We have seen a number of situations so far where two optimization problems are paired, and approach the same optimal solution in a different way:

- BIN-PACKING: Input items with sizes, input bin size, compute minimum number of bins needed
- DUAL-BIN-PACKING: Input items with sizes, input number of bins, compute minimum bin size that suffices to fit all the items
- KNAPSACK: Input items with weights and values, input weight target, compute maximum value obtainable within weight target
- DUAL-KNAPSACK: Input items with weights and values, input value target, compute minimum weight needed to meet value target

In each case the optimization depends on two variables, and the transformation from the problem to its dual depends on interchanging the roles of the two variables.

Consider our general linear programming *minimization* problem in slack form, where the equality constraints have been absorbed into the redefinition of this **primal problem**:

- variable n-vector x with $x \geq \mathbf{0}$
- minimize cost $C \cdot x$
- inequality constraints $Ax \geq b$, A an m by n matrix, b an m-vector

The **dual linear program** is as follows:

- variable m-vector y with $y \geq 0$
- maximize benefit $y \cdot b$
- inequality constraints $yA \leq C$, A the same m by n matrix, C an n-vector

Remember our example of the **dieter's problem**, where x is a vector of food amounts, A_{ij} indicates how much of nutrient i is in a unit of food j, b is a list of the required amounts of each nutrient, and C gives the cost per unit of each food. The dieter wants to satisfy the nutrient constraints with the minimum total cost.

We can call the dual problem to this one the **vitamin seller's problem**. The vitamin seller wants to pick an m-vector y that gives the price per unit of a supplement for each nutrient. The seller wants to set the price so that the dieter will buy supplements instead of food. So no food must be able to provide any nutrient at a cheaper price than set by y. We look at the n-vector yA, which gives the price, using y, of replacing the nutrients in each food – we must have $ya \leq C$. Given this constraint, the vitamin seller wants to maximize their total revenue, which is $y \cdot b$.

Recall the **maximum flow** problem, in a network with n nodes and m edges:

- We have a flow vector f of length m.
- We have an m by n matrix B, where B_{eu} is -1 if e flows out of u, 1 if e flows into u, and 0 otherwise. So fB is an n-vector giving the net flow into each vertex.
- We have an n-vector d, with a -1 for s, 1 for t, and 0 for each other vertex. Our goal is to maximize $(fB) \cdot d$.
- We have an m-vector c, giving the capacity of each edge.

Let's now try to squeeze this problem into the primal and dual LP form. We'll reset n to be the number of intermediate vertices (not s or t). Starting with the dual, which corresponds to the original max flow problem:

- We will set y to be f.
- We will set A to be an m by $n + m$ matrix that has B in its first n columns and the identity matrix I_m in the rest.
- We set C to be an $(n + m)$ vector with n zeros followed by our former capacity vector c.
- The constraint $yA \leq C$ now means that $fB \leq 0$ and $f \leq c$, as before.
- Our objective function to maximize is now defined as $y \cdot b$ where b is an m-vector that has ones for each edge going out of s.

Now we look at the primal LP for this dual:

- Our $(n+m)$-vector x is a pair (u,w) where u is a weight function on the intermediate vertices and w a weight function on the edges.
- With the same $A = (B, I)$ as before, the constraint $Ax \geq b$ now means $Bu + w \geq b$. For each edge $e = (i,j)$, we have $w_e - u_j + u_i \geq 0$, unless $i = s$ in which case $w_e - u_j \geq 1$.
- The objective function $C \cdot x$ is now $c \cdot w$.

(Note a typo in Adler p. 154, with $w_{ij} \leq \ldots$ instead of $w_{ij} \geq \ldots$) Formulated correctly, which is somewhere between what I have here and what the Adler notes have, the constraints require that u be extended to a non-negative function on all vertices, including s and t, with $u_t - u_s \geq 1$ and w_e for each edge e being at least $u_j - u_i$, The optimal solution for this turns out to be 0/1 valued in both u and w, with $u_i = 0$ for vertices on one side of a cut and $u_i = 1$ on the other. Then $w_e = 1$ for edges that cross the cut, and the weighted size of the cut is $w \cdot c$, the figure we are minimizing.

In general the primal problem (minimize $C \cdot x$ subject to $Ax \geq b$ and $x \geq 0$) is related to the dual problem (maximize $y \cdot b$ subject to $yA \leq C$ and $y \geq 0$) by the following:

Weak Duality Theorem: If x_0 is the optimal solution to the primal and y_0 is the optimal solution to the dual, then $C \cdot x_0 \geq y_0 b$.

