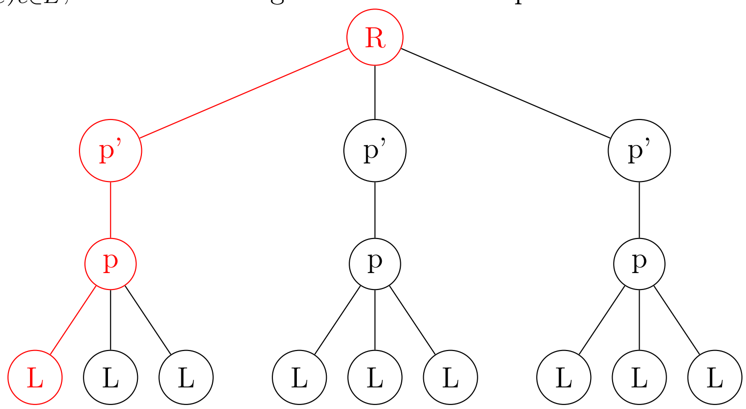
CS170 Fall 2013 Solutions to Homework 8

Zackery Field, section Di, 103, `cs170-fe`

November 8, 2013

1. (10 pts.) Longest path in a tree

Give a linear time algorithm that, given an undirected tree $T = (V, E)$ with edges weights $(w_e)_{e \in E}$, returns the weight of the heaviest path in T.



For some tree T with root R, and leaf vertices L. Let p be the parent of a given set of leaf vertices and p' be the parent of vertex p. Choose the largest edge weight $p - L_i$ and update the value of $p' - p$ to: $(p' - p) = (p' - p) + \max\{\forall_i p - L_i\}$.

The smallest subproblems are the set of vertices p s.t. each p_i is the parent of a leaf vertex. We can preprocess the vertices V (in an adjacency list) and determine the set of nodes p that satisfy the above condition by looking collecting the immediate parents of all of the leaf nodes in a new set H.

Continued on Page 7

2. (10 pts.) Problem 7.2, Mathewheatical modeling

Duckwheat is produced in Kansas and Mexico and consumed in New York and California. Kansas produces 15 shnupells of duckwheat and Mexico 8. Meanwhile, New York consumes 10 shnupells and California 13. The transportation costs per shnupell are \$4 from Mexico to New York, \$1 from Mexico to California, \$2 from Kansas to New York, and \$3 and from Kansas to California. Write a linear program that decides the amounts of duckwheat (in shnupells and fractions of a shnupell) to be transported from each producer to each consumer, so as to minimize the overall transportation cost.

$$\begin{aligned}
p_{k,n} &= \text{Kansas to New York} = \$2, p_{k,c} = \text{Kansas to California} = \$3 \\
p_{m,n} &= \text{Mexico to New York} = \$4, p_{m,c} = \text{Mexico to California} = \$1 \\
x_{k,n} &= \text{amt form Kansas to New York}, x_{k,c} = \text{amt form Kansas to California} \\
x_{m,n} &= \text{amt form Mexico to New York}, x_{m,c} = \text{amt form Mexico to California}
\end{aligned}$$

$$\begin{aligned}
\text{min: } & x_{k,n}p_{k,n} + x_{k,c}p_{k,c} + x_{m,n}p_{m,n} + x_{m,c}p_{m,c} \\
& x_{k,n} + x_{k,c} \leq 15 \\
& x_{m,n} + x_{m,c} \leq 8 \\
& x_{k,n} + x_{m,n} \geq 10 \\
& x_{k,c} + x_{m,c} \geq 13
\end{aligned}$$

3. (10 pts.) Problem 7.7, A feasibility study

Find necessary and sufficient conditions on real numbers a and b under which the linear program

$$\begin{aligned} \text{max: } & x + y \\ ax + by &\leq 1 \\ x, y &\geq 0 \end{aligned}$$

Assume that $x, y \in \mathbb{R}$
y-intercept: $1/b$
x-intercept: $1/a$

(a) Is infeasible.

Since both x and y could be 0, for any choice of a and b, $a * 0 + b * 0 \leq 1$. Therefore, no choice of a and b exists that makes the solution to max: $x + y$ infeasible.

(b) Is unbounded.

