{x}$  and  $x^*$  (b).

### 3 Threshold Rounding: Vertex Cover

Recall the linear program for vertex cover on  $G = (V, E)$ .

$$\begin{aligned} \min \quad & \sum_{v \in V} x_v \\ \text{s.t.} \quad & x_v + x_w \geq 1 \quad \forall (v, w) \in E \\ & x_v \geq 0 \quad \forall v \in V \end{aligned}$$

Here, we relaxed the integrality requirement on  $x_v$ .

$$x_v \in \{0, 1\} \longrightarrow x_v \geq 0$$

Let  $x^*$  be a fractional solution to the LP relaxation of vertex cover. We will construct an integer solution by selecting variables of which have a value in  $x^*$  above a certain threshold. Set  $\hat{x}$  to be:

$$\hat{x}_v = \begin{cases} 1 & x_v^* \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Obviously,  $\hat{x}$  is integer. We will show that  $\hat{x}$  is feasible and that the approximation ratio is at most 2.

1. Every constraint is on some edge  $(v, w)$ . In  $x^*$ , the sum of  $x_v^*$  and  $x_w^*$  is at least 1, therefore at least one of the endpoints is rounded to 1. Thus, in  $\hat{x}$ , at least one endpoint has value 1, and the constraint is satisfied.
2. Per vertex, we rounded to  $\hat{x}_v = 1$  only if  $x_v^*$  was at least 1/2 (and kept it 0 otherwise), therefore  $\hat{x}_v \leq 2x_v^*$  for all vertices  $v$ . Summing over the vertices:

$$\begin{aligned} \text{ALGO} &= \sum_{v \in V} \hat{x}_v \\ &\leq 2 \sum_{v \in V} x_v^* = 2\text{OPT} \end{aligned}$$

### 4 Randomized Rounding: Set Cover

Recall the linear program for set cover, with universe  $X$  and sets  $S$  ( $|X| = n$  and  $|S| = m$ ):

$$\begin{aligned} \min \quad & \sum_{s \in S} c_s x_s \\ \text{s.t.} \quad & \sum_{e \in s} x_s \geq 1 \quad \forall e \in X \\ & x_s \geq 0 \quad \forall s \in S \end{aligned}$$

Suppose a linear programming solver gives us a fractional optimal solution  $x^*$ . We round to construct  $\hat{x}$ :

$$\hat{x}_s = \begin{cases} 1 & \text{w.p. } x_s^* \\ 0 & \text{w.p. } (1 - x_s^*) \end{cases}$$

The  $\hat{x}_s$  are Bernoulli random variables. On expectation, the rounded solution has cost equal to the fractional solution's.

$$\mathbb{E} \left[ \sum_{s \in S} c_s \hat{x}_s \right] = \sum_{s \in S} c_s \mathbb{E}[\hat{x}_s] = \sum_{s \in S} c_s x_s^*$$

Whenever we round randomly, we run the risk of the rounded solution being infeasible. Indeed, it is possible for this to occur for  $\hat{x}$ . Consider the case for  $e \in X$  where there are  $m$  singleton sets containing  $e$  ( $s_i = \{e\}$ ). A fractional feasible solution may assign  $1/m$  weight to each of these  $s_i$ , making the individual probability of any  $s_i$  being selected polynomially low. It is very likely for  $e$  to be uncovered in  $\hat{x}$ . In a sense, this is the “worst case” for any element constraint; other distributions will have failure probability  $\leq (1 - 1/m)$ .

In general, what is the probability of failing feasibility?  $\hat{x}$  is infeasible if at least one element  $e \in X$  is uncovered by the sets chosen with  $\hat{x}_s = 1$ .  $e$  may be in as many as  $m$  sets, so:

$$\Pr(\hat{x} \text{ selects no sets covering } e) \leq \left(1 - \frac{1}{m}\right)^m = \frac{1}{e}$$

So the solution is infeasible on  $e$  with constant probability. This not good enough, since we would like to do a union bound over all  $n$  elements. Instead, we need a bound polynomially small. As we have done before with randomized algorithms, we can repeat the algorithm to *boost* the probability of success. Formally, we perform  $r$  rounding trials as  $\hat{x}^{(i)}$ ,

$$\hat{x}_s^{(i)} = \begin{cases} 1 & \text{w.p. } x_s^* \\ 0 & \text{w.p. } (1 - x_s^*) \end{cases}$$

then construct a rounded solution  $\hat{x}$  as follows:

$$\hat{x}_s = \begin{cases} 1 & \text{if } \hat{x}_s^{(i)} = 1 \text{ for any trial } i \\ 0 & \text{otherwise} \end{cases}$$

