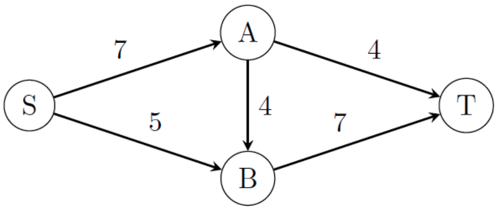
Homework 7

Name: 方嘉聪 ID: 2200017849

Problem 1. (Max Flow, Min Cut, and Duality).

In this exercise, we will demonstrate that LP duality can be used to show the max-flow min-cut theorem.

Consider the following instance of max flow:



Let f_1 be the flow pushed on the path $\{S, A, T\}$, f_2 be the flow pushed on the path $\{S, A, B, T\}$, and f_3 be the flow pushed on the path $\{S, B, T\}$. The following is an LP for max flow in terms of the variables f_1, f_2, f_3:

$$\begin{aligned}
\max\ & f_1 + f_2 + f_3 \\
f_1 + f_2 &\leqslant 7 && \text{(Constraint for } (S, A)) \\
f_3 &\leqslant 5 && \text{(Constraint for } (S, B)) \\
f_1 &\leqslant 4 && \text{(Constraint for } (A, T)) \\
f_2 &\leqslant 4 && \text{(Constraint for } (A, B)) \\
f_2 + f_3 &\leqslant 7 && \text{(Constraint for } (B, T)) \\
f_1, f_2, f_3 &\geqslant 0
\end{aligned}$$

(a) Find the dual of this LP, where the variables in the dual are x_e for each edge e in the graph.

(b) Show that the dual of the LP for any max-flow problem is an LP for the corresponding min-cut problem. ◀

Answer. (a) 把原问题的线性规划写成矩阵形式:

$$\text{maximize} \quad \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

$$\text{subject to} \quad \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \leq \begin{pmatrix} 7 \\ 5 \\ 4 \\ 4 \\ 7 \end{pmatrix}$$

$$f_1, f_2, f_3 \geq 0$$

那么对偶问题为:

$$\text{minimize}\quad \begin{pmatrix}7 & 5 & 4 & 4 & 7\end{pmatrix}\begin{pmatrix}x_{(S,A)}\\ x_{(S,B)}\\ x_{(A,T)}\\ x_{(A,B)}\\ x_{(B,T)}\end{pmatrix}$$

$$\text{subject to}\quad \begin{pmatrix}1 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 1\\ 0 & 1 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x_{(S,A)}\\ x_{(S,B)}\\ x_{(A,T)}\\ x_{(A,B)}\\ x_{(B,T)}\end{pmatrix} \geq \begin{pmatrix}1\\ 1\\ 1\end{pmatrix}$$

$$x_{(S,A)}, x_{(S,B)}, x_{(A,T)}, x_{(A,B)}, x_{(B,T)} \geq 0$$

其中 $x_e, \forall e \in E$ 表示边 e 是否在割集中 (是为 1, 否为 0), 约束即为每条 s-t 路径至少有一条边在割中, 那么对偶问题是一个整数线性规划, 可以看出对偶问题是对应的最小割问题.

(b) 考虑一个任意的有向图 $G = (V, E)$, 源点 s, 汇点 t, 边容量 $c(e)$. 设

$$P = \{p \mid p \text{ is a simple path from } s \text{ to } t\}$$

注意到如果一个流有环, 那么去掉环不会使得流变差, 故我们这里只考虑简单路径. 那么类似第一问, 最大流问题的线性规划形式为 (f_p 表示路径 p 的流):

$$\begin{aligned} \text{maximize}\quad & \sum_{p\in P} f_p \\ \text{subject to}\quad & \sum_{p\in P:(u,v)\in p} f_p \leq c(u,v), \quad \forall (u,v)\in E \\ & f_p \geq 0, \quad \forall p \in P \end{aligned}$$

写成矩阵形式有 (不妨设 $|P| = m, |E| = n$):

$$\begin{aligned} \text{maximize}\quad & \mathbf{1}^T \boldsymbol{f} \\ \text{subject to}\quad & \boldsymbol{A}\boldsymbol{f} \leq \boldsymbol{c} \\ & \boldsymbol{f} \geq 0 \end{aligned}$$

其中 $\boldsymbol{f} = (f_i)_m, \boldsymbol{c} = (c_i)_n$, $\boldsymbol{A} = (A_{ij})_{n\times m}$, A_{ij} 表示边 e_i 是否在路径 p_j 上. 那么对偶问题为

$$\begin{aligned} \text{minimize}\quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to}\quad & \boldsymbol{A}^T \boldsymbol{x} \geq \mathbf{1} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

其中 $\boldsymbol{x} = (x_i)_n$ 表示边 e_i 是否在割集中, 则最小割问题等价对偶的整数线性规划. 那么下面问题在于这一整数线性规划和原对偶线性规划的解是否相同.

