Lecture 3: Linear Programs, Rounding, and Duality

Lecturer: *Sahil Singla*

Last updated: January 18, 2022

One of the running themes in this course is the notion of *approximate solutions.* Of course, this notion is tossed around a lot in applied work: whenever the exact solution seems hard to achieve, you do your best and call the resulting solution an approximation. In theoretical work, approximation has a more precise meaning whereby you *prove* that the computed solution is close to the exact or optimum solution in some precise metric. We saw some earlier examples of approximation in sampling-based algorithms; for instance our hashing-based estimator for set size. It produces an answer that is w.h.p. within $(1 + \epsilon)$ of the true answer. Today we will see many other examples of approximation that rely upon linear programming (LP).

1 Quick Refresher on Linear Programming

A linear program has a set of variables (in the example below, $x_1, \ldots, x_n$), a linear objective (in the example below, $\vec{c} \cdot \vec{x}$), and a system of linear constraints (in the example below, $A_{ji} \cdot \vec{x} \leq b_j$, for all j, and $x_i \geq 0$ for all i). A linear program in "standard form" therefore takes the following form:

$$\begin{aligned} \max \quad & \sum_i c_i x_i \\ \text{s.t.} \quad & \sum_i A_{ji} x_i \leq b_j, \qquad \forall j \\ & x_i \geq 0, \qquad \forall i. \end{aligned}$$

Recall that it is OK to have variables which aren't constrained to be non-negative, equalities instead of inequalities, min instead of max, etc. (and all such linear programs are equivalent to one written in standard form — if you're unfamiliar with LPs, you may want to prove this as an exercise). Linear programs can be solved in *weakly polynomial time* via the Ellipsoid algorithm (which we'll see later in class) which means the following:

- You are given as input an n-dimensional vector $\vec{c}$, and an $m \times n$ matrix A. Each entry in $\vec{c}$ and A will be a rational number which can be written as the ratio of two b-bit integers.
- Therefore, the input is of size $\text{poly}(n, m, b)$. A weakly polynomial time algorithm is just an algorithm which terminates in time $\text{poly}(n, m, b)$ (and the Ellipsoid algorithm is one such algorithm).
- A stronger stance might be to say that the input is really of size $\text{poly}(n, m)$, but you acknowledge that of course doing numerical operations on b-bit integers will take

time poly(b). A strongly polynomial-time algorithm would be one which performs poly(n, m) numerical operations (and then the algorithm will also terminate in time poly(n, m, b), because each operation terminates in time poly(b). A major (major, major) open problem is whether a strongly poly-time algorithm exists for solving linear programs. Note that the Ellipsoid algorithm does *more numerical operations* if the input numbers have more bits, it's not just that each operation takes longer.

1.1 Basic Solution

For any given a linear program $\max c^T x$ such that $Ax \leq b$ and $0 \leq x_i$, there could be infinitely many optimal solutions. Intuitively, a basic optimal solution is an optimal solution that is at the "vertex" of the feasible region. Such a solution has the advantage that we know that several inequalities will be tight.

Formally, consider a linear program in n fractional variables $x_1, \ldots, x_n$. If the given LP is feasible and finite, a basic optimal solution is guaranteed to have at least n *tight* constraints, i.e., at least n equalities. This has several applications, e.g., we will see one in PSet 1 where we will use the fact that besides the box constraints ($-1 \leq x_i \leq 1$) there are only a small number k of other constraints; hence a basic solution x^* will have at least $n-k$ equalities of the form $x_i^* = -1$ or $x_i^* = 1$, which means there are at least $n-k$ integral variables in x^*.

2 Integer Programs, Bipartite Matching, & Approximation

In discrete optimization problems, we are usually interested in finding 0/1 solutions. Using LP one can find *fractional* solutions, where the relevant variables are constrained to take real values in $[0, 1]$. Sometimes, we can get lucky: you write an LP relaxation for a problem, and the LP happens to produce a 0/1 solution. Now, you know that this 0/1 solution is clearly optimal: not only is it the best 0/1 solution, it's even the best $[0, 1]$ solution. An example of this phenomenon is when we use a linear program to find the minimum cut in a graph (a problem for which we also have non-LP algorithms).

