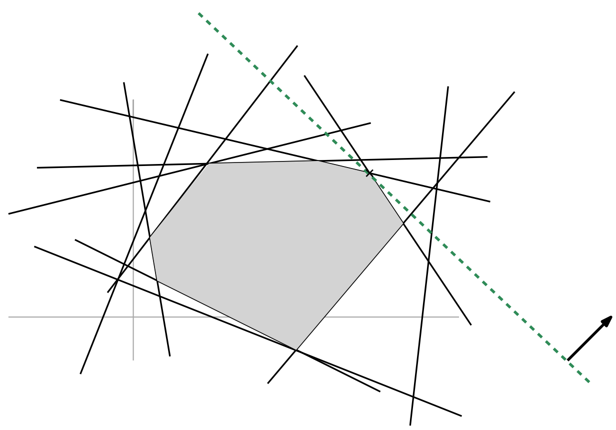
In this lecture we describe a very nice algorithm due to Seidel for Linear Programming in low-dimensional spaces. We then discuss the general notion of Linear Programming Duality, a powerful tool that you should definitely master.

1 Seidel's LP algorithm

We now describe a linear-programming algorithm due to Raimund Seidel that solves the 2-dimensional (i.e., 2-variable) LP problem in $O(m)$ time (recall, m is the number of constraints), and more generally solves the d-dimensional LP problem in time $O(d!m)$.

Setup: We have d variables $x_1, \ldots, x_d$. We are given m linear constraints in these variables $\mathbf{a}_1 \cdot \mathbf{x} \leq b_1, \ldots, \mathbf{a}_m \cdot \mathbf{x} \leq b_m$ along with an objective $\mathbf{c} \cdot \mathbf{x}$ to maximize. (Using boldface to denote vectors.) Our goal is to find a solution $\mathbf{x}$ satisfying the constraints that maximizes the objective. In the example above, the region satisfying all the constraints is given in gray, the arrow indicates the direction in which we want to maximize, and the cross indicates the $\mathbf{x}$ that maximizes the objective.



(You should think of sweeping the green dashed line, to which the vector $\mathbf{c}$ is normal (i.e., perpendicular), in the direction of $\mathbf{c}$, until you reach the last point that satisfies the constraints. This is the point you are seeking.)

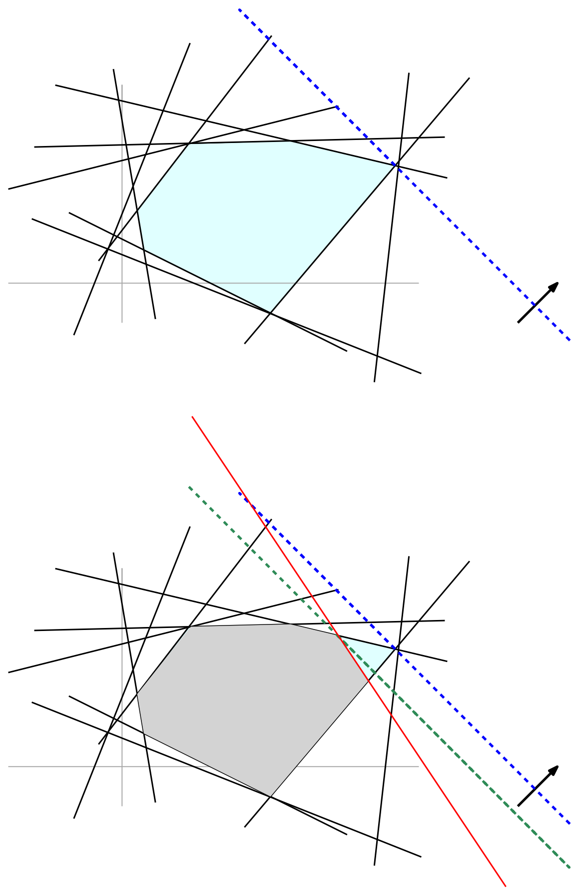
The idea: Here is the idea of Seidel's algorithm. [1] Let's add in the (real) constraints one at a time, and keep track of the optimal solution for the constraints so far. Suppose, for instance, we have found the optimal solution $\mathbf{x}^*$ for the first $m-1$ constraints, and now we add in the mth constraint $\mathbf{a}_m \cdot \mathbf{x} \leq b_m$. There are two cases to consider:

Case 1: If $\mathbf{x}^*$ satisfies the constraint, then $\mathbf{x}^*$ is still optimal. Time to perform this test: $O(d)$.

