Assignment 6

Xiaoting Li (xil139)
Ziyu Zhang (ziz41)
Deniz Unal (des2014)

January 30, 2019

12. Consider the following problem. The input consists of a directed graph $G = (V, E)$, a designated sink vertex $t \in V$, a collection $S \subset V$ of source vertices, and a profit p_v for each vertex $v \in S$. A feasible solution is a subset T of S such that there exists a collection of vertex disjoint paths from the elements of T to t in G. The objective is to maximize the aggregate profit of the elements of T. So think about the following application: the set S is clients, the profits are how much each client is willing to pay for connectivity, and the graph is a computer network that can only support one connection per router, and the goal is to make as much profit as possible by selling collectivity. Give a polynomial time algorithm for this problem. Analyze the run time of the algorithm if it is implemented in a straight-forward way using known algorithms.

Answer: We can reduce this problem to max flow problem. The max flow we need is the maximum matching in a bipartite graph. So we can further reduce this problem to looking for a maximum matching in a bipartite graph using matroid intersection. If we introduce an artificial 'super source' vertex x into our original graph and connect it to each original source vertex $s \in S$ with directed edges with infinite capacity. We represent each source vertex $s \in S$ with two vertices s_{in} and s_{out}. Every incoming edge to s would be coming to s_{in} and every outgoing edge from s would be going from s_{out}. And we would connect s_{in} and s_{out} with an edge with capacity equal to weight/profit of S. We rewrite the problem using matroid representation, $I_1 = \{F \subset E | \forall s_{in} \in S_1 : |F \cap \delta(s_{in}) \leq 1\}$. $I_2 = \{F \subset E | \forall s_{out} \in S_2 : |F \cap \delta(s_{out}) \leq 1\}$. $S_1 \cup S_2$ are the vertices in S and the intersection $I_1 \cup I_2$ is the set of the matching. And in this problem, what we need to do is find a maximum matching.We can run Edmonds-Karp implementation of Ford Fulkerson method to find the max flow from our artificial source x to the designated sink t, since the maximum flow would be equal to maximum matching.

13.Consider the following problem. The input consists of a directed

graph $G = (V, E)$, a designated sink vertex $t \in V$, a collection $S \subset V$ of source vertices, and a time limit L. A feasible solution is a subset T_i of S for each $i \in [1, L]$ such that for each T_i there exists a collection of vertex disjoint paths from the elements of T_i to t in G. The objective is to the aggregate number of elements of S that are in at least one T_i, that is max $|U_{i=1}^{L} T_i|$. So intuitively you want to provide as many customers as possible unit time connectivity within a time window of L units of time. Give a polynomial time algorithm for this problem. Analyze the run time of the algorithm if it is implemented in a straight-forward way using known algorithms.

Answer: We can reduce this problem to a max flow problem by using matroid intersection to find the maximum matching in a bipartite graph. The first matroid $M_1 = (E, I_1)$, I_1 is collection of customers, which are vertices in S. The second matroid $M_2 = (E, I_2)$, I_2 is collection of time slots L. So what we need to do is have edges between I_1 and I_2 to get maximum matching (max flow) using matroid intersection. The solution could be from the source collection S, pick as many number of subset of S such that those subset of S could be connected to sink with the largest time limit that is less or equal than L. And then do this pick up procedure iteratively, so that at the end, we found for each $i \subset [1, L]$. Since the graph have limited number of S, the time for each of the step is poly-time.