Another important polynomial-time problem that admits a linear program which exactly solves the integral problem is *max-weight bipartite matching*. Given a bipartite graph $G = ((A, B), E)$ with edge weights $w : E \to \mathbb{R}_{\geq 0}$ (i.e., the vertices in G can be partitioned into sets A and B and each edge in E is of the form (a, b) for some vertex $a \in A$ and $b \in B$), the max-weight bipartite matching problem is to find a subset of edges $M \subseteq E$ that do not share a vertex while maximizing $\sum_{e \in M} w(e)$. We won't prove it in class but the optimal value of the following linear program returns the max-weight matching:

$$\begin{aligned}
\max \sum_{(a,b) \in E} & w((a, b)) \cdot x_{(a,b)} & \\
& 0 \leq x_{(a,b)} \leq 1 & \forall (a, b) \in E \\
& \textstyle\sum_{b:(a,b) \in E} x_{(a,b)} \leq 1 & \forall a \in A \\
& \textstyle\sum_{a:(a,b) \in E} x_{(a,b)} \leq 1 & \forall b \in B.
\end{aligned}$$

Needless to say, we don't expect this magic to repeat for NP-hard problems. So the LP relaxation yields a fractional solution in general and the objective provides an upper/lower bound on the optimum for a maximization/minimization problem. Then we give a way to *round* the fractional solutions to 0/1 solutions. This is accompanied by a mathematical proof that the new integral solution is approximates the objective, say to within a multiplicative factor of α. Hence we get an α-approximation algorithm for the Integral problem.

Next we discuss an LP rounding scheme to design approximation algorithms.

3 Deterministic Rounding (Vertex Cover)

First we give an example of the most trivial rounding of fractional solutions to 0/1 solutions: round variables $< 1/2$ to 0 and $\geq 1/2$ to 1. Surprisingly, this is good enough in some settings.

In the *weighted vertex cover* problem, which is NP-hard, we are given a graph $G = (V, E)$ and a weight for each node; the nonnegative weight of node i is w_i. The goal is to find a *vertex cover*, which is a subset S of vertices such that every edge contains at least one vertex of S. Furthermore, we wish to find such a subset of minimum total weight. Let $VC_{\min}$ be this minimum weight. The following is the LP relaxation:

$$\begin{array}{lll} \min & \sum_i w_i x_i & \\ & 0 \leq x_i \leq 1 & \forall i \\ & x_i + x_j \geq 1 & \forall \{i, j\} \in E. \end{array}$$

Let OPT_f be the optimum value of this LP. It is no more than $VC_{\min}$ since every 0/1 solution (including in particular the 0/1 solution of minimum cost) is also an acceptable fractional solution.

Applying *deterministic* rounding, we can produce a new set S: every node i with $x_i \geq 1/2$ is placed in S and every other i is left out of S.

Claim 1: S is a vertex cover.
Reason: For every edge $\{i, j\}$ we know $x_i + x_j \geq 1$, and thus at least one of the x_i's is at least 1/2. Hence at least one of i, j must be in S.

Claim 2: The weight of S is at most $2OPT_f$.
Reason: $\mathrm{OPT}_f = \sum_i w_i x_i$, and we are only picking those i's for which $x_i \geq 1/2$. $\square$.

Thus we have constructed a vertex cover whose cost is within a factor 2 of the optimum cost *even though we don't know the optimum cost per se.*

Remark: This 2-approximation was discovered a long time ago, and despite myriad attempts we still don't know if it can be improved. Using the so-called PCP Theorems, Dinur and Safra [1] showed (improving a long line of work) that 1.36-approximation is NP-hard. Khot and Regev [2] showed that computing a $(2 - \epsilon)$-approximation is UG-hard, which is a new form of hardness popularized in recent years.

4 Randomized Rounding (Set Cover)

In this section we will use LPs to design an approximation algorithm for the NP-hard Set-Cover problem.

In the *min-cost Set Cover problem* we are given a universe $U := \{1, 2, \ldots, n\}$ of n elements and a collection of m subsets $\mathcal{S} := \{S_1, \ldots, S_m\}$ where $\bigcup_k S_k = U$. We are also given a cost function $c : \mathcal{S} \to \mathbb{R}_{\geq 0}$ that assigns a cost to every subset in $\mathcal{S}$. The goal is to find a min-cost sub-collection of $\mathcal{S}$ such that its union equals U. This problem generalizes Vertex Cover when the edges of the graph form the universe U and the subsets correspond to the subset of edges incident on each vertex.

