CPS 590.1 - Linear and integer programming

Homework 2: Multi-unit combinatorial auctions (due Nov. 7 before class)

Please read the rules for assignments on the course web page. Contact Vince (conitzer@cs.duke.edu) with any questions. All questions have the same number of points.

In this assignment, we again consider combinatorial auctions.[1] In class, the combinatorial auctions that we considered were "single-unit" combinatorial auctions, in the sense that there was only one unit (copy) of each item. Now, we consider multi-unit combinatorial auctions. In these, for each item i, we have some number of units (copies) q_i of this item for sale.

For example, suppose there are two items (or, if you prefer, two types of item), A and B. We have 10 units of item A for sale, and 5 units of item B for sale. We have the following bids:

- $(\{10A, 4B\}, 11)$
- $(\{8A, 1B\}, 8)$
- $(\{5A, 10B\}, 10)$

This means, for example, that the first bidder wants to buy 10 units of A and 4 units of B, and is willing to pay 11 for this. Throughout this assignment, we will allow fractional acceptance of bids—for example, it is possible to accept 1/2 of the first bid, giving the first bidder 5 units of A and 2 units of B, and collecting 5.5 from this bidder. In fact, for simplicity, throughout this assignment, you are also allowed to accept a bid *more than once*. For example, it is possible to accept the second bid 1.25 times, giving that bidder 10 units of A and 1.25 units of B, and collecting 10 from this bidder. Of course, we cannot accept this bid (say) 1.3 times, because we do not have enough units of A for this.

[1] In auction theory terms, we consider "first-price" combinatorial auctions, meaning that every winning bidder simply pays what she bid. We will not concern ourselves with strategic/game-theoretic bidding, etc., in this assignment; we focus on the winner determination problem (just as we did in class).

1. Give a linear program formulation for the winner determination problem, trying to maximize the value that we can collect from the bidders under the constraint that we cannot allocate more units of an item than we have available. Note that you should give a *general* formulation, not just one for the 3-bid example above. (You will use this throughout the assignment, so make it clean and elegant...)

2. Give the dual of this linear program formulation. **Interpret** it in terms of prices, and **give** an intuitive argument for weak duality (that is, why does a feasible solution to the dual give you an upper bound on what you can obtain?).

3. Solve the specific 3-bid example given above by using the simplex algorithm (by hand) on the linear program formulation, starting from the dictionary where every (non-slack) variable is set to 0. You can use whatever pivoting rule you like (think before you pivot!). For the final dictionary, you only need to write down enough of it to prove that you are done. You should **state** what the optimal primal solution is, *and* what the optimal dual solution is.

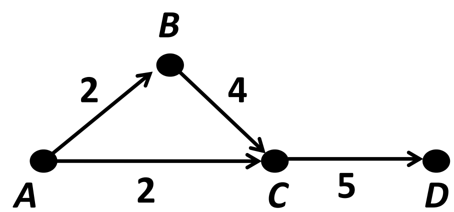

Figure 1: The graph for an instance of the network combinatorial auction problem.

We now consider a version of the multi-unit combinatorial auction winner determination problem where there are exponentially many bids. We will call this problem the "network combinatorial auction problem." Specifically, we consider a directed graph G with vertices V and edges E. The edges are the items, and for every edge e, there are q_e units available of this edge. For every bidder b, there is an "starting vertex" s_b and a "target vertex" t_b. Bidder b bids v_b for *every* path from s_b to t_b, that is, for every bundle of items (edges) corresponding to a path between s_b and t_b. (We can think of this as follows: a single bidder places multiple bids; there is one bid for each path.) Again, a bid can be accepted fractionally; it can also be accepted more than once.

As an example, consider the graph in Figure 1. We have 2 units of edge AB, 2 of AC, 4 of BC, and 5 of CD. Now suppose we have the following 3 bidders:

- $s_1 = A$, $t_1 = D$, $v_1 = 10$
- $s_2 = A$, $t_2 = C$, $v_2 = 6$
- $s_3 = C$, $t_3 = D$, $v_3 = 3$

We can reformulate this in terms of the original multi-unit combinatorial auction winner determination problem as follows. The first bidder really places two bids, corresponding to the two different paths from A to D. One of these bids is $(\{1AB, 1BC, 1CD\}, 10)$, and one of them is $(\{1AC, 1CD\}, 10)$. (Note that the coefficients on the edges within the bids must *always* be 1 in a network combinatorial auction.) The third bidder corresponds to only a single bid, namely $(\{1CD\}, 3)$. It is easy to see that the optimal solution here is to accept the first bidder 4 times (namely, 2 times $(\{1AB, 1BC, 1CD\}, 10)$ and 2 times $(\{1AC, 1CD\}, 10)$), and the third bidder 1 time, for a total of $4 \cdot 10 + 1 \cdot 3 = 43$.

4. To solve these instances, we can still use the same linear program as the one that you developed in the first questions of this assignment; the only problem is that there are exponentially many bids, one for every path. (The number of items is fine; we can easily enumerate them.) To make this work, **describe** a simple (and efficient) algorithm for finding the most violated constraint. (Do you need to find the most violated constraint in the primal or in the dual?)

5. Instead of using your general combinatorial auction winner determination problem LP formulation with the help of constraint/column generation, we can also create a more specific and direct LP formulation for the network combinatorial auction problem that uses only polynomially many variables and constraints. **Give** such a formulation; make it as elegant as possible. (For once, you do *not* have to give the dual... Not that it isn't interesting...)

6. We know that for the combinatorial auction problem *in general*, it is possible to give examples where, even though the input numbers are all integers, the only optimal solution is fractional (bids are accepted only partially)—this is true even in the single-unit case. Is it also possible to give such an example in the network combinatorial auction problem, or is there an integrality theorem for this problem? **If** the former, **give** such an example. **If** the latter, **prove** it. You will get no points for simply guessing without an explanation.