

CS499 Homework 10 (First Draft)

Interstellar

Exercise 10.1

Since

$$\sum_{v \in V} f(s, v) = \sum_{v \in V \setminus S} f(s, v) + \sum_{v \in S} f(s, v)$$

we only need to prove that

$$\sum_{v \in S} f(s, v) = \sum_{u \in S - s, v \in V \setminus S} f(u, v)$$

Since

$$\sum_{v \in S} f(s, v) = - \sum_{v \in S - s} f(v, s)$$

we only need to prove that

$$\sum_{u \in S - s, v \in s + V \setminus S} f(u, v) = 0$$

It is obvious to see that

$$\sum_{u \in S - s, v \in S - s} f(u, v) = 0$$

So, we only need to prove that

$$\sum_{u \in S - s, v \in s + V \setminus S} f(u, v) + \sum_{u \in S - s, v \in S - s} f(u, v) = \sum_{u \in S - s, v \in V} f(u, v) = 0$$

According to the definition,

$$\sum_{u \in S - s, v \in V} f(u, v) = 0$$

Done.

Exercise 10.2

Define the minimum cut between i and j as $\text{minCut}(i, j)$. According to the Max Flow Min Cut Theorem, $\text{minCut}(s, r) \geq k$, $\text{minCut}(r, t) \geq k$. Obviously, $\text{minCut}(s, t) \geq \min\{\text{minCut}(s, r), \text{minCut}(r, t)\} \geq k$, which means there is a flow from s to t of value k .

Exercise 10.4

Suppose there is a s - t -path in G that has less than k edges. Then, at least one edge in the path moves more than one level forward, which contradicts Definition 10.3. So, $\text{dist}(s, t) \geq k$.

Exercise 10.5

Since $\text{dist}(s, t) = k$, we have a path $s \rightarrow u_1 \rightarrow u_2 \cdots u_{k-1} \rightarrow t$. For convenience, we call $s : u_0, t : u_k$. We construct $V_0, V_1 \cdots V_K$ as follows:

Step1: $u_0 \in V_0, u_1 \in V_1, u_2 \in V_2, \dots, u_{k-1} \in V_{K-1}, u_k \in V_K$.

Step2: $\forall v \in V \setminus V_k$, if there is an edge (v, t) , let V_{k-1} contain v . Obviously, $\forall n, 0 \leq n \leq k-2, u_n$ can't link to t . Otherwise, we can skip u_{k-1} and get a shorter path from s to t , which contradicts $\text{dist}(s, t) = k$.

054 Step3: $\forall v \in V \setminus (V_k \cup V_{k-1})$, if $\exists \text{edge}(v, w), w \in V_{k-1}$, let V_{k-2} contain v . $\forall n, 0 \leq n \leq k-3, u_n$
055 can't link to any vertex in V_{k-1} .
056 \vdots
057
058 Step k : $\forall v \in V \setminus \bigcup_{i=2}^k V_i$, if $\exists \text{edge}(v, w), w \in V_2$, let V_1 contain v . Similarly, u_0 can't link to any
059 vertex in V_2 .
060 Step $k+1$: $\forall v \in V$ and $v \notin V_i, i \neq 1, 2, 3, \dots, k, v \in V_0$.
061
062 Therefore, if $\text{dist}(s, t) = k$, (G, s, t, c) has a k -layering.
063
064 **Exercise 10.6**
065 Obviously, (1) $s \in V_0$ and (2) $t \in V_k$ are satisfied, we consider condition (3). Since (G, s, t, c) is
066 a flow network and V_0, V_1, \dots, V_k is an optimal layering, every edge in G moves at most one level
067 forward and $\text{dist}_G(s, t) = k$. And we denote the path p as $s \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_{k-1} \rightarrow t$.
068 Then we can get $s \in V_0, v_1 \in V_1, \dots, v_i \in V_i, \dots, v_{k-1} \in V_{k-1}$ and $t \in V_k$. Each edge is between
069 two adjacent layerings. Since residual network G_f only add a reverse edge of each edge in p , these
070 additional edges are also between two adjacent layerings. So V_0, V_1, \dots, V_k satisfy condition (3) and
071 it is a layering of (G_f, s, t, c_f) .
072
073 **Exercise 10.7**
074 Because there is a path from s to t , we suppose $\text{dist}_G(s, t) = k$. Based on the construction method
075 in 10.5, we can find a k -layering and it is an optimal layering. Therefore, every network (G, s, t, c)
076 has an optimal layering, provided there is a path from s to t .
077
078 **Exercise 10.8**
079 We consider each while-loop of EK algorithm. In every iteration, EK algorithm choose p to be a
080 shortest s-t-path in G_f . And we denote the path p as $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_{k-1} \rightarrow v_k$ and
081 $v_0 = s \in V_0, v_1 \in V_1, \dots, v_i \in V_i, \dots, v_{k-1} \in V_{k-1}$ and $v_k = t \in V_k$. Then, EK algorithm
082 routes c_{\min} flow along p . In $G, \exists (v_i, v_{i+1}) \in p$ where $i \in 0, 1, 2, \dots, k, c_f(v_i, v_{i+1}) = c_{\min}$. In G_f ,
083 $c'_f(v_i, v_{i+1}) = 0$ and $c'_f(v_{i+1}, v_i) = c_{\min}$.
084 Obviously, if V_0, V_1, \dots, V_k is still an optimal layering, $c_f(v_i, v_{i+1})$ is always 0. Otherwise, there is a
085 feasible s-t-path from v_{i+1} to v_i , which is impossible. Therefore, in every iteration, the total number
086 of the edges, which are from V_i to $V_{i+1} (\forall i \in 0, 1, 2, \dots, k)$ and in feasible s-t-path ($\text{dist}(s, t) = k$),
087 will minus at least one. Obviously, these edges are less than or equal to m . So, after m iteration,
088 there is no feasible s-t-path which $\text{dist}(s, t) = k$, and $\text{dist}(s, t)$ will be larger than k . So, after at
089 most m iterations of the while-loop, V_0, V_1, \dots, V_k ceases to be an optimal layering.
090
091 **Exercise 10.9**
092 According to **Exercise 10.8**, a particular layering is no more optimal after at most m iterations.
093 Since a layering is at least 1-layering and at most n -latering, after at most $m * n$ iterations, there is
094 no optimal layering, which means there is no s-t-path, the algorithm terminates.
095
096 **Exercise 10.10**
097 According to **Exercise 10.9**, the Edmonds-Karp algorithm terminates after nm iterations of the
098 while-loop, which is to say, we can get the max flow f after finite steps by Edmonds-Karp algo-
099 rithm.
100
101 **Question**
102 Can we use the Edmonds-Karp algorithm in multi-source network?
103
104
105
106
107