Algorithm. We start by solving the following natural LP relaxation for set cover:

$$\min \sum_{k \in \{1,\ldots,m\}} c(S_k) \cdot x_k$$
$$\sum_{k : S_k \ni e} x_k \geq 1 \qquad \forall e \in \{1, \ldots, n\}$$
$$x_k \geq 0 \qquad \forall k \in \{1, \ldots, m\}$$

Let x^* be an optimal fractional solution to this LP. Our randomized rounding algorithm selects each set S_k independently with probability $\min\{1\ ,\ x_k^* \cdot c \log n\}$, where $c \geq 1$ is an appropriately chosen constant. The following is the main theorem.

Theorem 1. *The above randomized rounding algorithm returns with $\Omega(1)$ probability an $O(\log n)$ approximation feasible solution to the min-cost Set Cover problem.*

Proof. Let OPT denote the cost of the optimal integral solution. Since by setting $x_k = 1$ for sets S_k in the optimal solution, and otherwise setting $x_k = 0$, we get a feasible solution to the LP of cost OPT, we know that the optimal fractional solution x^* to the LP satisfies $\sum_k c(S_k) x_k^* \leq OPT$.

Since we independently select each set independently w.p. $\min\{1\ ,\ x_k^* \cdot c \log n\}$, we get a solution of expected total cost at most $c \log n \cdot \sum_k c(S_k) x_k^* \leq c \log n \cdot \text{OPT}$. Using Markov's Inequality, the probability that this random solution has cost more than $4c \log n \text{OPT}$ is at most $1/4$. Next, we we argue that with probability at least $3/4$ the returned solution is also feasible for set cover. Hence, with probability at least $3/4 - 1/4 = 1/2$ the returned solution is both feasible and has cost at most $4 \log n$.

To prove feasibility, we first calculate the probability of any single element e being covered in the random solution. By LP feasibility, $\sum_{k : S_k \ni e} x_k^* \geq 1$. So, if there is a set $S_k \ni e$ with $x_k^* \geq 1/(c \log n)$ then element e will be covered with probability 1 since $\min\{1\ ,\ x_k^* \cdot c \log n\} = 1$. On the other hand, if each $S_k \ni e$ has $x_k^* < 1/(c \log n)$ then we get $\sum_{k : S_k \ni e} \min\{1\ ,\ x_k^* \cdot c \log n\} > c \log n$. So, by Chernoff bounds and by choosing the constant c sufficiently large, the probability that e is covered is at least $1 - 1/(4n^2)$. Taking a union bound over all the n elements, we get that w.p. at least $1 - 1/(4n) > 3/4$ all the elements are covered and we get a feasible solution, which completes the proof of the theorem. $\square$

Interestingly, Theorem 1 is tight, i.e., there exist set cover problem instances where no polytime algorithm can get $o(\log n)$ approximation, assuming $P \neq NP$ [3]. On the other hand, one could hope to do better for some special family of set cover instances. For instance, in Section 3 we already saw how we can get 2-approximation for the special case of vertex cover. More generally, the approach in Section 3 can be extended to get an f-approximation for min-cost Set Cover, where f is the maximum number of sets in which an element of U appears (in vertex cover, each edge is incident to at most 2 vertices).

5 Weak LP Duality

LP Duality is an extremely useful tool for analyzing structural properties of linear programs. While there are indeed applications of LP duality to directly design algorithms, it is often more useful to gain structural insight (such as approximation guarantees, etc.).

Consider a linear program of the form:

$$\max \sum_i c_i x_i$$
$$\sum_i A_{ji} x_i \leq b_j, \quad \forall j$$
$$x_i \geq 0, \quad \forall i.$$

We'll call this the *primal* LP. $\vec{x}$ is called a *primal* solution, and our goal is to find a primal solution that maximizes our objective, subject to the feasibility constraints. On the other hand, instead of thinking about directly searching for good primal solutions, we could alternatively think about searching for good upper bounds on how good a primal can possibly be. This is called the *dual* problem. How can we derive an upper bound on how good a primal can possibly be?