What is the probability that the solution is infeasible on the constraints for  $e$  now?

$$\Pr(\hat{x} \text{ selects no sets covering } e) \leq \left[ \left(1 - \frac{1}{m}\right)^m \right]^r = e^{-r}$$

For  $r = c \ln n$ , this gives us a failure probability of  $1/n^c$ . If we take the union bound afterwards:

$$\Pr(\hat{x} \text{ is infeasible}) \leq \sum_{e \in X} \Pr(\hat{x} \text{ selects no sets covering } e) = \frac{1}{n^{c-1}}$$

However, the expected cost of  $\hat{x}$  suffers from this boosting.

$$\mathbb{E} \left[ \sum_{s \in S} c_s \hat{x}_s \right] = \sum_{s \in S} c_s \mathbb{E}[\hat{x}_s] = (c \ln n) \sum_{s \in S} c_s x_s^*$$

So the expected approximation ratio is  $O(\log n)$ .

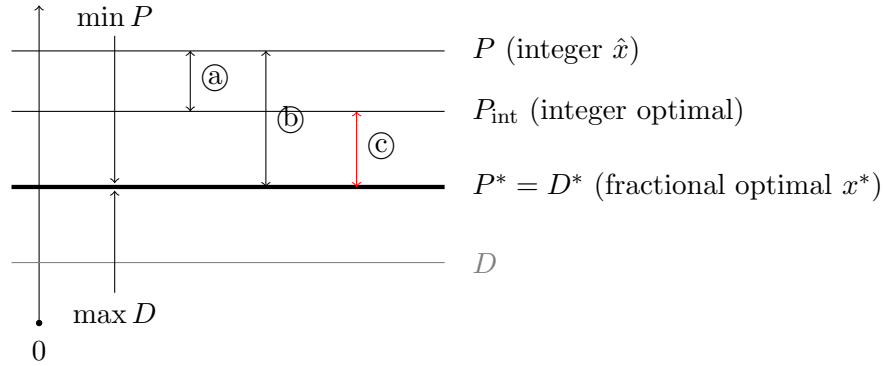


Figure 2: The same figure as before. We bound the approximation ratio of an LP rounding scheme (a) by (b). This analysis will always suffer the integrality gap of the relaxation (c), even if the (a) is small.

## 5 Integrality Gap

You may have noticed that the past few approximation algorithms for set cover were all  $(\log n)$ -approximate. This analysis is tight in the sense that the *integrality gap* for the set cover linear program we described is indeed  $(\log n)$  (i.e. there are examples we can construct where the optimal integer solution is  $(\log n)$  times more expensive than the fractional optimal).

**Definition 1.** For an integer minimization problem, let  $\text{OPT}_{\text{int}}$  be its integer optimal solution and  $\text{OPT}^*$  be the optimal solution to a fractional relaxation. Let the possible instances to the problem be the set of  $I$ . The **integrality gap** of this relaxation of the problem is:

$$\max_I \frac{\text{OPT}_{\text{int}}(I)}{\text{OPT}^*(I)}$$

A similar form exists for maximization problems.

In other words, any integer approximation which relies on a bound against the fractional optimal of this program will incur this penalty in the approximation ratio. There are some problem/relaxation pairs for which there is no gap, however, and the LP optimal admits integer solutions; a few examples we have already seen include maximum flow and maximum bipartite matching. In these cases, we say that the linear program is *exact*.

Finally, integrality gaps are often *unconditional* – they do not rely on assumptions such as  $P = NP$ , unlike many other approximation lower bounds (for example, lower bounds derived from PCP theorem). However, they only affect the specific relaxation, and may not apply to other approximation schemes for the optimization problem.

