Mathematical Foundations
of
Computer Science

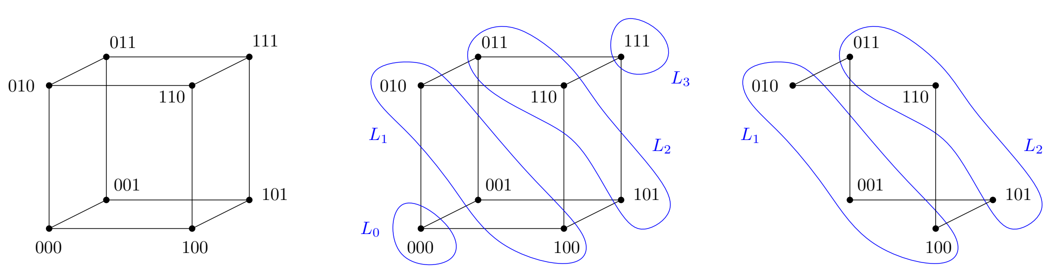
CS 499, Shanghai Jiaotong University, Dominik Scheder

11 Matchings and Network Flow

- Homework assignment published on Monday 2018-05-14
- Submit questions and first solutions by Sunday, 2018-05-20, 12:00
- Submit final solution by Sunday, 2018-05-27.

11.1 Matchings

Consider the Hamming cube $\{0,1\}^n$. We can view it as a graph H_n, where the vertex set is $\{0,1\}^n$ and two vertices x, y are connected by an edge if x and y differ in exactly one coordinate. Define the k^{th} layer to be $L_k := \{x \in \{0,1\}^n \mid |x|_1 = k\}$, where $|x|_1$ denotes the number of 1s in x. Note that the subgraph induced by layer k and layer $k+1$ is a bipartite graph $H_n[L_k \cup L_{k+1}]$. See the picture below for an illustration ($n = 3, k = 1$):



Exercise 11.1. Let $0 \leq k < n/2$. Show that the bipartite graph $H_n[L_k \cup L_{k+1}]$ has a matching of size $|L_k| = \binom{n}{k}$.

Answer We can get the maximum matching by maximum flow and minimum cut.

Construct a network flow G, s and t as the start node and the end node. Let $(S, \bar{S})$ be a minimum cut, because every node in L_k can reach L_{k+1} and so as L_{k+1}, then all the vertices except s and t are all in S or all in $\bar{S}$, otherwise $cap(S, \bar{S})$ should be an infinite number.

Here we can regard L_k as U, regard L_{k+1} as V. According to the proof of $K\ddot{o}nig$'s Theorm, we can divide U and V into four parts: $U \cap \bar{S}$, $V \cap \bar{S}$, $U \cap S$(called A) and $V \cap S$(called B). The vertex cover is $|U \setminus A| + |B|$. So the minimum cut has two situations:

The first situation is that S contain s and other vertices, $t \in \bar{S}$. Since $|U \setminus A|$ is empty and $|B| = |V|$, in this situation, we can know that only $|V|$ is the vertex cover.

The second situation is that S contain only s, $\bar{S}$ contain t and other nodes. Because $|U \setminus A| = |U|$ and $|B|$ is empty, in this situation $|U|$ is the vertex cover.

Because $|V|$ is always not smaller than $|U|$, the minimum vertex cover is $|U|$ which is also the maximum matching by $K\ddot{o}nig$'s Theorm.

Exercise 11.2. Let $G = (V, E)$ be a bipartite graph with left side L and right side R. Suppose G is d-regular (every vertex has degree d), so in particular $|L| = |R|$. Show that G has a perfect matching (that is, a matching M of size $|L|$).

Proof. Suppose M is a match of G and C is a cover, $|L| = |R| = n$, then $|E| = n \cdot d$. Since each vertex covers at most d edges, $|C| \geq n$. What's more, we know that L is a cover. So $max|M| = min|C| = n$, and G has a perfect matching. □

Exercise 11.3. Let G a d-regular bipartite graph. Show that the edges $E(G)$ can be partitioned into d perfect matchings. That is, there are matchings $M_1, \ldots, M_d \subseteq E(G)$ such that (1) $M_i \cap M_j = \emptyset$ for $1 \leq i < j \leq d$ and (2) $M_1 \cup M_2 \cup \cdots \cup M_d = E(G)$.

Proof. If G is a d-regular bipartite graph, we can do the following two steps:

1. Find a matching M which size is n. The matching is exist, which we have proved in the previous exercise.

2. Delete M from E. Then G should be a $d-1$-regular bipartite graph.

Since after the two steps G is still a regular bipartite graph, we can do this d times until $|E| = 0$, and we will get d different matching M, let them be $M_1, \dots, M_d \subseteq E(G)$, then they meet the conditions $M_i \cap M_j = \emptyset$ for $1 \le i < j \le d$ and $M_1 \cup M_2 \cup \cdots \cup M_d = E(G)$. □

11.2 Networks with Vertex Capacities

Suppose we have a directed graph $G = (V, E)$ but instead of *edge capacities* we have *vertex capacities* $c : V \to \mathbb{R}$. Now a flow f should observe the *vertex capacity constraints*, i.e., the outflow from a vertex u should not exceed $c(u)$:

$$\forall u \in V : \sum_{v \in V, f(u,v) > 0} f(u,v) \le c(u) \ .$$

Exercise 11.4. Consider networks with vertex capacities.