Consider the following: if we have weights $w_j \geq 0$ for each inequality j, and take a linear combination of the feasibility constraints, we may directly conclude that any feasible $\vec{x}$ must satisfy:

$$\sum_i \Big(\sum_j w_j \cdot A_{ji} \Big) x_i \;\leq\; \sum_j w_j \cdot b_j.$$

Okay, so we can upper bound some linear function of any feasible $\vec{x}$, so what? Well, if we happen to have chosen our w_js so that $\sum_j w_j A_{ji} = c_i$ for all i, now we're in business! We'll have directly shown that $\sum_i c_i x_i \leq \sum_j w_j \cdot b_j$. In fact, because $x_i \geq 0$, even if we only have $\sum_j w_j A_{ji} \geq c_i$ we're in business, as we'd have:

$$\sum_i c_i x_i \;\leq\; \sum_i \Big(\sum_j w_j A_{ji} \Big) \cdot x_i \;\leq\; \sum_j w_j \cdot b_j.$$

Note that the first inequality is only true because $x_i \geq 0$. So now we can think of the following "dual" approach: search over all weights w_j to find the ones that induce the best upper bound. Note that our search is constrained to find weights such that $c_i \leq \sum_j w_j A_{ji}$, so this itself is a linear program:

$$\min \sum_j w_j \cdot b_j$$
$$\sum_j w_j \cdot A_{ji} \geq c_i, \quad \forall i$$
$$w_j \geq 0, \quad \forall j.$$

This is called the dual LP. As an exercise, verify that the dual of the dual LP is itself the primal. Note that we have already proved that *every* feasible solution of the dual provides an upper bound on how good any primal solution can possibly be. Therefore, we have established what is called weak LP duality:

Theorem 2 (Weak LP Duality)**.** *Let LP1 be any maximization LP and LP2 be its dual (a minimization LP). Then if:*

- *The optimum of LP1 is unbounded ($+\infty$), then the feasible region of LP2 is empty.*
- *The optimum of LP1 finite, it is less than or equal to the optimum of LP2, or the feasible region of LP2 is empty.*

Proof. We have already proven the second bullet. To see the first bullet, observe that if the feasible region of LP2 is non-empty, then we have directly found a finite upper bound on LP1. So if LP1 is unbounded, LP2 must be empty. □

In fact, we will see a stronger claim later. Weak Duality is easy to prove, and it's good to remember this intuition. Strong Duality (later) is good to know, but the intuition is largely captured by the proof of Weak Duality.

5.1 Complementary Slackness

We'll also want to discuss properties of optimal primal/dual pairs. One useful property is called *complementary slackness*. A $\vec{x}$ and $\vec{w}$ are said to satisfy complementary slackness if they satisfy condition 1) in the theorem statement below.

Theorem 3. *Consider a primal LP, LP1 and its dual LP, LP2, and feasible (not necessarily optimal) solutions $\vec{x}$ for the primal and $\vec{w}$ for the dual. Then the following are equivalent:*

1. *($w_j = 0$ OR $\sum_i A_{ji} x_i = b_j$ for all j) AND $\left(x_i = 0 \text{ OR } \sum_j A_{ji} w_j = c_i \text{ for all } i\right)$.*
2. *$\sum_i c_i x_i = \sum_j w_j b_j$ (and therefore both $\vec{x}$ is an optimal primal and $\vec{w}$ is an optimal dual).*

Proof. Note that we can write:

$$\sum_i c_i \cdot x_i - \sum_j w_j b_j \leq \sum_i (\sum_j A_{ij} w_j) \cdot x_i - \sum_j w_j b_j = \sum_j w_j \cdot \left(\sum_i A_{ij} x_i - b_j\right).$$

The inequality is because $\vec{w}$ is a feasible solution to LP2. The equality is just rearranging the order of sums. Let's now analyze the RHS. Observe that $\sum_i A_{ij}x_i - b_j \le 0$ for all j as $\vec{x}$ is feasible for LP1. Observe also that $w_j \ge 0$ for all j, as $\vec{w}$ is feasible for LP2. So every term in the summand multiplies a non-negative number by a non-positive number and is therefore non-positive. **This means that the RHS is zero if and only if for all j, $w_j = 0$ or $\sum_i A_{ij}x_i - b_j = 0$.**

Now we turn our attention to the inequality. Note that because $c_i \le \sum_j A_{ij}w_j$ for all i, the inequality is strict if and only if there exists an i for which $x_i > 0$ and $c_i < \sum_j A_{ij}w_j$. **So the LHS is equal to the middle term if and only if for all i, $x_i = 0$ or $c_i = \sum_j A_{ij}w_j$.**

Taking the two bold-font claims together, this means that the LHS is equal to zero if and only if 1) holds. If 1) does not hold, then either the RHS is < 0, or the LHS is less than the middle term (which is ≤ 0). Finally, observe that 2) holds if and only if the LHS above is equal to zero. $\square$

6 Dual Fitting (Set Cover)

In this section we will see another $O(\log n)$ approximation algorithm for the min-cost Set Cover problem from Section 4 that use weak LP duality. Interestingly, our algorithm will never solve an LP, it will use an LP and its dual to only analyze the algorithm.