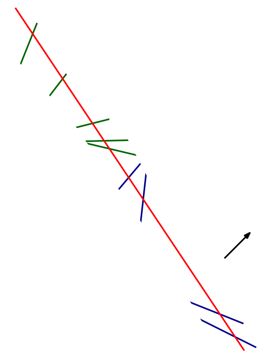
Case 2: If $\mathbf{x}^*$ doesn't satisfy the constraint, then the new optimal point will be on the $(d-1)$-dimensional hyperplane $\mathbf{a}_m \cdot \mathbf{x} = b_m$, or else there is no feasible point.

[1] To keep things simple, let's assume that we have inequalities of the form $-\lambda \leq x_i \leq \lambda$ for all i with sufficiently large λ which are separate from the "real" constraints, so that the starting optimal point is one of the corners of the box $[-\lambda, \lambda]^d$. See Section 1.1 for how to remove this assumption.

As an example, consider the situation below, before and after we add in the linear constraint $\mathbf{a}_m \cdot \mathbf{x} \leq b_m$ colored in red. This causes the feasible region to change from the light blue region to the smaller gray region, and the optimal point to move.



Let's now focus on the case $d = 2$ and consider the time it takes to handle Case 2 above. With $d = 2$, the hyperplane $\mathbf{a}_m \cdot \mathbf{x} = b_m$ is just a line, and let's call one direction "right" and the other "left". We can now scan through all the other constraints, and for each one, compute its intersection point with this line and whether it is "facing" right or left (i.e., which side of that point satisfies the constraint). We find the rightmost intersection point of all the constraints facing to the right and the leftmost intersection point of all that are facing left. If they cross, then there is no solution. Otherwise, the solution is whichever endpoint gives a better value of $\mathbf{c} \cdot \mathbf{x}$ (if they give the same value – i.e., the line $\mathbf{a}_m \cdot \mathbf{x} = b_m$ is perpendicular to $\mathbf{c}$ – then say let's take the rightmost point). In the example above, the 1-dimensional problem is the one in the figure below, with the green constraints "facing" one direction and the blue ones facing the other way. The direction of $\mathbf{c}$ means the optimal point is given by the "lowest" green constraint.



The total time taken here is $O(m)$ since we have $m-1$ constraints to scan through and it takes $O(1)$ time to process each one.

Right now, this looks like an $O(m^2)$-time algorithm for $d = 2$, since we have potentially taken $O(m)$ time to add in a single new constraint if Case 2 occurs. But, suppose we add the constraints in *a random order*? What is the probability that constraint m goes to Case 2?

Notice that the optimal solution to all m constraints (assuming the LP is feasible and bounded) is at a corner of the feasible region, and this corner is defined by two constraints, namely the two sides of the polygon that meet at that point. If both of those two constraints have been seen already, then we are guaranteed to be in Case 1. So, if we are inserting constraints in a random order, the probability we are in Case 2 when we get to constraint m is at most $2/m$. This means that the *expected* cost of inserting the mth constraint is at most:

$$\text{E[cost of inserting } m\text{th constraint]} \leq (1 - 2/m)O(1) + (2/m)O(m) = O(1).$$

This is sometimes called "backwards analysis" since what we are saying is that if we go backwards and pluck out a random constraint from the m we have, the chance it was one of the constraints that mattered was at most $2/m$.

So, Seidel's algorithm is as follows. Place the constraints in a random order and insert them one at a time, keeping track of the best solution so far as above. We just showed that the expected cost of the ith insert is $O(1)$ (or if you prefer, we showed $T(m) = O(1) + T(m-1)$ where $T(i)$ is the expected cost of a problem with i constraints), so the overall expected cost is $O(m)$.

1.1 Handling Special Cases, and Extension to Higher Dimensions*

(We will not be testing you on this part, but you should try to understand it all the same.)

What if the LP is infeasible? There are two ways we can analyze this. One is that if the LP is infeasible, then it turns out this is determined by at most 3 constraints. So we get the same as above with $2/m$ replaced by $3/m$. Another way to analyze this is imagine we have a separate account we can use to pay for the event that we get to Case 2 and find that the LP is infeasible. Since that can only happen once in the entire process (once we determine the LP is infeasible, we stop), this just provides an additive $O(m)$ term. To put it another way, if the system is infeasible, then there will be two cases for the final constraint: (a) it was feasible until then, in which case we pay $O(m)$ out of the extra budget (but the above analysis applies to the the (feasible) first $m-1$ constraints), or (b) it was infeasible already in which case we already halted so we pay 0.

What about unboundedness? We had said for simplicity we could put everything inside a bounding box $-\lambda \leq x_i \leq \lambda$. E.g., if all c_i are positive then the initial $\mathbf{x}^* = (\lambda, \ldots, \lambda)$. However, what value of λ should we choose? We could actually do the calculations viewing λ symbolically as a limiting quantity which is arbitrarily large. For example, in 2-dimensions, if $\mathbf{c} = (0,1)$ and we have a constraint like $2x_1 + x_2 \le 8$, then we would see it is not satisfied by (λ, λ), and hence intersect the contraint with the box and update to $\mathbf{x}^* = (4 - \lambda/2, \lambda)$.

So far we have shown that for $d = 2$, the expected running time of the algorithm is $O(m)$. For general values of d, there are two main changes. First, the probability that constraint m enters Case 2 is now d/m rather than $2/m$. Second, we need to compute the time to perform the update in Case 2. Notice, however, that this is a $(d-1)$-dimensional linear programming problem, and so we can use the same algorithm recursively, after we have spent $O(dm)$ time to project each of the $m-1$ constraints onto the $(d-1)$-dimensional hyperplane $\mathbf{a}_m \cdot \mathbf{x} = b_m$. Putting this together we have a recurrence for expected running time:

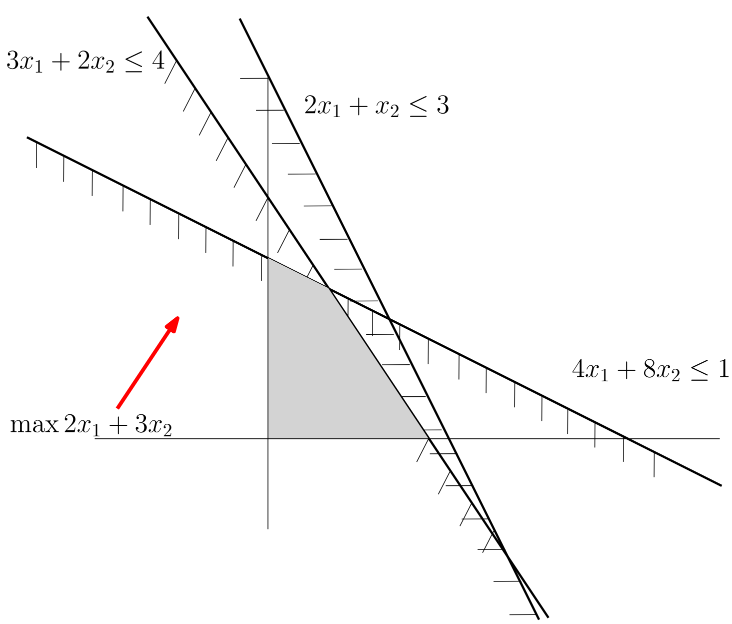
$$T(d,m) \;\le\; T(d,m-1) + O(d) + \frac{d}{m}[O(dm) + T(d-1,m-1)].$$

This then solves to $T(d,m) = O(d!m)$.

2 Linear Programming Duality

Consider the following LP

$$\begin{aligned} P = \max(2x_1 + 3x_2) \\ \text{s.t.} \quad 4x_1 + 8x_2 \le 12 \\ 2x_1 + x_2 \le 3 \\ 3x_1 + 2x_2 \le 4 \\ x_1, x_2 \ge 0 \end{aligned} \tag{1}$$



In an attempt to solve P we can produce upper bounds on its optimal value.

- Since $2x_1 + 3x_2 \le 4x_1 + 8x_2 \le 12$, we know $\text{OPT}(P) \le 12$. (The first inequality uses that $2x_1 \le 4x_1$ because $x_1 \ge 0$, and similarly $3x_2 \le 8x_2$ because $x_2 \ge 0$.)
- Since $2x_1 + 3x_2 \le \frac{1}{2}(4x_1 + 8x_2) \le 6$, we know $\text{OPT}(P) \le 6$.

- Since $2x_1 + 3x_2 \leq \frac{1}{3}((4x_1 + 8x_2) + (2x_1 + x_2)) \leq 5$, we know $\mathrm{OPT}(P) \leq 5$.

In each of these cases we take a positive[2] linear combination of the constraints, looking for better and better bounds on the maximum possible value of $2x_1 + 3x_2$.

How do we find the "best" lower bound that can be achieved as a linear combination of the constraints? This is just another algorithmic problem, and we can systematically solve it, by letting y_1, y_2, y_3 be the (unknown) coefficients of our linear combination. Then we must have

$$\begin{aligned} 4y_1 + 2y_2 + 3y_2 &\geq 2 \\ 8y_1 + y_2 + 2y_3 &\geq 3 \\ y_1, y_2, y_3 &\geq 0 \\ \text{and we seek } \min(12y_1 + 3y_2 + 4y_3) \end{aligned} \tag{2}$$

This too is an LP! We refer to this LP (2) as the "dual" and the original LP 1 as the "primal". We designed the dual to serve as a method of constructing an upper bound on the optimal value of the primal, so if y is a feasible solution for the dual and x is a feasible solution for the primal, then $2x_1 + 3x_2 \leq 12y_1 + 3y_2 + 4y_3$. If we can find two feasible solutions that make these equal, then we know we have found the optimal values of these LP.