1. Show how to model networks with vertex capacities by networks with edge capacities. More precisely, show how to transform $G = (V, E, c)$ with $c : V \to \mathbb{R}^+$ into a network $G' = (V', E', c')$ with $c' : E' \to \mathbb{R}^+$ such that every s-t-flow f in G that respects the vertex capacities corresponds to an s-t-flow f' (of same value) in G' that respects edge capacities, and vice versa.

2. Draw a picture illustrating your solution.

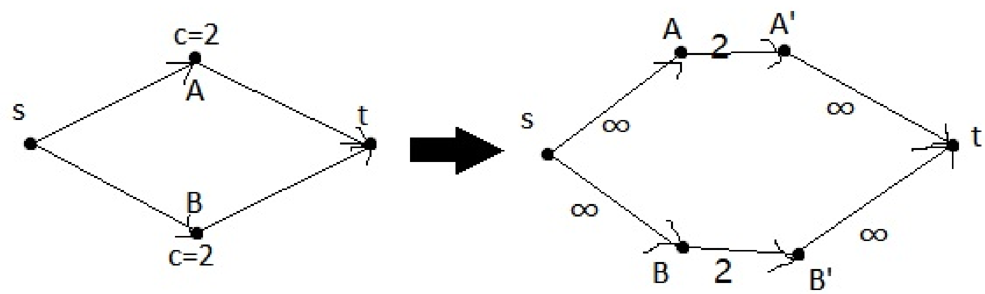
3. Show that there is a polynomial time algorithm solving the following problem: Given a directed graph $G = (V, E)$ and two vertices $s, t \in V$. Are there k paths $p_1, \dots, p_k$, each from s to t, such that the paths are *internally vertex disjoint*? Here, internally vertex disjoint means that for $i \ne j$ the paths p_i, p_j share no vertices besides s and t.

Answer

1. For every vertex v, define set I contains all the edges (u, v), $u \ne v$, and set O contain all the edges (v, u), $u \ne v$. And give these edges an

infinite capacity. Then eliminate all the edges in O from v, and insert a new vertex v', change the edges (v, u) in O into (v', u). Finally insert an edge (v, v') with capacity $c(u)$. In this way, we model a network with vertex capacities by networks with edge capacities.

2.



3. For every vertex v, give it a capacity of 1. Then we get a network with vertex capacities. Now model the network with edge capacities with the method we give. Run Edmonds-Karp algorithm on the network, and if the maximum flow equals to k, we know the paths are internally disjoint.

Exercise 11.5. Let H_n be the n-dimensional Hamming cube. For $i < n/2$ consider L_i and L_{n-i}. Note that $|L_i| = \binom{n}{i} = \binom{n}{n-i} = L_{n-i}$, so the L_i and L_{n-i} have the same size. Show that there are $\binom{n}{i}$ paths $p_1, p_2, \ldots, p_{\binom{n}{i}}$ in H_n such that (i) each path p starts in L_i and ends in L_{n-i}; (ii) two different paths p, p' do not share any vertices.

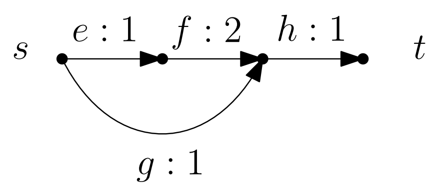
Answer We first prove another stronger proposition: the whole subsets A_i of the set $\{1, 2, \ldots, n\}$ can be divided as k_n disjoint symmetrical paths $A_1 \subset A_2 \subset \cdots \subset A_t, (|A_1| + |A_t| = n$ and $|A_{i+1}| - |A_i| = 1)$. Now we use induction to prove this. Assume $n = 1$, it's obviously correct. Assume this assumption also satisfy $n - 1$ and we can get a path $(0)A_1 \subset A_2 \subset \cdots \subset A_t$ is one of the k_{n-1} symmetrical paths. Now let's construct another two paths for n: $(1)A_1 \subset A_2 \subset \cdots \subset A_t \subset A_t \bigcup \{n\}, (2)A_1 \bigcup \{n\} \subset A_2 \bigcup \{n\} \subset \cdots \subset A_t \subset A_{t-1} \bigcup \{n\}$. It is obvious that both (1) and (2) are symmetrical paths. It is obvious that (1) and (2) are disjoint and different (0) will lead to different (1) and (2). Suppose A is a subset of $\{1, 2, \ldots, n\}$, if $n \notin A$, then A must in a path similar to (1) and A can not be in a path like (2). If $n \in A$, then $A-\{n\}$ is in a path like (0). When it equals A_t, A is in a path like (1). When it doesn't equal A_t, A is in a path like (2). Now we have proved this proposition and let's go back to our exercise 11.5.

