IEORE4008: Computational Discrete Optimization 04/22/2020

Assignment 5

Deadline: 05/06/2020, h: 8pm

Exercise 1
The *Stable Roommate Problem* is a modification of the Stable Marriage problem in which the underlying graph is not restricted to be bipartite. Recall that Gale-Shapley algorithm implies that, in the Stable Marriage case, a stable matching always exists.

a) Give an example showing that there are instances of the Stable Roommate Problem where a stable matching does not exist.

b) Give an example showing that a stable matching in the Stable Roommate Problem can be of half the size of a maximum matching (i.e., a matching of maximum cardinality, that ignores the preference lists of nodes). Can it be smaller than half? Why?

Exercise 2
Consider the following version of `Scale-bid` for the AdWords problem:

Input: A set U of nodes, and a budget B_u for each node in U.
Algorithm: At each step, assign the incoming node v_i to a neighbor u with enough residual capacity (if any) that maximizes $w_{uv_i} \times \frac{RB_u}{B_u}$, where RB_u denotes the residual budget of node u.

a) Run the algorithm of the instance from Figure 1.

b) Can you provide an instance where `Scale-bid` performs better than both `Greedy` and `Balance`?

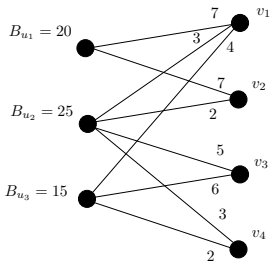

Figure 1: In this istance, assume that node arrive in this order: $v_1, v_2, v_1, v_3, v_2, v_4$.

Exercise 3
In this exercise, you are going to test three algorithms for the *AdWords* problem. There are 5 companies (numbered with integers from 1 to 5), each with budget 200, and an horizon of $t = 100$. There are three kinds of words:

- A common word, for which company i bids i ($i = 1, \ldots, 5$);
- A less common word, for which company i bids $2 \times i$ ($i = 1, \ldots, 5$);
- A rare word, for which each company i bids $4 \times i$ ($i = 1, \ldots, 5$).

Test the performance of algorithm `Greedy`, `Balance`, and `Scale-Bid` on the following instance:

- The common word appears 60 times, then the less common word appears 30 times, then the rare word appears 10 times;

- The words appear randomly, where the common word has 60% probability of appearing, the less common 30%, and the rare 10%.

For each of the instances above, report the profit obtained by each algorithm (for the second instance, report the average over $m = 50$ repetitions). For `Scale-Bid`, use the optimal scaling function explained in class, i.e.:

> At each step, assign the incoming node v_i to a neighbor u with enough residual capacity (if any) that maximizes $w_{uv_i} \times (1 - e^{-\frac{RB_u}{B_u}})$, where RB_u denotes the residual budget of node u.