Algorithm **Spring 2017**

Solution : Assignment 3

Lecturer: Li Guoqiang *Homework taker: Yu Han*

Due Time: May 11

Problem 1. Prove the Konig theorem: Let G be bipartite, then cardinality of maximum matching = cardinality of minimum vertex cover.

Solution:

We use C_{match} donates the cardinality of maximum matching and C_{cover} donates the cardinality of minimum vertex cover.

M is a maximum matching for G. Then no vertex in a vertex cover can cover more than one edge of M, for M is a set of edges no two of which share an endpoint. Thus:

$$C_{cover} \geq |M| = C_{match}$$

Let vertex set V be partitioned into left set L and right set R. let U be the set of unmatched vertices in L (possibly empty), and let Z be the set of vertices that are either in U or are connected to U by alternating paths (paths that alternate between edges that are in the matching and edges that are not in the matching). Let

$$K = (L \setminus Z) \cup (R \cap Z)$$

K forms a vertex cover and every vertex in K is an endpoint of a matched edge. Thus:

$$C_{cover} \leq C_{match}$$

Thus:

$$C_{cover} = C_{match}$$

■

Problem 2. Use layering to get a factor f approximation algorithm for set cover, where f is the frequency of the most frequent element. Provide a tight example for this algorithm.

Solution:

1.$\mathcal{S}_0 = \mathcal{S}, C = \emptyset, i = 0$

2.Remove S where $S \cap U = \emptyset$, say this set is $\mathcal{S}_i$

3.Compute $c = min\{w(S)/|S|\}$ for all $S \in \mathcal{S}_i$

4.Let $t_i(S) = c \cdot |S|$ and $w(S) = w(S) - t_i(S)$ for all $S \in \mathcal{S}_i$

5.Let $W_i = \{S \in \mathcal{S}_i | w(S) = 0\}, C = C \cup W_i$

6.Let $U = U / \bigcup_{S \in W_i}$, Increase i by 1 and goto step 2 until U is empty set.

Tight Example: $S_1 = S_2 = \cdots = S_m = U$. The optimistic solution will choose only one subset, and the approximate solution will choose all the subsets. ■

Problem 3. Given an undirected graph. The problem is to remove a minimum number of edges such that the residual graph contains no triangle. (I.e., there is no three vertices a, b, c such that edges (a, b),(b, c),(c, a) are all in the residual graph.) Give a factor 3 approximation algorithm.

Solution:

Repeatedly search the graph G. If there is a triangle (u,v,w), remove edge (u,v),(v,w) and (u,w) from G.

■

Problem 4. Given n points in $\mathbb{R}^2$, define the optimal Euclidean Steiner tree to be a minimum length tree containing all n points and any other subset of points from $\mathbb{R}^2$. Prove that each of the additional points must have degree three, with all three angles being 120 .

Solution:

If a additional point has degree one, this additional point is obviously redundant.

If a additional point has degree two, this additional point is also redundant for triangle inequality.

If a additional point has degree more than two, it should be Fermat points or else we always could construct a smaller Steiner tree.

■

Problem 5. Consider a more restricted algorithm than First-Fit, called Next-Fit, which tries to pack the next item only in the most recently started bin. If it does not fit, it is packed in a new bin. Show that this algorithm also achieves factor 2. Give a factor 2 tight example.

Solution:

Suppose we have a set $I = \{1, 2, ..., n\}$ of items and each item is of size $s(i) \in (0, 1]$. Let k be the number of bins found by NEXT-Fit and the corresponding assignment is denoted by $A : I \rightarrow \{1, 2, ...k\}$. Let $k^\star$ be the optimal solution.

$$k^\star \geq \lceil \sum_{i \in I} s(i) \rceil$$

For bins $j = 1, 2, ..., \lfloor k/2 \rfloor$, we have

$$\sum_{i:A(i) \in \{2j-1, 2j\}} s(i) > 1$$

Combining inequalities:

$$\lfloor \frac{k}{2} \rfloor < \sum_{i \in I} s(i)$$

Thus

$$\frac{k-1}{2} \leq \lfloor \frac{k}{2} \rfloor \leq \lceil \sum_{i \in I} s(i) \rceil - 1$$

$$k \leq 2 \cdot \lceil \sum_{i \in I} s(i) \rceil - 1 \leq 2 \cdot k^\star - 1$$

■

Problem 6. Given an undirected complete graph, each edge is assigned with a nonnegative cost by the function c. Find a Hamilton cycle with the largest cost by the greedy approach, and prove the guarantee factor is 2.

Solution:

Find the max spanning tree T of graph G. By doubling its edges we obtain an Eulerian graph connecting all vertices. Find an Euler tour of this graph, for example by traversing the edges in DFS (depth first search) order. Next obtain a Rudrata cycle by traversing the Euler tour and short-cutting vertices and previously visited vertices ■