In this case the feasible solutions $x_1 = \frac{1}{2}, x_2 = \frac{5}{4}$ and $y_1 = \frac{5}{16}, y_2 = 0, y_3 = \frac{1}{4}$ give the same value 4.75, which therefore must be the optimal value.

Exercise: The dual LP is a *minimization* LP, where the constraints are of the form $lhs_i \geq rhs_i$. You can try to give *lower* bounds on the optimal value of this LP by taking positive linear combinations of these constraints. E.g., argue that

$$12y_1 + 3y_2 + 4y_3 \geq 4y_1 + 2y_2 + 3y_2 \geq 2$$

(since $y_i \geq 0$ for all i) and

$$12y_1 + 3y_2 + 4y_3 \geq 8y_1 + y_2 + 2y_3 \geq 3$$

and

$$12y_1 + 3y_2 + 4y_3 \geq \frac{2}{3}(4y_1 + 2y_2 + 3y_2) + (8y_1 + y_2 + 2y_3) \geq \frac{4}{3} + 3 = 4\frac{1}{3}.$$

Formulate the problem of finding the best lower bound obtained by linear combinations of the given inequalities as an LP. Show that the resulting LP is the same as the primal LP 1.

Exercise: Consider the "primal" LP below on the left:

$$\begin{aligned} P = \max(7x_1 - x_2 + 5x_3) \\ \text{s.t.} \quad x_1 + x_2 + 4x_3 &\leq 8 \\ 3x_1 - x_2 + 2x_3 &\leq 3 \\ 2x_1 + 5x_2 - x_3 &\leq -7 \\ x_1, x_2, x_3 &\geq 0 \end{aligned} \qquad \begin{aligned} D = \min(8y_1 + 3y_2 - 7y_3) \\ \text{s.t.} \quad y_1 + 3y_2 + 2y_3 &\geq 7 \\ y_1 - y_2 + 5y_3 &\geq -1 \\ 4y_1 + 2y_2 - y_3 &\geq 5 \\ y_1, y_2, y_3 &\geq 0 \end{aligned}$$

Show that the problem of finding the best upper bound obtained using linear combinations of the constraints can be written as the LP above on the right (the "dual" LP). Also, now formulate the problem of finding a lower bound for the dual LP. Show this lower-bounding LP is just the primal (P).

Exercise: In the examples above, the maximization LPs had constraints of the form $lhs_i \leq rhs_i$, and the rhs were all scalars, so taking positive linear combinations gave us *blah* $\leq$ *number*, i.e., an *upper* bound as we wanted. However, suppose the primal LP has some "nice" constraints $lhs_i \leq rhs_i$ and others are "not nice" $lhs_i \geq rhs_i$, e.g., like the left one below. Show that the dual has non-positive

[2] Why positive? If you multiply by a negative value, the sign of the inequality changes.

variables for the non-nice constraints. For example,

$$\begin{aligned} P = \max(7x_1 - x_2 + 5x_3) \\ \text{s.t.} \quad x_1 + x_2 + 4x_3 \le 8 \\ 3x_1 - x_2 + 2x_3 \ge 3 \\ x_1, x_2, x_3 \ge 0 \end{aligned} \qquad \begin{aligned} D = \min(8y_1 + 3y_2) \\ \text{s.t.} \quad y_1 + 3y_2 \ge 7 \\ y_1 - y_2 \ge -1 \\ 4y_1 + 2y_2 \ge 5 \\ y_1 \ge 0, y_2 \le 0 \end{aligned}$$

Another way is to replace $lhs_i \ge rhs_i$ in P by the equivalent constraint $(-lhs_i) \le (-rhs_i)$ and get to an LP P' with only nice constraints. Show that the dual D' for P' is equivalent to the dual D for P.

2.1 The Method

Consider the examples/exercises above. In all of them, we started off with a "primal" maximization LP:

$$\begin{aligned} \text{maximize } & \mathbf{c}^T\mathbf{x} \\ \text{subject to } & A\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \ge \mathbf{0}, \end{aligned} \tag{3}$$

The constraint $\mathbf{x} \ge \mathbf{0}$ is just short-hand for saying that the $\mathbf{x}$ variables are constrained to be non-negative.[3] And to get the best lower bound we generated a "dual" minimization LP:

$$\begin{aligned} \text{minimize } & \mathbf{r}^T\mathbf{y} \\ \text{subject to } & P\mathbf{y} \ge \mathbf{q} \\ & \mathbf{y} \ge \mathbf{0}, \end{aligned} \tag{4}$$

The important thing is: this matrix P, and vectors $\mathbf{q}, \mathbf{r}$ are not just any vectors. Look carefully: $P = A^T$. $\mathbf{q} = \mathbf{c}$ and $\mathbf{r} = \mathbf{b}$. The dual is in fact:

$$\begin{aligned} \text{minimize } & \mathbf{y}^T\mathbf{b} \\ \text{subject to } & \mathbf{y}^T A \ge \mathbf{c}^T \\ & \mathbf{y} \ge \mathbf{0}, \end{aligned} \tag{5}$$

And if you take the dual of (5) to try to get the best lower bound on this LP, you'll get (4). *The dual of the dual is the primal.* The dual and the primal are best upper/lower bounds you can obtain as linear combinations of the inputs.

The natural question is: maybe we can obtain better bounds if we combine the inequalities in more complicated ways, not just using linear combinations. Or do we obtain optimal bounds using just linear combinations? In fact, we get optimal bounds using just linear combinations, as the next theorems show.

2.2 The Theorems

It is easy to show that the dual (5) provides an upper bound on the value of the primal (4):

Theorem 1 (Weak Duality) *If $\mathbf{x}$ is a feasible solution to the primal LP (4) and $\mathbf{y}$ is a feasible solution to the dual LP (5) then*

$$\mathbf{c}^T\mathbf{x} \le \mathbf{y}^T\mathbf{b}.$$

[3]We use the convention that vectors like $\mathbf{c}$ and $\mathbf{x}$ are column vectors. So $\mathbf{c}^T$ is a row vector, and thus $\mathbf{c}^T\mathbf{x}$ is the same as the inner product $\mathbf{c} \cdot \mathbf{x} = \sum_i c_i x_i$. We often use $\mathbf{c}^T\mathbf{x}$ and $\mathbf{c} \cdot \mathbf{x}$ interchangeably. Also, $\mathbf{a} \le \mathbf{b}$ means component-wise inequality, i.e., $a_i \le b_i$ for all i.

Proof: This is just a sequence of trivial inequalities that follow from the LPs above:

$$\mathbf{c}^T\mathbf{x} \leq (y^T A)\mathbf{x} = y^T(A\mathbf{x}) \leq y^T b.$$

■

The amazing (and deep) result here is to show that the dual actually gives a perfect upper bound on the primal (assuming some mild conditions).

Theorem 2 (Strong Duality Theorem) *Suppose the primal LP (4) is feasible (i.e., it has at least one solution) and bounded (i.e., the optimal value is not ∞). Then the dual LP (5) is also feasible and bounded. Moreover, if $\mathbf{x}^*$ is the optimal primal solution, and $\mathbf{y}^*$ is the optimal dual solution, then*

$$\mathbf{c}^T\mathbf{x}^* = (\mathbf{y}^*)^T\mathbf{b}.$$

In other words, the maximum of the primal equals the minimum of the dual.

Why is this useful? If I wanted to prove to you that $\mathbf{x}^*$ was an optimal solution to the primal, I could give you the solution $\mathbf{y}^*$, and you could check that $\mathbf{x}^*$ was feasible for the primal, $\mathbf{y}^*$ feasible for the dual, and they have equal objective function values.

This min-max relationship is like in the case of *s*-*t* flows: the maximum of the flow equals the minimum of the cut. Or like in the case of zero-sum games: the payoff for the maxmin-optimum strategy of the row player equals the (negative) of the payoff of the maxmin-optimal strategy of the column player. Indeed, both these things are just special cases of strong duality!

We will not prove Theorem 2 in this course, though the proof is not difficult. But let's give a geometric intuition of why this is true in the next section.

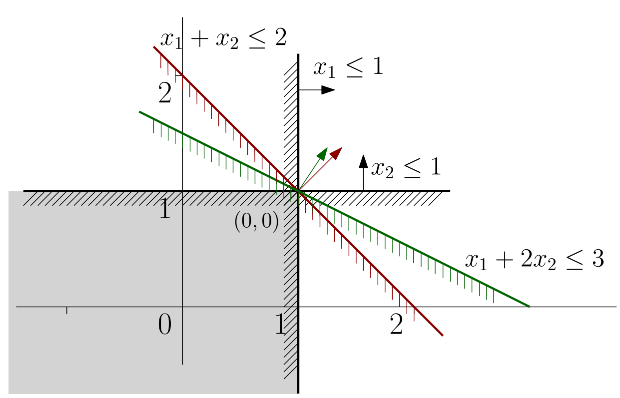
2.3 A Geometric Viewpoint

To give a geometric view of the strong duality theorem, consider an LP of the following form:

$$\begin{aligned} \text{maximize } & \mathbf{c}^T\mathbf{x} \\ \text{subject to } & A\mathbf{x} \leq \mathbf{b} \end{aligned} \tag{6}$$

Given two constraints like $\mathbf{a}_1 \cdot \mathbf{x} \leq b_1$ and $\mathbf{a}_2 \cdot \mathbf{x} \leq b_2$, notice that you can add them to create more constraints that have to hold, like $(\mathbf{a}_1 + \mathbf{a}_2) \cdot \mathbf{x} \leq b_1 + b_2$, or $(0.7\mathbf{a}_1 + 2.9\mathbf{a}_2) \cdot \mathbf{x} \leq (0.7\mathbf{b}_1 + 2.9\mathbf{b}_2)$. In fact, any positive linear combination has to hold.