We can regard the subsets mentioned above as vertices in our Hamming cube. (If one element in the subset is chosen then the corresponding position in the vertice code is set to 1, otherwise it will be set to 0.) Then the symmetrical paths mentioned above is just a path in the Hamming cube. When we are finding the paths between L_i and L_{n-i}, for every vertices in L_i, it must be on one of the paths and every two vertices can not be on the same path, as is proved above. Since some of the paths might start from $L_u(u \leq i)$ and end at L_{n-u}, we can just delete these redundant vertices. Therefore, there are $\binom{n}{i}$ paths from L_i to L_{n-i}.

11.3 Always, Sometimes, or Never Full

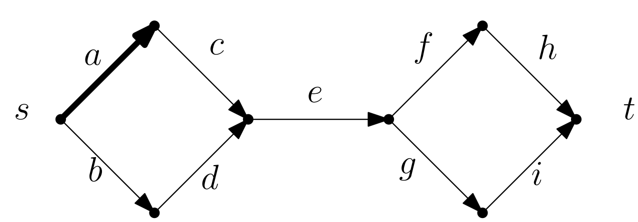
Let (G, s, t, c) be a flow network, $G = (V, E)$. A directed edge $e = (u, v)$ is called always full if $f(e) = c(e)$ for every maximum flow; it is called sometimes full if $f(e) = c(e)$ for some but not all maximum flows; it is called never full if $f(e) < c(e)$ for all maximum flows.

Let $(S, V \setminus S)$ be a cut. That is, $s \in S, t \in V \setminus S$. We say the edge $e = (u, v)$ is crossing the cut if $u \in S$ and $v \in V \setminus S$. We say e is always crossing if it crosses every minimum cut; sometimes crossing if it crosses some, but not all minimum cuts; never crossing if it crosses no minimum cut. For example, look at this flow network:



Example network: the edges e, g are sometimes full and never crossing; f is never full and never crossing; h is always full and always crossing.

Exercise 11.6. Consider this network:



The fat edge a has capcity 2, all other edges have capacity 1.

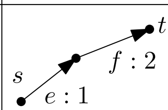
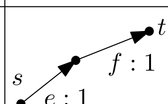
1. Indicate which edges are (i) always full, (ii) sometimes full, (iii) never full.

2. Indicate which edges are (i) always crossing, (ii) sometimes crossing, (iii) never crossing.

Answer

1. (i) e
(ii) b,c,d,f,g,h,i
(iii) a
2. (i) e
(ii) none
(iii) all the other edges

Exercise 11.7. An edge e can be (x) always full, (y) sometimes full, (z) never full; it can be (x') always crossing, (y') sometimes crossing, (z') never crossing. So there are nine possible combinations: (xx') always full and always crossing, (xy') always full and sometimes crossing, and so on. Or are there? Maybe some possibilities are impossible. Let's draw a table:

<table>
<tr><th>The edge e is:</th><th>x: always full</th><th>y: sometimes full</th><th>z: never full</th></tr>
<tr><td>x': always crossing</td><td>
</td><td>Possible
or
impossible?</td><td>Possible
or
impossible?</td></tr>
<tr><td>y': sometimes crossing</td><td>
</td><td>Possible
or
impossible?</td><td>Possible
or
impossible?</td></tr>
<tr><td>z': never crossing</td><td>Possible
or
impossible?</td><td>Possible
or
impossible?</td><td>Possible
or
impossible?</td></tr>
</table>

The nine possible cases, some of which are maybe impossible.

The two very simple flow networks in the table already show that (xx') and (yy') are possible; that is, it is possible to be always full and always crossing, and it is possible to be always full and sometimes crossing. Fill out the table! That is, for each of the remaining seven cases, find out whether it is possible or not. If it is possible, draw a (simple) network showing that it is possible; if impossible, give a proof of this fact.

Lemma 11.8. *The edges across the minimum cut is always full in the maximum flow.*

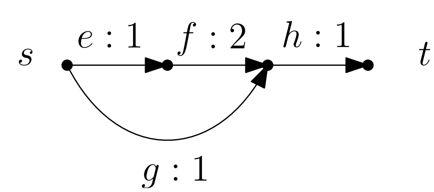
Proof. Let E' be the set of edges across a minimum cut of G. Since maximum flow must flow through (some of) the edges in E' and the capacity of maximum flow is equal to that of minimum cut, the edges in E' must be full. □

Answer

$(x'y), (x'z), (y'y), (y'z)$ are impossible due to the lemma above(if there's an edge across a minimum cut, it must be "always full").

(xz') is impossible. An edge is never crossing means if this edge is removed, the capacity of minimum cut (and maximum flow) of the graph won't change. However an edge is always full means every maximum flow of the graph flows through this edge. If the edge is removed, the capacity of the maximum flow must be decreased. Therefore, an edge cannot be both "never crossing" and "always full".

$(z'y)$ and $(z'z)$ are possible: e.g.



Example network: the edges e, g are sometimes full and never crossing; f is never full and never crossing.