$a \leq 0 \vee b \leq 0$. Let $(-)$ denote some nonpositive number, and $(+)$ denote some nonnegative number, these possible scenarios occur

$$\begin{aligned} (-)x + (-)y &\leq 1 \\ (+)x + (-)y &\leq 1 \\ (-)x + (+)y &\leq 1 \end{aligned} \tag{1}$$

For equation (1) any choice of x and y will satisfy the inequality. For equation (2), holding x at some value that satisfies the inequality will allow y to be any number, making the solution unbounded. Equivalently, for equation (3), holding y at some value that satisfies the inequality will allow x to be any number, making the solution unbounded.

(c) Has a unique and optimal solution.

$a > 0 \wedge b > 0$. As before, let $(-)$ denote some nonpositve number, and $(+)$ denote some nonnegative number. There is only one possible configuration in this case $(+)x + (+)y \leq 1$. Since there is a nonnegative constraint on x and y, then there is some constraint on the maximum values of x and y, Per the argument in the infeasibility section, there is no choice of a and b that would be small enough to make the solution unbounded.

4. (10 pts.) Problem 7.12, Provably optimal

For a linear program

$$\begin{aligned} \text{max: } & x_1 - 2x_3 \\ x_1 - x_2 &\le 1 \\ 2x_2 - x_3 &\le 1 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

Show that the solution $(x_1, x_2, x_3) = (3/2, 1/2, 0)$ is optimal.

To show that this solution is optimal I will construct a dual LP and show that it has the same optimal value $x_1 - 2x_3 = 3/2 - 2 * 0 = 3/2$.

Multiplier	Inequality
y_1	$x_1 - x_2 \le 1$
y_2	$2x_2 - x_3 \le 1$

$$(y_1)(x_1 - x_2) + (y_2)(2x_2 - x_3) \le y_1 + y_2$$

$$(y_1)x_1 + (2y_2 - y_1)x_2 - (y_2)x_3 \le y_1 + y_2$$

$$x_1 - 2x_3 \le y_1 + y_2 \text{ if } \begin{cases} y_1, y_2 & \ge 0 \\ y_1 & \ge 1 \\ 2y_2 - y_1 & \ge 0 \\ y_2 & \ge -2 \end{cases}$$

$$\begin{aligned} \text{min: } & y_1 + y_2 \\ y_1, y_2 &\ge 0 \\ y_1 &\ge 1 \\ 2y_2 - y_1 &\ge 0 \\ y_2 &\ge -2 \end{aligned}$$

By inspection, the maximum solution to this dual LP is $(y_1, y_2) = (1, 1/2) \Rightarrow 3/2$. By the duality theorem, the primal (given) LP and the dual LP both have the same optimum value, which gurantees that the given solution $(x_1, x_2, x_3) = (3/2, 1/2, 0)$ is optimal.

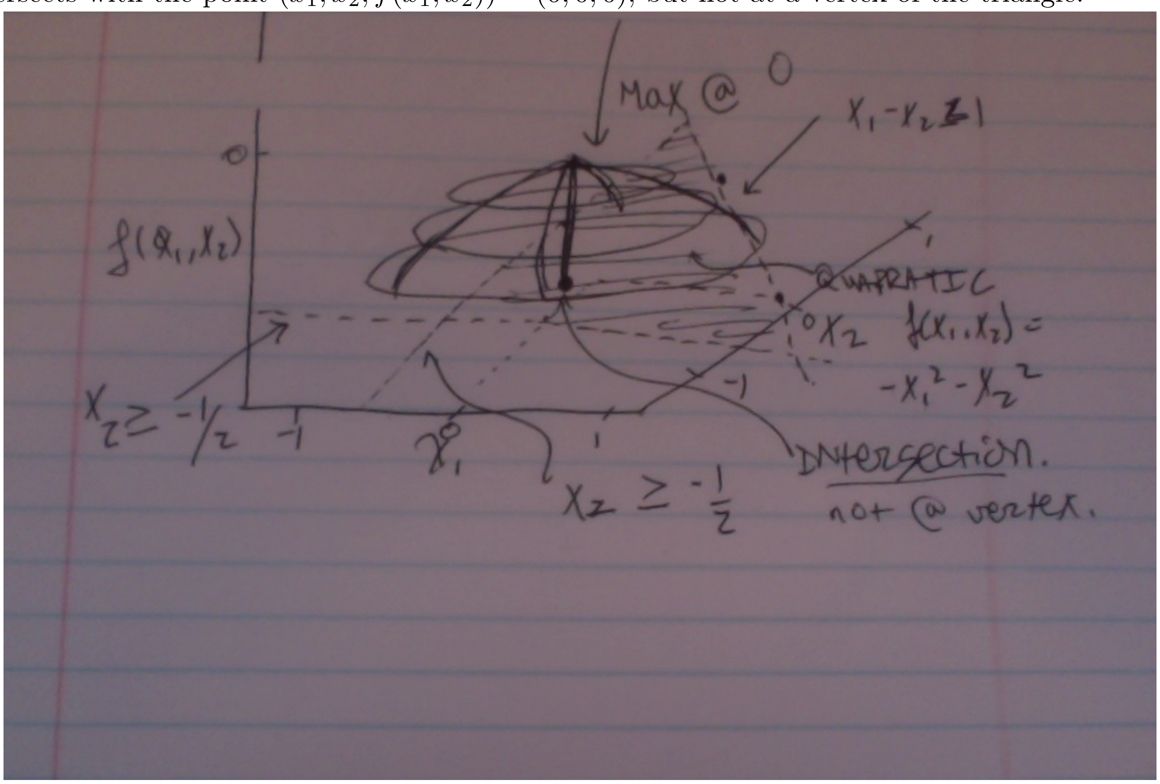
5. (10 pts.) Problem 7.9, Non-linear programming

