CS124 **Lecture 24**

Duality

As it turns out, the max-flow min-cut theorem from last lecture is a special case of a more general phenomenon called *duality*. Basically, duality means that for each maximization problem there is a corresponding minimizations problem with the property that any feasible solution of the min problem is greater than or equal any feasible solution of the max problem. Furthermore, and more importantly, *they have the same optimum*.

Consider the network shown in Figure 24.2, and the corresponding max-flow problem. We know that it can be written as a linear program as follows:

$$
\begin{array}{llllllll}
\max & f_{SA} & +f_{SB} & & & & & \\
 & f_{SA} & & & & & \leq 3 & \\
 & & f_{SB} & & & & \leq 2 & \\
 & & & f_{AB} & & & \leq 1 & \\
 & & & & f_{AT} & & \leq 1 & \qquad\qquad P \\
 & & & & & f_{BT} & \leq 3 & \\
 & f_{SA} & & -f_{AB} & -f_{AT} & & =0 & \\
 & & f_{SA} & +f_{AB} & & -f_{BT} & =0 & \\
 & & & & & & f \geq 0 &
\end{array}
$$

Consider now the following linear program:

$$
\begin{array}{llllllllll}
\min 3y_{SA} & +2y_{SB} & +y_{AB} & +y_{AT} & +3y_{BT} & & & & \\
y_{SA} & & & & & +u_A & & \geq 1 & \\
 & y_{SB} & & & & & +u_B & \geq 1 & \\
 & & y_{AB} & & & -u_A & +u_B & \geq 0 & \qquad\qquad D \\
 & & & y_{AT} & & -u_A & & \geq 0 & \\
 & & & & y_{BT} & & -u_B & \geq 0 & \\
 & & & & & & & y \geq 0 &
\end{array}
$$

This LP describes the min-cut problem! To see why, suppose that the u_A variable is meant to be 1 if A is in the cut with S, and 0 otherwise, and similarly for B (naturally, by the definition of a cut, S will always be with S in the cut, and T will never be with S). Each of the y variables is to be 1 if the corresponding edge contributes to the cut capacity, and 0 otherwise. Then the constraints make sure that these variables behave exactly as they should. For example, the second constraint states that *if A is not with S, then SA must be added to the cut.* The third one states that *if A is with S and B is not* (this is the only case in which the sum $-u_A + u_B$ becomes -1), *then AB must contribute*

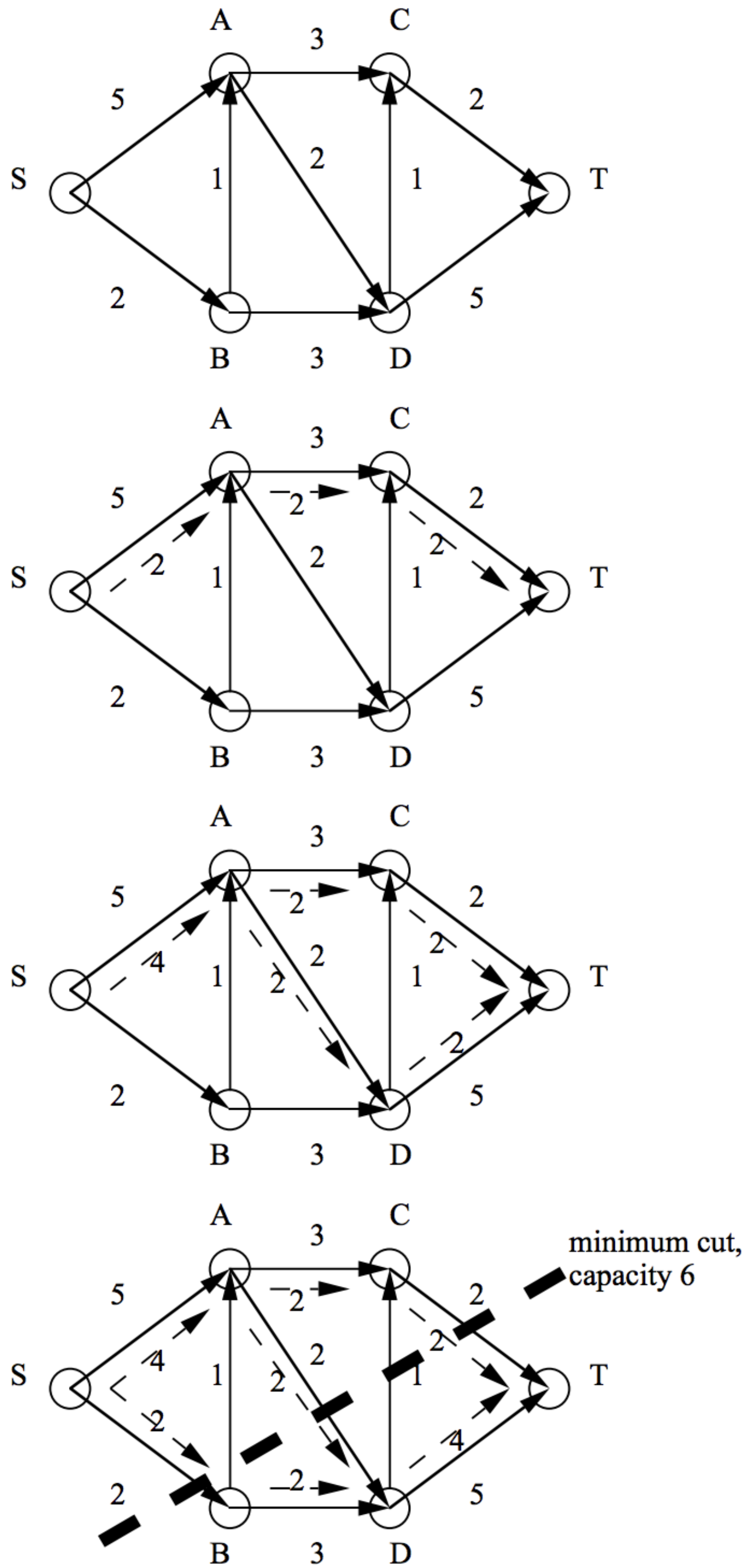

Figure 24.1: Max flow

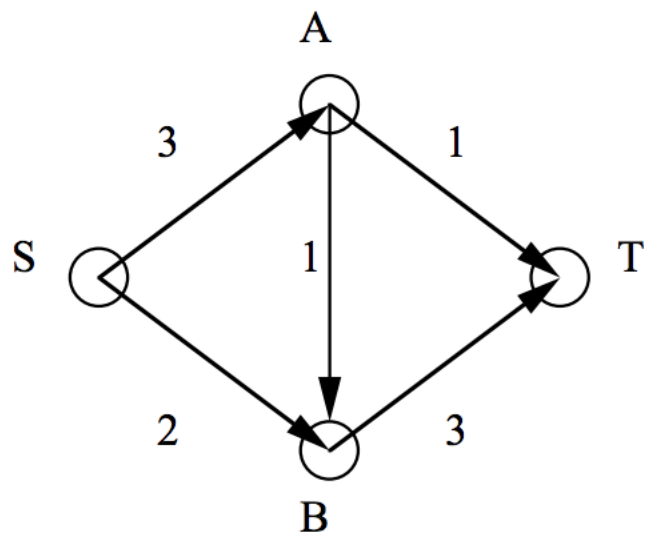

Figure 24.2: A simple max-flow problem

to the cut. And so on. Although the y and u's are free to take values larger than one, they will be "slammed" by the minimization down to 1 or 0.

Let us now make a remarkable observation: these two programs have strikingly symmetric, *dual*, structure. This structure is most easily seen by putting the linear programs in matrix form. The first program, which we call the primal (P), we write as:

max	1	1	0	0	0		
	1	0	0	0	0	$\leq$	3
	0	1	0	0	0	$\leq$	2
	0	0	1	0	0	$\leq$	1
	0	0	0	1	0	$\leq$	1
	0	0	0	0	1	$\leq$	3
	1	0	-1	-1	1	$=$	0
	0	1	1	0	-1	$=$	0
	$\geq$	$\geq$	$\geq$	$\geq$	$\geq$		

Here we have removed the actual variable names, and we have included an additional row at the bottom denoting that all the variables are non-negative. (An unrestricted variable is denoted by unr.)

The second program, which we call the dual (D), we write as:

min	3	2	1	1	3	0	0		
	1	0	0	0	0	1	0	$\geq$	1
	0	1	0	0	0	0	1	$\geq$	1
	0	0	1	0	0	-1	1	$\geq$	0
	0	0	0	1	0	-1	0	$\geq$	0
	0	0	0	0	1	0	-1	$\geq$	0
	$\geq$	$\geq$	$\geq$	$\geq$	$\geq$	unr	unr		

Each variable of P corresponds to a constraint of D, and vice-versa. Equality constraints correspond to unrestricted variables (the u's), and inequality constraints to restricted variables. Minimization becomes maximization. The matrices are transpose of one another, and the roles of right-hand side and objective function are interchanged.

Such LPs are called *dual* to each other. It is mechanical, given an LP, to form its dual. Suppose we start with a maximization problem. Change all inequality constraints into $\leq$ constraints, negating both sides of an equation if necessary. Then

- transpose the coefficient matrix
- invert maximization to minimization
- interchange the roles of the right-hand side and the objective function
- introduce a nonnegative variable for each inequality, and an unrestricted one for each equality
- for each nonnegative variable introduce a $\geq$ constraint, and for each unrestricted variable introduce an equality constraint.

If we start with a minimization problem, we instead begin by turning all inequality constraints into $\geq$ constraints, we make the dual a maximization, and we change the last step so that each nonnegative variable corresponds to a $\leq$ constraint. Note that it is easy to show from this description that the dual of the dual is the original primal problem!

By the max-flow min-cut theorem, the two LPs P and D above have the same optimum. *In fact, this is true for general dual LPs!* This is the *duality theorem*, which can be stated as follows (we shall not prove it; the best proof comes from the simplex algorithm, very much as the max-flow min-cut theorem comes from the max-flow algorithm):

If an LP has a bounded optimum, then so does its dual, and the two optimal values coincide.

Games

We can represent various situations of conflict in life in terms of *matrix games*. For example, the game shown below is the *rock-paper-scissors* game. The Row player chooses a row strategy, the Column player simultaneously (or at least independently) chooses a column strategy, and then Column pays to Row the value at the intersection (if it is negative, Row ends up paying Column).

$$\begin{array}{c c} & \begin{array}{ccc} r & p & s \end{array} \\ \begin{array}{c} r \\ p \\ s \end{array} & \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix} \end{array}$$

Games do not necessarily have to be symmetric (that is, Row and Column have the same strategies, or, in terms of matrices, $A = -A^T$). For example, in the following fictitious *Romney-Clinton* game the strategies may be the issues on which a candidate for office may focus (the initials stand for "foreign policy," "taxes," "health," and "climate") and the entries are the number of voters gained by Romney/Row.

$$\begin{array}{c c} & \begin{array}{cc} h & c \end{array} \\ \begin{array}{c} f \\ t \end{array} & \begin{pmatrix} 3 & -1 \\ -2 & 1 \end{pmatrix} \end{array}$$

We want to explore how the two players may play "optimally" these games. It is not clear what this means. For example, in the first game there is no such thing as an optimal "pure" strategy (it very much depends on what your opponent does; similarly in the second game). But suppose that you play this game repeatedly. Then it makes sense to *randomize*. That is, consider a game given by an $m \times n$ matrix G_{ij}; define a *mixed strategy* for the row player to be a vector $(x_1, \ldots, x_m)$, such that $x_i \geq 0$, and $\sum_{i=1}^m x_i = 1$. Intuitively, x_i is the probability with which Row plays strategy i. Similarly, a mixed strategy for Column is a vector $(y_1, \ldots, y_n)$, such that $y_j \geq 0$, and $\sum_{j=1}^n y_j = 1$.

Suppose that, in the Romney-Clinton game, Row decides to play the mixed strategy $(.5, .5)$. What should Column do? The answer is easy: If the x_i's are given, there is a *pure strategy* (that is, a mixed strategy with all y_j's zero except for one) that is optimal. It is found by comparing the n numbers $\sum_{i=1}^m G_{ij} x_i$, for $j = 1, \ldots, n$ (in the Romney-Clinton game, Column would compare .5 with 0, and of course choose the smallest —remember, the entries denote what Column pays). That is, *if* Column knew Row's mixed strategy, s/he would end up paying the smallest among the n outcomes $\sum_{i=1}^m G_{ij} x_i$, for $j = 1, \ldots, n$. On the other hand, Row will seek the mixed strategy that *maximizes this minimum;* that is,

$$\max_x \min_j \sum_{i=1}^{m} G_{ij} x_i.$$

This maximum would be the best possible *guarantee* about an expected outcome that Row can have by choosing a mixed strategy. Let us call this guarantee z; what Row is trying to do is solve the following LP:

$$\begin{array}{llll} \max z & & & \\ z & -3x_1 & +2x_2 & \leq 0 \\ z & +x_1 & -x_2 & \leq 0 \\ & x_1 & +x_2 & = 1 \end{array}$$

Symmetrically, it is easy to see that Column would solve the following LP:

$$\begin{array}{llll} \min w & & & \\ w & -3y_1 & +y_2 & \geq 0 \\ w & +2y_1 & -y_2 & \geq 0 \\ & y_1 & +y_2 & = 1 \end{array}$$

The crucial observation now is that *these LPs are dual to each other*, and hence have the same optimum, call it V.

Let us summarize: By solving an LP, Row can guarantee an expected income of at least V, and by solving the dual LP, Column can guarantee an expected loss of at most the same value. It follows that this is the uniquely defined optimal play (it was not *a priori* certain that such a play exists). V is called *the value of the game*. In this case, the optimum mixed strategy for Row is $(3/7, 4/7)$, and for Column $(2/7, 5/7)$, with a value of $1/7$ for the Row player.

The existence of mixed strategies that are optimal for both players and achieve the same value is a fundamental result in Game Theory called *the min-max theorem*. It can be written in equations as follows:

$$\max_x \min_y \sum x_i y_j G_{ij} = \min_y \max_x \sum x_i y_j G_{ij}.$$

It is surprising, because the left-hand side, in which Column optimizes last, and therefore has presumably an advantage, should be intuitively smaller than the right-hand side, in which Column decides first. Duality equalizes the two, as it does in max-flow min-cut.

Matching

It is often useful to *compose* reductions. That is, we can reduce a problem A to B, and B to C, and since C we know how to solve, we end up solving A. A good example is the matching problem.

Suppose that the *bipartite* graph shown in Figure 24.3 records the compatibility relation between four boys and four girls. We seek a maximum matching, that is, a set of edges that is as large as possible, and in which no two edges share a node. For example, in Figure 24.3 there is a *complete* matching (a matching that involves all nodes).

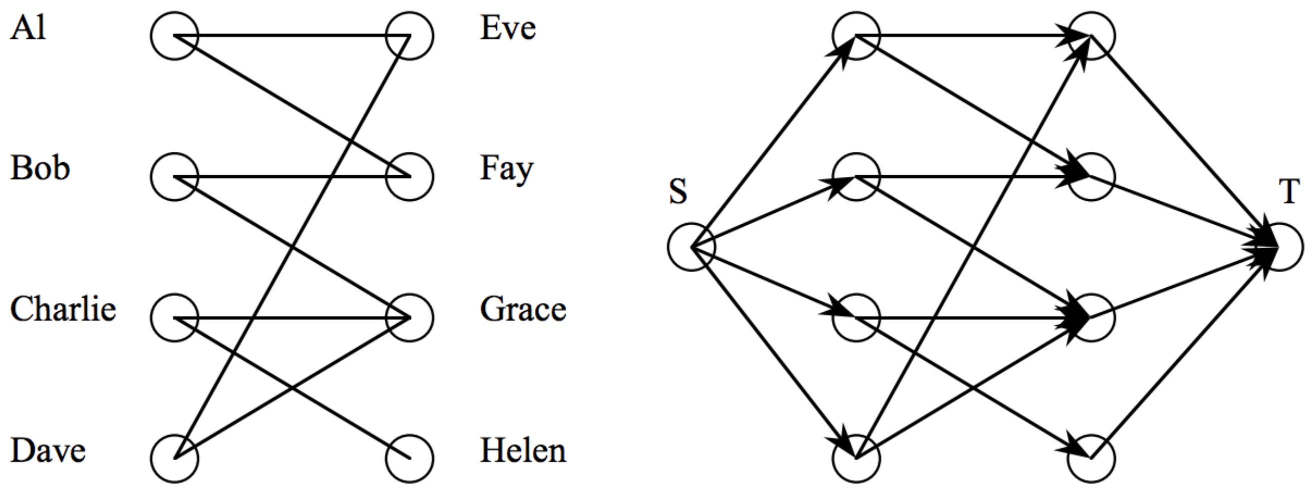

Figure 24.3: Reduction from matching to max-flow (all capacities are 1)

To reduce this problem to max-flow, we create a new source and a new sink, connect the source with all boys and all girls with the sinks, and direct all edges of the original bipartite graph from the boys to the girls. All edges have capacity one. It is easy to see that the maximum flow in this network corresponds to the maximum matching.

Well, the situation is slightly more complicated than was stated above: what is easy to see is that the optimum *integer-valued* flow corresponds to the optimum matching. We would be at a loss interpreting as a matching a flow that ships .7 units along the edge Al-Eve! Fortunately, what the algorithm in the previous section establishes is that *if the capacities are integers, then the maximum flow is integer.* This is because we only deal with integers throughout the algorithm. Hence *integrality comes for free in the max-flow problem.*

(Contrast this with linear programming and integer linear programming, where integrality not only doesn't come free, but adding the constraint that solutions must be integers turns a polynomial-time solvable problem into an NP-hard one.)