Proof: Because y_0 is feasible for the dual, $y_0 A \leq C$ and so $y_0 A x_0 \leq C x_0$ by multiplication on the right. Because x_0 is feasible for the primal, $Ax_0 \geq b$, and hence $y_0 A x_0 \geq y_0 b$ by multiplication on the left. This gives us $Cx_0 \geq y_0 b$ by transitivity.

In the vitamin example, we have that the minimum cost to buy food to meet the nutrition requirements must be at least the maximum revenue from vitamins to replace the food, given that each of the vitamins is sold at a competitive price.

In fact the two optimal values are not only related in this way but are the same. This is an example of the:

Strong Duality Theorem: If the primal and dual both have solutions, the values of their objective functions are equal. If the primal is infeasible, then the dual is unbounded, and vice versa.

We can thus choose one of the problems to solve and get the solution to the other – in fact there is a **primal-dual method** that attacks both problems simultaneously to see which is easier to solve.

We won't prove the general Strong Duality Theorem here, but we'll prove a special case that contains the key geometric idea:

Farkas' Lemma: Let x be a variable column m-vector, y a variable row n-vector, b a fixed n-vector, and A an m by n matrix. Then exactly one of the following two sets of constraints is solvable:

$$\begin{aligned} (1): &\quad Ax = b, x \geq \mathbf{0} \\ (2): &\quad yA \geq \mathbf{0}, y \cdot b < 0 \end{aligned}$$

Proof: If (1) and (2) were both solvable by some x and y, we would have $yAx = yb$ from (1) and multiplication, but $(yA)x$ is the product of two non-negative vectors and $yb < 0$ is given by (2), a contradiction.

Suppose that (1) has no solution – we will find a solution to (2). Let S be the **cone** given by the set $\{Ax : x \geq \mathbf{0}\}$. By our assumption, b is not in S.

Since S is a closed and convex set, there must exist a hyperplane with S on one side and b on the other. Numerically, we can call this hyperplane $\{y : y \cdot v = \alpha\}$ for some fixed vector v and scalar α.

By multiplying both v and α by -1 if necessary, we can ensure that $b \cdot v < \alpha < y \cdot v$ for any y in S.

Since $\mathbf{0} \in S$, we know $\alpha < 0$. Since $v \cdot y > \alpha$ for any $y \in S$, $vAw > \alpha$ for any vector w with $w \geq \mathbf{0}$. But then the vector vA must satisfy $vA \geq \mathbf{0}$, since if it had any negative entry we could pick a w to make vAw arbitrarily small.

Now v satisfies $vA \geq \mathbf{0}$ and $v \cdot b < \alpha < 0$, satisfying (2).

I'll post a practice exam shortly. The format will be similar to that of the midterm except:

- I will try to make it considerably shorter, because we must deal with a fairly hard time limit of two hours. (I can and will give you a few minutes before and after the scheduled slot, but not very much.)

- Along with the ordinary questions I will have several "true-false with explanation" questions. These will consist of a statement which you should be able to *argue* is true or false based on the material of the course. So if it said "If the MST problem is **NP**-complete, then $\mathbf{P} = \mathbf{NP}$", you would say "true" and then argue that (1) MST is in **P**, and (2) if any problem is both in **P** and in **NP**-complete then $\mathbf{P} = \mathbf{NP}$. These subarguments could be made convincingly in about a sentence each. The truth of the statement should not depend on unproven assumptions like $\mathbf{P} \neq \mathbf{NP}$.

Before the midterm we dealt with three main topics, which will be the subject collectively of about a quarter of the final:

- Basic techniques of analysis, asymptotic notation, recurrences, the Master Theorem, divide and conquer algorithms
- Greedy algorithms and matroids
- Dynamic programming and shortest-path algorithms, including matrix multiplication methods

Since these areas are covered more lightly on the exam, it is less likely that more subtle points would be the subject of a large question, though the threshold for being relevant enough for a true/false question is lower.

There were four main topics after the midterm, each of which will be the subject of 15-20% of the final exam:

- Randomized Algorithms: Examples and analysis
- **NP**-completeness theory, proving problems to be **NP**-complete, basic **NP**-complete problems
- Approximation algorithms, the classes of approximation problems (FPTAS, PTAS, constant, log, and polynomial)
- Linear Programming: Definitions, the simplex algorithm, and duality