## 6 Iterative Rounding: Maximum Bipartite Matching

For maximum bipartite matching, we will round the solution  $x^*$  variable by variable (iteratively) until the number of fractional variables is 0. Each iterative step, we use local information to choose

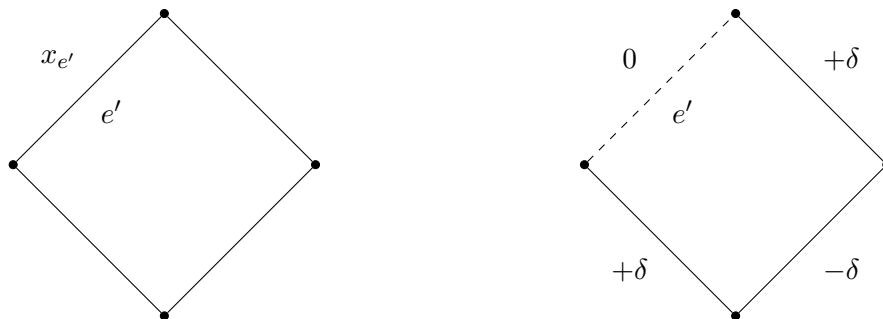


Figure 3: (Phase 1) We break cycles in the subgraph with  $x_e > 0$  (while keeping weight on the cycle constant) by shifting  $\delta = x_{e'}$  around the cycle, where  $e'$  has the minimum weight in the cycle.

the variables and the way to round them. Recall the linear program for bipartite matching:

$$\begin{aligned} \max \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{v \in V} x_{v,w} \geq 1 \quad \forall w \in V \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

Let  $x^*$  be the fractional optimal solution. We construct the rounded solution  $\hat{x}$  iteratively, in two phases. Initialize  $\hat{x} = x^*$ .

1. In the first phase, we make the subgraph where  $x_e > 0$  acyclic using *cycle breaking*.

Suppose  $C$  is a cycle where  $x_e > 0$ . To break  $C$ , we find the minimum weight edge  $e' \in C$ , and force  $x_{e'} = 0$  by subtracting  $\delta = x_{e'}$  from the variable. To preserve the cost on  $C$  and maintain feasibility on all  $v \in C$ , we subtract  $\delta$  from every other edge of  $C$  (starting at  $e'$ ) and add it to the others. Since the graph is bipartite,  $C$  must be even length.

In terms of network flows, one can think of this as pushing flow  $\delta = x_{e'}$  along the residual cycle  $C$  in order to saturate  $e'$ ; preservation of flow balance from augmenting a cycle guarantees that the matching vertex constraints remain feasible.

This process does not create new cycles, since we never add weight to edges which have  $x_e = 0$ . Repeating a sufficient number of times, the subgraph of  $x_e > 0$  edges becomes acyclic.

2. When the subgraph with  $x_e > 0$  is tree-like (i.e. acyclic), we iteratively round from each leaf (degree 1 vertex) to create edges with  $x_e = 1$  and reduce the number of fractional edges.

Suppose we have a leaf edge  $(v, w)$ , where  $v$  is the leaf vertex. The only neighbor of  $v$  is  $w$ . From feasibility, the constraint on  $v$  tells us that:

$$\sum_{v' \in V} x_{v',w} = x_{v,w} + \sum_{v' \neq v} x_{v',w} \leq 1$$

We will round by setting:

$$\begin{aligned} x_{v,w} &= 1 \\ x_{v',w} &= 0 \quad (v' \neq v) \end{aligned}$$



Figure 4: (Phase 2) For leaf vertex  $v$  in the (now acyclic) subgraph of  $0 < x_e < 1$ , we move all the weight from edges around its parent onto the leaf edge. The constraint for  $v$  remains feasible, since in the beginning the sum of weight on the edges adjoining  $v$  did not exceed 1.

The constraints for  $v$  and  $w$  are now tight.

$$\begin{aligned}
 \sum_{v' \in V} x_{v',w} &= x_{v,w} + \sum_{v' \neq v} x_{v',w} \\
 &= 1 + 0 = 1 \\
 \sum_{v' \in V} x_{v',v} &= x_{w,v} \\
 &= 1
 \end{aligned}$$

Since we only removed weight from the  $(v', w)$  edges, constraints on any  $v'$  cannot become infeasible from this rounding. After we remove all edges with  $0 < x_e < 1$ , the only edges with  $x_e > 0$  are the edges with  $x_e = 1$  (matching edges), and we have an integer solution.

How did the objective change? In the first phase, the cost over the cycle was preserved, so the objective value did not change after the first phase. In the second phase, we replaced a set of  $x_e > 0$  which summed to at most 1 with a set which summed exactly to 1, so the second phase does not decrease the objective value. Of course, optimality of  $x^*$  means that  $\hat{x}$  cannot have a greater objective value, therefore it must be that  $\hat{x}$  has the same objective value as  $x^*$ . This rounding procedure is exact.