A quadratic programming problem seeks to maximize a quadratric objective function (with terms like $3x_1^2, 5x1x2$) subject to a set of linear constraints. Give an example of a quadratic program in two variables x_1, x_2 such that the feasible region is nonempty and bounded, and yet none of the vertices of this region optimize the (quadratic) objective.

$f(x_1, x_2) = -x_1^2 - x_2^2$, subject to constraints

$$\begin{aligned} x_1 &\geq -1/2 \\ x_2 &\geq -1/2 \\ x_2 - x_1 &\leq 1 \end{aligned}$$

The maximum of $f(x_1, x_2)$ is at $f(0, 0) = 0$. The 3 constraints form a triangle that intersects with the point $(x_1, x_2, f(x_1, x_2)) = (0, 0, 0)$, but not at a vertex of the triangle.



6. (10 pts.) Problem 7.21, Flows through cuts

An edge of an $s - t$ flow network is called *critical* if decreasing the capacity of this edge results in a decrease in the $s - t$ maximum flow. Give an efficient algorithm that finds a critical edge in a network. - Run the max-flow algorithm on the given network (takes $O(|V||E|^2)$ time p. 203 in txt). By the max-flow min-cut theorem, the max-flow algorithm will return the minimum (s,t)-cut (see p.202 in text). This cut defines the maximum flow capacity. Any edge in the minimum (s,t)-cut with positive capacity ($s \to t$ directionality) will be a critical edge, because the reduction of this edge's capacity will reduce the maximum possible flow from $s - t$. Proof: see section 7.2 in the text.

Runtime of max-flow defined in p.203 of the text: $O(|V||E|^2)$.

Extra space for Problem 1

Continued from Page 1

Starting with the list H and moving upward through the tree always updating the edge $v' - v$ (v' is the parent of v) with the maximum of the edge weights of all of the paths $v - v_{children}$. This will lead to the heaviest weight path being stored in one of the edges connecting the root to its children. To return the actual path that led to this heaviest path, start by assuming that all edge weights are distinct, and then simply follow the edges downward the tree by maximum value back to a leaf node.

Proof by induction: The base case is the set of vertices H. Take one of these vertices p. If we want to find the maximum edge weight extending from p, then we can trivially observe the largest edge weight extending from p to its children (which are leaf nodes). Let v be some arbitrary vertex, with parent v', and children v_c. We can hypothesize that all vertices $v - v_c$ have been updated with the maximum edge weights to each v_c. It follows that the process of updating the edge $v' - v$ with the maximum of the weights $v - v_c$ will require that $v' - v$ now contains the maximum edge weight path.

The preprocessing of the grpah to find the set of vertices H, where each vertex in H only has leaf vertex children, takes linear time. Since each path is looked at exactly once in order to find the maximum, and then the parent path is updated once at most for each parent edge, then the total running time is linear; $O(|V| + |E|)$.

: