Introduction to Algorithms	Problem Set 5
CS 4820 Spring 2016	Lillyan Pan

1 Problem 2

1.1 Algorithm

Assume costs are given as lists and the graph is created as an adjacency list. Let V be the set of groups. Certain pairs of groups will be neighbors, and so let E be the set of all pairs of neighboring groups. Hence, we have undirected graph $G = (V, E)$. For each group i, there is the renting cost r_i if group i moves into the new building with group A and a cost h_i that it will stay in the old building with group H. Assume $r_i, h_i \geq 0$. Note, $h_i = 0$, as there is no renting cost for staying in the original building. Additionally, for each pair, there is a collaboration penalty $c_{ij} \geq 0$ if i and j are not in the same building. Note, if groups i, j don't collaborate, $c_{ij} = 0$. To construct $G' = (V', E')$ where V' is the set of V groups with additional nodes s, t

1. Label H as the source s and A and the sink t
2. Create directed edges from s to each group i, (s, i) with capacity r_i and from each group i to t, (i, t) with capacity h_i. Note, edge (s, t) will have capacity 0.
3. Model each neighboring pair (i, j) with two directed edges $(i, j), (j, i)$ each with capacity c_{ij}.
4. For an $s - t$ cut (X, Y) which partitions the groups into sets X, Y, group the edges that cross the cut (X, Y) into the following:
   - Edges (s, j), where $j \in Y$, will contribute r_i to the capacity of the cut, as they will be in the same building as A
   - Edges (i, t), where $t \in X$, will contribute h_i to the capacity of the cut, as they will be in the same building as H
   - Edges (i, j), where $i \in X$ and $j \in Y$, will contribute c_{ij} to the capacity of the cut
5. Calculate
$$c(A, B) = \sum_{i \in X} h_i + \sum_{j \in Y} r_j + \sum_{(i,j) \in E, |Ai,j|=1}^{n} c_{ij} = q'(X, Y)$$
6. Run a minimum-cut algorithm and return the cut that minimizes $q'(X, Y)$

1.2 Time and Space Complexity

Let $|V| = n$, and $|E| = m$. Step 1-2 is bounded by $O(2n + 1) = O(n)$ as we create edges from s to every group and every group to t as well as edge (s, t). Step 3 is bounded by $O(m)$ as we are adding an additional edge to each pair in constant time. Step 3 is bounded by $O(m + n)$ as we iterate through all the edges to assign capacities. The minimum-cut algorithm is given to run in $O(mC)$ time, where C is a bound for the maximum possible flow value (edges out of s would be completely saturated with flow). Hence, total running time is polynomial. Because the time complexity is polynomial, the space complexity is polynomial as well.

1.3 Correctness

Lemma 1. *The solution can be obtained by a minimum-cut algorithm in the graph G' constructed above. For a minimum cut (X', Y'), the partition (X, Y) obtained by deleting s and t minimizes the separation value $q'(X, Y)$.*

Proof of Lemma 1. The flow network is set so that the capacity of cut (X, Y) exactly measures the quantity $q'(X, Y)$, which represents the total cost of dividing groups along the cut. The three kinds of edges crossing the cut (X, Y) correspond to the three kinds of terms in the expression for $q'(X, Y)$

- $\sum\limits_{i \in X} h_i$: edges from nodes in side X to the sink, which represent the renting cost = 0 for staying in the old building
- $\sum\limits_{j \in Y} r_j$: edges from the source to nodes in side Y, which represent the renting cost for going to the new building
- $\sum\limits_{(i,j) \in E, |Ai,j|=1}^{n} c_{ij}$: edges involving neither the source nor sink, which represent collaboration costs between groups if there exists one.

Note, nodes in side X represent groups that stay in the old building, while nodes in side Y represent groups that moved to the new building. Hence, if we want to minimize $q'(X, Y)$, we find a cut of minimum capacity. Thus, through solving this minimum-cut problem, the minimum cut represents the minimum cost way to split groups across the two buildings.