COS 330: Great Ideas in Theoretical Computer Science

Fall 2026

Lecture 13

Duality

Module: Optimization

These notes cover the same material as the Lecture 13 slides, written in a long text format instead of slide format.

1 Certificates of Optimality

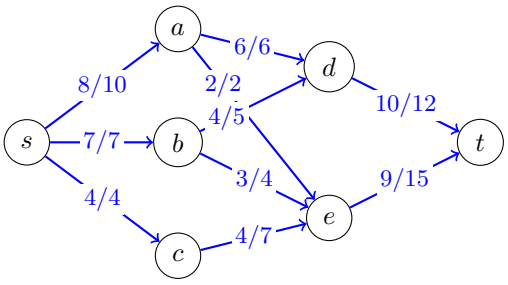
Over the last two lectures we used linear programs as a tool for approximation algorithms. The pattern was always the same: write the problem as an integer program, relax it to an LP, and use the LP optimum $\mathrm{OPT}_{\mathrm{LP}}$ as a *certificate*, a bound on OPT that we can actually compute. For a minimization problem, the recipe was:

1. find a certificate $L(I) \le \mathrm{OPT}(I)$;
2. design an algorithm with $A(I) \le \alpha L(I)$;
3. conclude $A(I) \le \alpha L(I) \le \alpha\,\mathrm{OPT}(I)$.

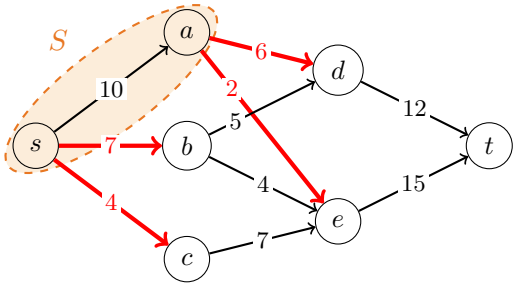
Today is a detour from approximation algorithms. Instead of using LPs to certify something else, we ask how to certify a solution *of the LP itself*. Suppose someone hands us a feasible solution to an LP and claims it is optimal. Checking feasibility is easy: plug the values into each constraint. But how could they convince us that no other feasible solution does better? The answer is a property of every LP called *duality*. It is a very powerful tool, but it's not always intuitive at first glance, so we will build up to it through two examples. You will see applications of duality to approximation and to algorithms in general in the PSet.

1.1 Warm-up: maximum flow

We have already seen a certificate of optimality in this course. Consider the network on the left below, with source s, sink t, and each edge labeled *flow/capacity*. You can check that this flow respects every capacity and conserves flow at each intermediate vertex.



A flow of value 19



A cut of capacity 19

This flow has value $8 + 7 + 4 = 19$, which proves that the maximum flow is at least 19. But maybe there is a better flow we haven't found.

To rule that out, we also exhibit an s-t cut, shown on the right. Take $S = \{s, a\}$ and $T = \{b, c, d, e, t\}$. The edges from S to T are $s \to b$, $s \to c$, $a \to d$, and $a \to e$, so the cut has capacity $7 + 4 + 6 + 2 = 19$. By the weak duality theorem for flows from Lecture 3, the value of any flow is at most the capacity of any cut. So no flow has value more than 19, and no cut has capacity less than 19: the flow and the cut are both optimal.

Notice what just happened. We had a *maximization* problem (max flow), and we certified the optimality of a solution using a related *minimization* problem (min cut). Feasible solutions to one bound the other, and when their values match, both are optimal. This is the essence of duality, and today we will see that it works for *every* linear program.

1.2 Example: smoothies

You have a stock of 3 oranges, 10 bananas, and 6 strawberries, and you can make three types of smoothies:

Smoothie	**Ingredients per liter**	**Profit per liter**
Type 1 (Orange–Banana)	1 orange, 2 bananas	\$5
Type 2 (Strawberry–Banana)	1 banana, 2 strawberries	\$4
Type 3 (Citro–Bananaberry)	1 orange, 1 banana, 1 strawberry	\$6

You can make fractions of liters, but no negative amounts. Letting x_j be the number of liters of type j, the problem of maximizing profit is the LP

$$\begin{aligned} \text{Maximize:} \quad & 5x_1 + 4x_2 + 6x_3 \\ \text{Subject to:} \quad & x_1 + x_3 \le 3 && \text{(oranges)} \\ & 2x_1 + x_2 + x_3 \le 10 && \text{(bananas)} \\ & 2x_2 + x_3 \le 6 && \text{(strawberries)} \\ & x_1, x_2, x_3 \ge 0. \end{aligned}$$

Someone claims that $x_1 = 3$, $x_2 = 3$, $x_3 = 0$ is optimal. It is feasible: it uses $3 \le 3$ oranges, $2(3) + 3 = 9 \le 10$ bananas, and $2(3) = 6 \le 6$ strawberries. Its profit is $5(3) + 4(3) = 27$. How can we prove that no feasible solution does better?

We need to show that $5x_1 + 4x_2 + 6x_3 \le 27$ for *every* feasible solution. The idea is to combine the constraints. Multiplying an inequality by a nonnegative number keeps it valid, and so does adding two valid inequalities. Multiply the orange constraint by 5 and the strawberry constraint by 2:

$$\begin{aligned} 5 \times [x_1 + x_3 \le 3] &\implies 5x_1 + 5x_3 \le 15, \\ 2 \times [2x_2 + x_3 \le 6] &\implies 4x_2 + 2x_3 \le 12. \end{aligned}$$

Adding these up gives $5x_1 + 4x_2 + 7x_3 \le 27$. Since $x_3 \ge 0$ for any feasible solution,

$$5x_1 + 4x_2 + 6x_3 \le 5x_1 + 4x_2 + 7x_3 \le 27.$$

So every feasible solution has profit at most 27, and we have a feasible solution with profit exactly 27: it is optimal. The multipliers $(5, 0, 2)$ for the orange, banana, and strawberry constraints are a *certificate of optimality*.

How did we know to use $5, 0, 2$? Let's look for multipliers systematically. Multiply the three constraints by $y_1, y_2, y_3 \geq 0$ and add them up:

$$(y_1 + 2y_2)\, x_1 + (y_2 + 2y_3)\, x_2 + (y_1 + y_2 + y_3)\, x_3 \leq 3y_1 + 10y_2 + 6y_3.$$

Since $x_1, x_2, x_3 \geq 0$, the left side is at least the profit $5x_1 + 4x_2 + 6x_3$ whenever each coefficient is large enough:

$$y_1 + 2y_2 \geq 5, \qquad y_2 + 2y_3 \geq 4, \qquad y_1 + y_2 + y_3 \geq 6.$$

Any such y gives the upper bound $3y_1 + 10y_2 + 6y_3$ on the profit, and we want the best (smallest) one. Finding it is itself an LP:

$$\begin{aligned} \text{Minimize:} \quad & 3y_1 + 10y_2 + 6y_3 \\ \text{Subject to:} \quad & y_1 + 2y_2 \geq 5, \quad y_2 + 2y_3 \geq 4, \quad y_1 + y_2 + y_3 \geq 6 \\ & y_1, y_2, y_3 \geq 0. \end{aligned}$$

This is called the *dual* LP. Our multipliers $(5, 0, 2)$ are feasible for it, with value $15 + 12 = 27$.

2 Duality

Duality is all about finding *upper bounds* (for maximization problems) or *lower bounds* (for minimization problems) on the optimal value of a linear program. The key idea is that any valid linear combination of the constraints of a linear program gives a valid bound on the objective function.

2.1 The primal LP

So let's say we have a maximization linear program (LP) that looks like the following:

$$\begin{aligned} \text{Maximize:} \quad & \sum_j c_j x_j \\ \text{Subject to:} \quad & \sum_j a_{i,j} x_j \leq b_i \quad \forall i \\ & x_j \geq 0 \quad \forall j \end{aligned}$$

We call this the *primal* LP. Its variables x_j are indexed by j, its constraints are indexed by i, and $a_{i,j}$ is the coefficient of variable j in constraint i. This is the same convention as in Lecture 12. An LP in exactly this shape (maximize, all constraints $\leq$, all variables nonnegative) is said to be in *standard form*.

2.2 Constructing the dual

Our goal is to take linear combinations of the constraints to get an upper bound on the objective function $\sum_j c_j x_j$. Here's the four-step "recipe" for doing this:

Step 1: Create dual variables. For each constraint in the primal LP, create a dual variable y_i. These variables represent the "weights" we'll use to combine the constraints.

Step 2: Ensure validity. When we take a linear combination of the primal constraints using weights y_i, we get:

$$\sum_i y_i \left(\sum_j a_{i,j} x_j \right) \leq \sum_i y_i b_i$$

For this combination to remain valid we need $y_i \geq 0$ (otherwise multiplying by a negative weight would flip the inequality). These conditions on y_i become the *validity constraints* in the dual.

Step 3: Ensure usefulness. We want the best upper bound on the objective possible, so we need the combined constraints to actually bound the objective function:

$$\sum_j c_j x_j \leq \sum_i y_i \left(\sum_j a_{i,j} x_j \right) = \sum_j \left(\sum_i a_{i,j} y_i \right) x_j$$

Above we swapped the order of the sums and rearranged the terms accordingly. By doing so, we can observe that, since every $x_j \geq 0$, we just need $c_j \leq \sum_i a_{i,j} y_i$ (the coefficient of x_j on the right must dominate the left) to satisfy this for all j.

These conditions become the *usefulness constraints* in the dual.

Step 4: Optimize the bound. We now have:

$$\sum_j c_j x_j \leq \sum_j \left(\sum_i a_{i,j} y_i \right) x_j \leq \sum_i y_i b_i$$

To get the tightest possible upper bound, we minimize $\sum_i b_i y_i$ over all valid choices of y_i. This becomes the dual LP! For completeness, here is the final dual LP:

$$\begin{aligned} \text{Minimize:} \quad & \sum_i b_i y_i \\ \text{Subject to:} \quad & \sum_i a_{i,j} y_i \geq c_j \quad \forall j \\ & y_i \geq 0 \quad \forall i \end{aligned}$$

Notice the roles of the indices: j indexes the primal variables and the dual constraints, while i indexes the primal constraints and the dual variables. Taking the dual "transposes" the structure of the LP. You can check that the smoothie dual from Section 1.2 is exactly what this recipe produces.

2.3 Weak duality

The four steps above already contain a proof that every feasible dual solution bounds every feasible primal solution.

Theorem 2.1 (Weak duality)**.** If x is feasible for the primal LP and y is feasible for the dual LP, then

$$\sum_j c_j x_j \leq \sum_i b_i y_i.$$

Proof. Since y is dual feasible, $\sum_i a_{i,j} y_i \geq c_j$ for every j, and since $x_j \geq 0$, multiplying by x_j preserves this inequality. Since x is primal feasible, $\sum_j a_{i,j} x_j \leq b_i$ for every i, and since $y_i \geq 0$, multiplying by y_i preserves this inequality too. Therefore

$$\sum_j c_j x_j \leq \sum_j \Big(\sum_i a_{i,j} y_i \Big) x_j = \sum_i y_i \Big(\sum_j a_{i,j} x_j \Big) \leq \sum_i y_i b_i. \qquad \blacksquare$$

In particular, if we find a primal feasible x and a dual feasible y with equal objective values, then both are optimal: no primal solution can beat $\sum_i b_i y_i$, and x already attains it. This is exactly what happened with the smoothies.

2.4 Other forms of LPs

We did not cover this section in class, and it is not essential to follow the rest of the lecture. Still, it is useful to know that *every* LP has a dual, not just LPs in the form we studied, and one way to see this is to transform any LP into that form.

One important note that we slightly glossed over above is that the steps we performed only work for a primal LP in standard form. For example, if x_j were unconstrained (could be negative), then we can still perform similar steps, but the validity and usefulness constraints would be slightly different.

Fortunately, every LP can be rewritten in standard form with mechanical transformations that preserve its solutions:

- **Minimization objective.** To convert a minimization problem to maximization, we negate the objective. For instance, minimizing $2x_1 + 3x_2$ is the same as maximizing $-2x_1 - 3x_2$: if the optimal value of the new LP is $-v$, then the optimal value of the original LP is v.
- **Greater-than-or-equal constraints.** To convert $\geq$ constraints to $\leq$ constraints, multiply both sides by -1. For instance, $2x_1 + x_2 \geq 4$ becomes $-2x_1 - x_2 \leq -4$.
- **Equality constraints.** An equality constraint $a = b$ is equivalent to both $a \leq b$ and $a \geq b$. So we can replace the equality with two inequalities, and then convert the second one as above. For instance, $x_1 + x_2 = 5$ becomes $x_1 + x_2 \leq 5$ and $-(x_1 + x_2) \leq -5$.
- **Unconstrained variables.** For an unconstrained variable x_2, we can write it as the difference of two non-negative variables: $x_2 = x_2^+ - x_2^-$ where $x_2^+, x_2^- \geq 0$, and substitute this everywhere x_2 appears. When $x_2 > 0$, we can set $x_2^+ = x_2$ and $x_2^- = 0$. When $x_2 < 0$, we can set $x_2^+ = 0$ and $x_2^- = -x_2$. When $x_2 = 0$, we can set both to 0.

So the recipe applies, after these conversions, to every LP.

2.5 The dual of the dual

The dual LP is itself an LP, so it has a dual. It turns out that taking the dual of the dual gives you back the primal. This shows that primal and dual is a *symmetric* relationship: neither is more "fundamental" than the other.

Exercise. Show that the dual of the dual LP is the primal LP.

Hint: It might be tempting to first convert the dual to standard maximization form and then apply the recipe from Section 2.2, but this ends up being more complicated than necessary. Instead, directly apply the four-step framework, but adapt the direction of inequalities and the objective for minimization (it should be very similar to Section 2.2, just with flipped inequalities). Since the dual is a minimization problem, you are now looking for the best *lower* bound on its objective.

2.6 Strong duality

Weak duality says that the primal value is at most the dual value whenever both are feasible. In the smoothie example, the two optima were equal. Is that a coincidence?

Theorem 2.2 (Strong duality)**.** If the primal LP has an optimal solution x^*, then the dual LP also has an optimal solution y^*, and their optimal values are equal:

$$\sum_j c_j x_j^* = \sum_i b_i y_i^*.$$

We won't prove this theorem. The proof is a bit long and uses some theory of LPs; see an optimization course (like ORF 307) for details. Its consequence is what we were looking for at the start of the lecture: every optimal solution of an LP has a certificate of optimality, namely an optimal dual solution.

Remark 2.1. The theorem assumes that the primal LP has an optimal solution, and in particular that it has at least one feasible solution. This is not automatic: some LPs have contradicting constraints, and then no feasible solution exists. For example, no x_1 satisfies both $x_1 \leq 1$ and $x_1 \geq 2$. For such LPs the theorem, in the form stated here, does not apply. So before using strong duality, we have to check that the LP is *feasible*, that is, that it admits at least one feasible solution. (Feasibility alone is not quite enough: an LP can also be unbounded, like maximizing x_1 subject only to $x_1 \geq 0$, in which case it has no optimal solution either. This is why in Section 3.3 we check both.)

3 Application: Matching and Vertex Cover

Let's revisit a problem from Lecture 11 through the lens of duality. Consider an undirected graph $G = (V, E)$. A *matching* is a set of edges with no shared endpoints, and a *vertex cover* is a set of vertices touching all edges. The maximum matching problem seeks the largest matching, while the minimum vertex cover problem seeks the smallest vertex cover.

In Lecture 11 we proved that for every matching M and every vertex cover S, $|M| \leq |S|$: the cover must contain an endpoint of each edge of M, and these endpoints are distinct. In particular, the maximum matching size is at most the minimum vertex cover size. We will now see this inequality again, this time as a consequence of duality.

3.1 The two relaxations

Maximum matching is the integer program

$$\begin{aligned} \text{Maximize:} \quad & \sum_{e \in E} x_e \\ \text{Subject to:} \quad & \sum_{e \text{ incident to } v} x_e \leq 1 \quad \forall v \in V \\ & x_e \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

The x_e represents whether edge e is in the matching, and the constraints ensure that each vertex is matched to at most one edge. Its LP relaxation replaces $x_e \in \{0, 1\}$ with $0 \leq x_e \leq 1$. As usual, the relaxation has a larger feasible set, so for this maximization problem $\text{OPT}_{\text{IP}} \leq \text{OPT}_{\text{LP}}$. The upper bound $x_e \leq 1$ is already implied by the vertex constraints, so we can drop it and write the LP relaxation for maximum matching as:

$$\begin{aligned} \text{Maximize:} \quad & \sum_{e \in E} x_e \\ \text{Subject to:} \quad & \sum_{e \text{ incident to } v} x_e \leq 1 \quad \forall v \in V \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

Similarly, minimum vertex cover is the integer program

$$\begin{aligned} \text{Minimize:} \quad & \sum_{v \in V} y_v \\ \text{Subject to:} \quad & y_u + y_v \geq 1 \quad \forall (u,v) \in E \\ & y_v \in \{0,1\} \quad \forall v \in V \end{aligned}$$

where y_v represents whether v is in the cover. Its LP relaxation replaces $y_v \in \{0,1\}$ with $0 \leq y_v \leq 1$, and for this minimization problem $\text{OPT}_{\text{LP}} \leq \text{OPT}_{\text{IP}}$. This is the same relaxation we rounded in Lecture 12.

3.2 From the Matching LP to the Vertex Cover LP

Note that the Matching LP is in the primal maximization form we discussed earlier, so we can apply the four-step process directly. Let's derive the dual step by step using the key principles:

Step 1: Create dual variables. For each constraint in the primal (one per vertex $v \in V$), we create a dual variable y_v. These will be the variables in our dual LP.

Step 2: Set up validity constraints. We take a linear combination of the primal constraints using weights y_v:

$$\sum_{v \in V} y_v \left(\sum_{e \text{ incident to } v} x_e \right) \leq \sum_{v \in V} y_v \cdot 1$$

We need $y_v \geq 0$ for all vertices v to ensure this combination remains valid. This gives us the *validity constraints*: $y_v \geq 0$ for all $v \in V$.

Step 3: Set up usefulness constraints. We can rewrite the left side by swapping the order of summation:

$$\sum_{v \in V} y_v \left(\sum_{e \text{ incident to } v} x_e \right) = \sum_{e \in E} \left(\sum_{v \text{ incident to } e} y_v \right) x_e$$

It's a bit subtle, but note how "e incident to v" becomes "v incident to e" when we swap the sums, and we go from summing over vertices to summing over edges. For this to be useful as an upper bound on the primal objective $\sum_{e \in E} x_e$, we need:

$$\sum_{e \in E} x_e \leq \sum_{e \in E} \left(\sum_{v \text{ incident to } e} y_v \right) x_e$$

Since each primal variable satisfies $x_e \geq 0$, this holds when:

$$1 \leq \sum_{v \text{ incident to } e} y_v \quad \forall e \in E$$

Each edge has exactly two endpoints, so for an edge $e = (u,v)$ this becomes:

$$y_u + y_v \geq 1 \quad \forall e = (u,v) \in E$$

These are the *usefulness constraints*.

Step 4: Minimize to get the tightest bound. Putting it all together, we have:

$$\sum_{e \in E} x_e \leq \sum_{e \in E} \left(\sum_{v \text{ incident to } e} y_v \right) x_e \leq \sum_{v \in V} y_v$$

To get the tightest upper bound on the primal objective, we minimize the right-hand side over all valid choices of y_v.

Thus, the resulting LP is:

$$\begin{aligned} \text{Minimize:} \quad & \sum_{v \in V} y_v \\ \text{Subject to:} \quad & y_u + y_v \geq 1 \quad \forall e = (u, v) \in E \\ & y_v \geq 0 \quad \forall v \in V \end{aligned}$$

Notice that this is exactly the LP relaxation for the minimum vertex cover problem! In other words, *the dual of the Matching LP is the Vertex Cover LP*. (As with matching, the upper bound $y_v \leq 1$ can be dropped: lowering any $y_v > 1$ to 1 keeps every constraint satisfied and only decreases the objective, so it does not change the optimal value.) The dual variable y_v indicates whether vertex v is in the cover, and the constraints ensure every edge is covered. The matching variables correspond to the vertex cover constraints (one per edge), and the matching constraints correspond to the vertex cover variables (one per vertex).

3.3 Putting it together

By strong duality (Theorem 2.2), the two LPs have the same optimal value. (The Matching LP does have an optimal solution: it is feasible, since $x = 0$ satisfies every constraint, and bounded, since its value is at most $|E|$; an LP that is feasible and bounded has an optimal solution, a fact we take for granted from the theory of LPs.) Combining this with the relationships between each integer program and its relaxation,

$$\begin{aligned} \text{Max matching (IP)} &\leq \text{Matching LP} \\ &= \text{Vertex Cover LP} \qquad \text{(strong duality)} \\ &\leq \text{Min vertex cover (IP)}. \end{aligned}$$

So the maximum matching size is at most the minimum vertex cover size in every graph: the same inequality as Lecture 11, now as a consequence of duality. In fact weak duality (Theorem 2.1) already gives it, since the inequality between the two LP optima is all we need.

The inequalities in this chain can be strict. In a triangle K_3, the maximum matching has one edge and the minimum vertex cover has two vertices. Setting every variable to $1/2$ is feasible for both LPs, with value $3/2$ in each. By weak duality, a primal and a dual solution with equal values are both optimal, so both LP optima equal $3/2$, strictly between 1 and 2.