To get a feel of what this looks like geometrically, say we start with constraints $x_1 \leq 1$ and $x_2 \leq 1$. These imply $x_1 + x_2 \leq 2$ (the red inequality), $x_1 + 2x_2 \leq 3$ (the green one), etc.



In fact, you can create any constraint running through the intersection point $(1, 1)$ that has the entire feasible region on one side by using different positive linear combinations of these inequalities.

Now, suppose you have the LP (6) in n variables with objective $\mathbf{c} \cdot \mathbf{x}$ to maximize. As we mentioned when talking about the simplex algorithm, unless the feasible region is unbounded (and let's assume for this entire discussion that the feasible region is bounded), the optimum point will occur at some vertex $\mathbf{x}^*$ of the feasible region, which is an intersection of n of the constraints, and have some value $v^* = \mathbf{c} \cdot \mathbf{x}^*$.

Consider the n *inequality* constraints that define the vertex $\mathbf{x}^*$, say these are

$$\mathbf{a}_1 \cdot \mathbf{x} \leq b_1, \quad \mathbf{a}_2 \cdot \mathbf{x} \leq b_2, \quad \ldots, \quad \mathbf{a}_n \cdot \mathbf{x} \leq b_n,$$

so that for each $i \in \{1, 2, \ldots, n\}$ the point $\mathbf{x}^*$ satisfies the *equalities*

$$\mathbf{a}_1 \cdot \mathbf{x} = b_1, \quad \mathbf{a}_2 \cdot \mathbf{x} = b_2, \quad \ldots, \quad \mathbf{a}_n \cdot \mathbf{x} = b_n.$$

Just as in the simple example above, if you take these n inequality constraints that define the vertex $\mathbf{x}^*$ and look at all positive linear combinations of these, you can again create any constraint you want going through $\mathbf{x}^*$ that has the entire feasible region on one side. One such constraint is $\mathbf{c} \cdot \mathbf{x} \leq v^*$. It goes through $\mathbf{x}^*$ (since we have $\mathbf{c} \cdot \mathbf{x}^* = v^*$) and every point in the feasible region is contained in it (since no feasible point has value more than v^*). So it is possible to create the constraint $\mathbf{c} \cdot \mathbf{x} \leq v^*$ using some positive linear combination of the $\mathbf{a}_i \cdot \mathbf{x} \leq b_i$ constraints.

Why is this interesting?

We've shown a *short proof* (a "succinct certificate") that $\mathbf{x}^*$ is optimal. Indeed, if I gave you a solution $\mathbf{x}^*$ and claimed it was optimal for the given constraints and the objective function $\mathbf{c} \cdot \mathbf{x}$, it is not clear how I would convince you of $\mathbf{x}$'s optimality. In 2-dimensions I could draw a figure, but in higher dimensions things get more difficult. But we've just shown that I can take a positive linear combination of the given constraints $\mathbf{a}_i \cdot x \leq b_i$ and create the constraint $\mathbf{c} \cdot \mathbf{x} \leq v^* = \mathbf{c} \cdot \mathbf{x}^*$, hence showing we can't do any better.

How do we find this positive linear combination of the constraints? Hey, it's actually just another linear program. Indeed, suppose we want to find the best possible bound $\mathbf{c} \cdot \mathbf{x} \leq v$ for as small a value v as possible. Say the original LP had the m constraints

$$\mathbf{a}_1 \cdot \mathbf{x} \leq b_1, \quad \mathbf{a}_2 \cdot \mathbf{x} \leq b_2, \quad \ldots, \quad \mathbf{a}_m \cdot \mathbf{x} \leq b_m,$$

written compactly as $A\mathbf{x} \leq \mathbf{b}$.