Recall, in *min-cost Set Cover* we are given a universe $U := \{1, 2, \ldots, n\}$ of n elements and a collection of m subsets $\mathcal{S} := \{S_1, \ldots, S_m\}$ where $\bigcup_k S_k = U$. We are also given a cost function $c : \mathcal{S} \to \mathbb{R}_{\ge 0}$ that assigns a cost to every subset in $\mathcal{S}$. The goal is to find a min-cost subcollection of $\mathcal{S}$ such that its union equals U.

Let OPT denote the cost of the optimal solution. We will analyze approximation guarantees of the Greedy Algorithm in Figure 1. The Greedy Algorithm always returns a valid solution since we assumed $\bigcup_k S_k = U$. We will prove the following result.

Let A represent uncovered elements and let $\mathcal{C}$ represent selected sets.
Initialization: $A = U$ and $\mathcal{C} = \emptyset$.
While $A \neq \emptyset$:

1. Find a set $S_k \in \mathcal{S}$ that maximizes $\alpha = \dfrac{1}{c(S_k)} \cdot \left(\left|S_k \cap A\right|\right)$.
2. For each newly covered element $e \in S_k \cap A$, set $\mathit{price}(e) = 1/\alpha$.
3. Update $A \leftarrow A \setminus S_k$ and $\mathcal{C} \leftarrow \mathcal{C} \cup k$.

Note that the second step doesn't affect the algorithm. It will be only used for analysis.

Figure 1: The Greedy Algorithm

Theorem 4. *The Greedy Algorithm gives an H_n approximation to the min-cost Set Cover problem, where $H_n := 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} = \Theta(\log n)$.*

There are several proofs known for this theorem. (You might want to try a direct combinatorial proof.) We will use the LP from Section 4 to prove the result. Recall that the optimal solution to that LP satisfies $\sum_k c(S_k)x_k^* \leq \text{OPT}$.

We will actually never compute x^* and only use the fact that the LP value gives a lower bound on OPT. Instead, we compare the Greedy Algorithm to the following dual LP (since the primal LP was a minimization problem, the dual LP is a maximization problem):

$$\begin{aligned} \max \sum_{e\in\{1,\ldots,n\}} & y_e \\ \sum_{e\in S_k} y_e \leq c(S_k) & \qquad \forall k \in \{1,\ldots,m\} \\ y_e \geq 0 & \qquad \forall e \in \{1,\ldots,n\} \end{aligned}$$

By weak-duality, we know that any feasible solution to the dual LP gives a lower bound on $\sum_k c(S_k)x_k^*$. Thus, our plan to upper bound the cost of the greedy algorithm is to show a feasible dual solution $\left(z_e\right)_{e\in E}$ which satisfies that the total cost of the Greedy Algorithm

$$\sum_{k\in\mathcal{C}} c(S_k) \leq H_n \cdot \sum_e z_e. \tag{1}$$

This will prove Theorem 4 since by weak-duality $\sum_e z_e \leq \sum_k c(S_k)x_k^* \leq \text{OPT}$.

We define $z_e = H_n \cdot \mathit{price}(e)$, where recall from Figure 1 that $\mathit{price}(e)$ intuitively denotes how much element e pays for S_k when it first gets covered. Proving (1) is easy because the Greedy Algorithm sets an element's price exactly once, and at that point the algorithm exactly distributes $c(S_k)$ between the newly covered elements. Thus the algorithm's total cost $\sum_{k\in\mathcal{C}} c(S_k) = \sum_e \mathit{price}(e) = \frac{1}{H_n}\sum_e z_e$.

Next, we prove that $\left(z_e\right)_{e\in E}$ is a feasible solution to the dual LP. By definition, $z_e \geq 0$. Now consider any set $S \in \mathcal{S}$. We want to show $\sum_{e\in S} z_e \leq c(S)$. Start by renaming the elements (for analysis) of S to be $\{e_1, e_2, \ldots, e_{|S|}\}$ in the order they are covered by the Greedy Algorithm (breaking ties arbitrarily). Observe that whenever an element $e_i \in S$ was first covered, the algorithm had the option of selecting S (and may be it even did) at a cost of $c(S)$ and cover $|S| - i + 1$ new elements. Thus, irrespective of which set the Greedy algorithm actually selects, we know that $\mathit{price}(e_i) \leq \frac{c(S)}{|S|-i+1}$. This implies

$$\sum_{e\in S} z_e \quad = \quad \frac{1}{H_n}\sum_{e_i\in S} \mathit{price}(e_i) \quad \leq \quad \frac{1}{H_n}\sum_{e_i\in S}\frac{c(S)}{|S|-i+1} \quad = \quad c(S)\cdot\frac{H_{|S|}}{H_n} \quad \leq \quad c(S).$$

7 Strong LP Duality

In Section 5 we discussed weak duality: using dual solutions as upper bounds on how good a primal solution could be. In fact, something quite strong is true: there is always a dual witnessing that the optimal primal is optimal. We'll give a proof, but note that most of the intuition (aside from geometry/linear algebra) is provided by Weak Duality. We'll just discuss the "classic" case, the "partial" case is similar and omitted. First recall the primal and the dual LPs.

$$\max \sum_i c_i x_i \qquad \text{(LP1)}$$
$$\sum_i A_{ji} x_i \leq b_j, \quad \forall j$$
$$x_i \geq 0, \quad \forall i.$$

$$\min \sum_j w_j \cdot b_j \qquad \text{(LP2)}$$
$$\sum_j w_j \cdot A_{ji} \geq c_i, \quad \forall i$$
$$w_j \geq 0, \quad \forall j.$$

Theorem 5 (Strong LP Duality)**.** *Let LP1 be any maximization LP and LP2 be its dual (a minimization LP). Then:*

- *If the optimum of LP1 is unbounded ($+\infty$), the feasible region of LP2 is empty.*
- *If the feasible region of LP1 is empty, the optimum of LP2 is either unbounded ($-\infty$), or also infeasible.*
- *If optimum of LP1 finite, then the optimum of LP2 is also finite, and they are equal.*

The key ingredient in the proof will be what's called the Separating Hyperplane Theorem.

Theorem 6 (Separating Hyperplane Theorem)**.** *Let P be a closed, convex region in $\mathbb{R}^n$, and $\vec{x}$ be a point not in P. Then there exists a $\vec{w}$ such that $\vec{x} \cdot \vec{w} > \max_{\vec{y} \in P}\{\vec{y} \cdot \vec{w}\}$.*

Proof. Consider the point $\vec{y} \in P$ closest to $\vec{x}$ (that is, minimizing $||\vec{x} - \vec{y}||_2$ over all $\vec{y} \in P$. As distance is a positive continuous function, and P is a closed region, such a $\vec{y}$ exists. Now consider the vector $\vec{w} = \vec{x} - \vec{y}$. We claim that the chosen $\vec{w}$ is the desired witness.

Observe first that $(\vec{x} - \vec{y}) \cdot \vec{w} = ||\vec{w}||_2^2 > 0$, so indeed $\vec{x} \cdot \vec{w} > \vec{y} \cdot \vec{w}$. We just need to confirm that $\vec{y} = \arg\max_{\vec{z} \in P}\{\vec{z} \cdot \vec{w}\}$ and then we're done.

Assume for contradiction that $\vec{z} \cdot \vec{w} > \vec{y} \cdot \vec{w}$ and $\vec{z} \in P$. Then as P is convex, $\vec{z}_\varepsilon = (1-\varepsilon)\vec{y} + \varepsilon\vec{z} \in P$ as well for all $\varepsilon > 0$. Observe that $||\vec{x} - \vec{z}_\varepsilon||_2^2 = ||\vec{x} - \vec{y} + \varepsilon(\vec{y} - \vec{z})||_2^2 = ||\vec{x} - \vec{y}||_2^2 + 2\varepsilon(\vec{x} - \vec{y}) \cdot (\vec{y} - \vec{z}) + \varepsilon^2||\vec{y} - \vec{z}||_2^2 = ||\vec{x} - \vec{y}||_2^2 + 2\varepsilon\vec{w} \cdot (\vec{y} - \vec{z}) + \varepsilon^2||\vec{y} - \vec{z}||_2^2$. By hypothesis, $\vec{w} \cdot (\vec{y} - \vec{z}) < 0$, and $||\vec{y} - \vec{z}||_2^2$ is finite, so for sufficiently small ε, we get $||\vec{x} - \vec{z}_\varepsilon||_2^2 < ||\vec{x} - \vec{y}||_2^2$, a contradiction. □