What's our goal? We want to find positive values $y_1, y_2, \ldots, y_m$ such that

$$\sum_i y_i \mathbf{a}_i = \mathbf{c}.$$

From this positive linear combination we can infer the upper bound

$$\mathbf{c} \cdot \mathbf{x} = (\sum_i y_i \mathbf{a}_i) \cdot \mathbf{x} \leq \sum_i y_i b_i.$$

And we want this upper bound to be as "tight" (i.e., small) as possible, so let's solve the LP:

$$\min \sum_i b_i y_i \qquad \text{subject to} \qquad \sum_i y_i \mathbf{a}_i = \mathbf{c}.$$

(In matrix notation, if $\mathbf{y}$ is a $m \times 1$ column vector consisting of the y_i variables, then we want to minimize $\mathbf{y}^T\mathbf{b}$ subject to $\mathbf{y}^T A = \mathbf{c}$.) This is yet again the same process as in the example at the beginning of lecture.

Let us summarize: we started off with the "primal" LP,

$$\begin{aligned} \text{maximize } & \mathbf{c}^T\mathbf{x} \\ \text{subject to } \quad & A\mathbf{x} \leq \mathbf{b} \end{aligned} \tag{7}$$

and were trying to find the best bound on the optimal value of this LP. And to do this, we wrote the "dual" LP:

$$\begin{aligned} \text{minimize } & \mathbf{y}^T\mathbf{b} \\ \text{subject to } \quad & \mathbf{y}^T A = \mathbf{c}^T \\ & \mathbf{y} \geq \mathbf{0}. \end{aligned} \tag{8}$$

Note that this primal/dual pair looks slightly different from the pair (4) and (5). There the primal had non-negativity constraints, and the dual had an inequality. Here the variables of the primal are allowed to be negative, and the dual has equalities. But these are just cosmetic differences; the basic principles are the same.

3 Example #1: Zero-Sum Games

Consider a 2-player zero-sum game defined by an n-by-m payoff matrix R for the row player. That is, if the row player plays row i and the column player plays column j then the row player gets payoff R_{ij} and the column player gets $-R_{ij}$. To make this easier on ourselves (it will allow us to simplify things a bit), let's assume that all entries in R are positive (this is really without loss of generality since as pre-processing one can always translate values by a constant and this will just change the game's value to the row player by that constant). We saw we could write this as an LP:

- Variables: $v, p_1, p_2, \ldots, p_n$.
- Maximize v,
- Subject to:

  $p_i \geq 0$ for all rows i,

  $\sum_i p_i = 1$,

  $\sum_i p_i R_{ij} \geq v$, for all columns j.

To put this into the form of (4), we can replace $\sum_j p_j = 1$ with $\sum_i p_i \leq 1$ since we said that all entries in R are positive, so the maximum will occur with $\sum_i p_i = 1$, and we can also safely add in the constraint $v \geq 0$. We can also rewrite the third set of constraints as $v - \sum_i p_i R_{ij} \leq 0$. This then gives us an LP in the form of (4) with

$$\mathbf{x} = \begin{bmatrix} v \\ p_1 \\ p_2 \\ \ldots \\ p_n \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \ldots \\ 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ 1 \end{bmatrix}, \text{and } A = \left[\begin{array}{c|ccc} 1 & & & \\ 1 & & -R^T & \\ \ldots & & & \\ 1 & & & \\ \hline 0 & 1 & \ldots & 1 \end{array}\right].$$

I.e., maximizing $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.

We can now write the dual, following (5). Let $\mathbf{y}^T = (y_1, y_2, \ldots, y_{m+1})$. We now are asking to minimize $\mathbf{y}^T\mathbf{b}$ subject to $\mathbf{y}^T A \geq c^T$ and $\mathbf{y} \geq \mathbf{0}$. In other words, we want to:

- Minimize y_{m+1},
- Subject to:

  $y_1 + \ldots + y_m \geq 1$,

  $-y_1 R_{i1} - y_2 R_{i2} - \ldots - y_m R_{im} + y_{m+1} \geq 0$ for all rows i,

  or equivalently,

  $y_1 R_{i1} + y_2 R_{i2} + \ldots + y_m R_{im} \leq y_{m+1}$ for all rows i.

So, we can interpret y_{m+1} as the value to the row player, and $y_1, \ldots, y_m$ as the randomized strategy of the column player, and we want to find a randomized strategy for the column player that minimizes y_{m+1} subject to the constraint that the row player gets *at most* y_{m+1} no matter what row he plays. Now notice that we've only required $y_1 + \ldots + y_m \geq 1$, but since we're minimizing and the R_{ij}'s are positive, the minimum will happen at equality.

Notice that the fact that the maximum value of v in the primal is equal to the minimum value of y_{m+1} in the dual follows from strong duality. Therefore, the minimax theorem is a corollary to the strong duality theorem.

4 Example #1: Shortest Paths

Duality allows us to write problems in multiple ways, which gives us power and flexibility. For instance, let us see two ways of writing the shortest s-t path problem, and why they are equal.