Now, consider the optimum $\vec{x}$ of LP1. Let S denote the j for which $\sum_i A_{ji} x_i = b_j$, and $\bar{S}$ the constraints for which $\sum_i A_{ji} x_i < b_j$. We claim that $\vec{c}$ can be written as a convex combination of the vectors $\vec{A}_j$, $j \in S$ (up to possible scaling).

Lemma 7. *Let $\vec{x}$ be the optimum of LP1, and let S denote the j for which $\sum_i A_{ji} x_i = b_j$. Then there exist $\{\lambda_j \geq 0\}_{j \in S}$ such that $c_i = \sum_{j \in S} \lambda_j A_{ji}$ for all i.*

Proof. Assume for contradiction that this were not the case. As the space X of all vectors $\vec{y}$ for which there exists $\{\lambda_j \geq 0\}_{j \in S}$ such that $y_i = \sum_{j \in S} \lambda_j A_{ji}$ for all i is clearly closed and convex, we can apply the separating hyperplane theorem. So there would exist some $\vec{\gamma}$ such that $\vec{c} \cdot \vec{\gamma} > \max_{\vec{y} \in X}\{\vec{y} \cdot \vec{\gamma}\}$. Now consider the vector $\vec{x} + \varepsilon\vec{\gamma}$.

We know that for all $j \in S$, $\sum_i A_{ji}\gamma_i \leq 0$. If not, then $\max_{\vec{y} \in X}\{\vec{y} \cdot \vec{\gamma}\} = +\infty$, because we could blow up λ_j. So for all $j \in S$, $\sum_i A_{ji}(x_i + \varepsilon\gamma_i) \leq b_j$. Moreover, for all $i \notin S$, $\sum_i A_{ji}x_i < b_j$, and $\sum_i A_{ji}\gamma_i$ is finite. So there exists a sufficiently small ε so that $\vec{x} + \varepsilon\vec{\gamma}$ is feasible for LP1.

Finally, observe that $\max_{\vec{y} \in X}\{\vec{y} \cdot \vec{\gamma}\} \geq 0$, as $\vec{0} \in X$. So $\vec{c} \cdot \vec{\gamma} > 0$, and we have shown that $\vec{x}$ was not optimal. □

Now with the lemma in hand, we want to show a dual whose value matches $\vec{c} \cdot \vec{x}$. Let $\vec{c} = \sum_{j \in S} \lambda_j \vec{A}_j$ with $\lambda_j \geq 0$ as guaranteed by the lemma. Set $w_j = \lambda_j$ for all $j \in S$, and $w_j = 0$ for all $j \notin S$. First, is it clear that $\vec{w}$ is feasible for LP2, as we have explicitly set w_j so that $c_i = \sum_j w_j A_{ij}$ for all i. Now we just need to evaluate its value:

$$\sum_j b_j w_j = \sum_{j \in S} b_j w_j + \sum_{j \notin S} b_j \cdot 0 = \sum_{j \in S} (\sum_i A_{ji} x_i) w_j = \sum_i \left(\sum_{j \in S} A_{ji} w_j \right) x_i = \sum_i c_i x_i.$$

So its objective value is exactly the same as LP1.

Notes

The lecture is partly based on COS 521 notes from Princeton University. The proof of Theorem 6 is adapted from Anupam Gupta's scribe lecture notes [4].

References

[1] Dinur, I., and Safra, S. (2005). On the hardness of approximating minimum vertex cover. Annals of mathematics, 439-485.

[2] Khot, S., and Regev, O. (2008). Vertex cover might be hard to approximate to within $2 - \epsilon$. Journal of Computer and System Sciences, 74(3), 335-349.

[3] Feige, Uriel. A threshold of ln n for approximating set cover. Journal of the ACM (JACM) 45.4 (1998): 634-652.

[4] https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture05.pdf