Here is an LP for computing an s-t shortest path with respect to the edge lengths $\ell(u, v) \geq 0$:

$$\begin{aligned} \max \quad & d_t \\ \text{subject to} \quad & d_s = 0 \\ & d_v - d_u \leq \ell(u, v) \qquad \forall (u, v) \in E \end{aligned} \tag{9}$$

The constaints are the natural ones: the shortest distance from s to s is zero. And if the s-u distance is d_u, the s-v distance is at most $d_u + \ell(u, v)$ — i.e., $d_v \leq d_u + \ell(u, v)$. It's like putting strings of length $\ell(u, v)$ between u, v and then trying to send t as far from s as possible—the farthest you can send t from s is when the shortest s-t path becomes tight.

Here is another LP that also computes the s-t shortest path:

$$\begin{aligned} & \min \textstyle\sum_e \ell(e)\, y_e \\ \text{subject to} \quad & \textstyle\sum_{w:(s,w)\in E} y_{sw} = 1 \\ & \textstyle\sum_{v:(v,t)\in E} y_{vt} = 1 \\ & \textstyle\sum_{v:(u,v)\in E} y_{uv} = \sum_{v:(v,w)\in E} y_{vw} \qquad \forall w \in V \setminus \{s, t\} \\ & y_e \geq 0. \end{aligned} \tag{10}$$

In this one we're sending one unit of flow from s to t, where the cost of sending a unit of flow on an edge equals its length ℓ_e. Naturally the cheapest way to send this flow is along a shortest s-t path length. So both the LPs should compute the same value. Let's see how this follows from duality.

4.1 Duals of Each Other

Take the first LP. Since we're setting d_s to zero, we could hard-wire this fact into the LP. So we could rewrite (9) as

$$\begin{aligned} \max \quad & d_t & & (11)\\ \text{subject to} \quad & d_v - d_u \le \ell(u,v) & & \forall (u,v) \in E, s \notin \{u,v\}\\ & d_v \le \ell(s,v) & & \forall (s,v) \in E\\ & -d_u \le \ell(u,s) & & \forall (u,s) \in E \end{aligned}$$

Moreover, the distances are never negative for $\ell(u,v) \ge 0$, so we can add in the constraint $d_v \ge 0$ for all $v \in V$.

How to find an upper bound on the value of this LP? The LP is in the standard form, so we can do this mechanically. But let us do this from starting from the definition of the dual as the "best upper bound".

Let us define $E_s^{out} := \{(s,v) \in E\}$, $E_s^{in} := \{(u,s) \in E\}$, and $E^{rest} := E \setminus (E_s^{out} \cup \mathrm{E}_s^{in})$. For every arc $e = (u,v)$ we will have a variable $y_e \ge 0$. We want to get the best upper bound on d_t by linear combinations of the the constraints, so we should find a solution to

$$\sum_{e \in E^{rest}} y_{uv} (d_v - d_u) + \sum_{e \in E_s^{out}} y_{sv}\, d_v - \sum_{e \in E_s^{in}} y_{us}\, d_u \ge d_t \tag{12}$$

(this is like $\mathbf{y}^T A \ge c$) and the objective function is to

$$\text{minimize} \sum_{(u,v) \in E} y_{uv}\, \ell(u,v). \tag{13}$$

(This is like $\min \mathbf{y}^T \mathbf{b}$.) Great, the objective function (13) is exactly what we want, but what about the craziness in (12)? Just collect all copies of each of the variables d_v, and it now says

$$\sum_{v \ne s} d_v \left(\sum_{u:(u,v) \in E} y_{uv} - \sum_{w:(v,w) \in E} y_{vw} \right) \ge d_t.$$

First, this must be an equality at optimality (since otherwise we could reduce the y values). Moreover, these equalities must hold regardless of the d_v values, so this is really the same as

$$\begin{aligned} \sum_{u:(u,v) \in E} y_{uv} - \sum_{w:(v,w) \in E} y_{vw} &= 0 \qquad \forall v \notin \{s,t\}. & (14)\\ \sum_{u:(u,t) \in E} y_{ut} - \sum_{w:(t,w) \in E} y_{tw} &= 1. \end{aligned}$$

Summing all these inequalities for all nodes $v \in V \setminus \{s\}$ gives us the missing equality:

$$\sum_{w:(s,w) \in E} y_{sw} - \sum_{u:(u,s) \in E} y_{us} = 1.$$

Finally, observe that since there's flow conservation at all nodes, and the net unit flow leaving s and reaching t, this means we must have a possibly-empty circulation (i.e., flow going around in circles) plus one unit of s-t flow. Removing the circulation can only lower the objective function, so at optimality we're left with one unit of flow from s to t. This is precisely the LP (10), showing that the dual of LP (9) is LP (10), after a small amount of algebra.