我们可以证明整数网络流问题的约束矩阵 $\boldsymbol{A}$ 是一个全幺模矩阵 (TUM)[1], 那么由课上提到的整数流定理[2]可知, 整数线性规划和原对偶线性规划的解是相同的. 证毕. ◁

Problem 2. (Network Flow with Vertex Capacities).
Let $G = (V, E)$ be a directed graph with a source vertex $s \in V$ and a sink vertex $t \in V$. Whereas the standard network flow problem involves capacities for edges, here we suppose instead that every vertex $v \in V$ has an integer capacity $c_v \geqslant 0$. A *vertex-capacitated* flow in G is a function $f : E \to [0, +\infty)$ such that

- (Capacity Constraint) For each vertex $v \in V$, we have

$$\sum_{e \text{ into } v} f(e) \leqslant c_v \text{ and } \sum_{e \text{ out of } v} f(e) \leqslant c_v$$

- (Conservation Constraint) For each vertex $v \in V/\{s, t\}$, we have

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

As usual, the value of a flow is defined as $\sum_{e \text{ out of } s} f(e)$. Give an efficient algorithm to find a maximum vertex-capacitated flow in G from s to t and analyze its running time. ◀

Answer. 如下构造新图 $G' = (V', E')$:

- 对任意 $v \in V$, 对应的在 G' 中加入 v_{in}, v_{out} 两个点, 并添加 $c(v_{in}, v_{out}) = c_v$ 的边.
- 对任意 $e = (u, v) \in E$, 对应的在 G' 中加入边 (u_{out}, v_{in}), 令 $c(u_{out}, v_{in}) = +\infty$.

由此得到了一个新图 G', 其源点与汇点分别为 $s' = s_{in}, t' = t_{out}$. 在 G' 中运行 Ford-Fulkerson 算法, 得到的最大流在原图 G 对应的流即为最大顶点容量流. 时间复杂度为 $O(|V| \cdot |E|^2)$. ◁

Problem 3. (Restoring the Balance!).
We are given a network $G = (V, E)$ whose edges have integer capacities $c(e)$, and a maximum flow f from source s to sink t. Explicitly, f is given to us in the representations of integer flows along every edge e, $(f(e))$.

However, we find out that one of the capacity values of G was wrong: for edge (u, v), we used $c(u, v)$ whereas it should have been $c(u, v) - 1$. This is unfortunate because the flow f uses that particular edge at full capacity (i.e., $f(u, v) = c(u, v)$). We could rerun Ford-Fulkerson (or its improved versions) from scratch, but there shall be a faster way.

Design an algorithm to fix the max-flow for this network in $O(|V| + |E|)$ time. **Please give a 3-part solution.** ◀

[1] 这个证明太复杂, 引入了网络矩阵 (Network Matrices) 的概念, 我在网上找到了这个 notes 给出了详细证明 `https://webdocs.cs.ualberta.ca/~zacharyf/courses/combopt_2016/notes/lec16.pdf`

[2] 整数流定理: 如果网络流问题的流入、流出、边容量都是整数，目标函数也是线性函数，那么对应的线性规划所有的基本可行解都是整数解. 一般的, 约束矩阵是 TUM(Totally Unimodular Matrix) 的线性规划存在整数最优解.

Answer. 不妨设最大流为 f, 且 $f(u,v)=c(u,v)$, 否则 $c(u,v)$ 减小 1 不会改变最大流. 考虑如下的算法:

Algorithm 1 Restoring the Balance

```
Input: Graph $G=(V,E)$, edge capacities $c(e)$, max flow $f$
Output: Correct max flow $f_{max}$
1: $f' \leftarrow f$
2: Use BFS to find a path $P$ form $s$ to $t$ that contains edge $(u,v)$.
3: Decrease the flow along the path $P$ by 1, i.e., $f'(e) \leftarrow f'(e)-1, \forall e \in P$
4: Run Ford-Fulkerson on $G$ with $f'$ to find the max flow $f_{max}$
5: return $f_{max}$
```

(1) 时间复杂度分析: 第 2-3 行 BFS 的时间复杂度为 $O(|V|+|E|)$, 注意到第 3 行路径 P 上的所有流减少 1 后, $|f'|=|f|-1$, 而 $|f_{max}| \leq |f|$, 而 Ford-Fulkerson 算法每次循环流大小至少增加 1. 故第 4 行只需要循环 1 次 (即使用 BFS 找一次残差网络的增广路径) 即可找到最大流 f_{max}, 时间复杂度为 $O(|V|+|E|)$. 总的时间复杂度为 $O(|V|+|E|)$.

(2) 正确性证明: 在课上证明了任何非最大流的残量网络中都可以找到一条增广路径. 要么第 4 行中能找到增广路径, 此时 $|f_{max}|=|f'|+1$. 要么找不到增广路径, 此时 f' 即为最大流 f_{max}. ◁

Problem 4. (Feasible Routing).

In this problem, we explore a question called *feasible routing.* Given a directed graph G with edge capacities, there are a collection of supply nodes and a collection of demand nodes. The supply nodes want to ship out flow, while the demand nodes want to receive flow. The question is whether there exists a flow that satisfies all supply and demand.

Formally, we are given a capacitated directed graph $G=(V,E)$, and each node is associated with a *demand value*, d_v. We say that v is a supply node if it has a negative demand value (i.e., flow out > flow in), and a demand node, if it has a positive demand value (i.e., flow in > flow out). A node can be neither demand or supply node, where $d_v=0$. Let $c(u,v)$ be the capacity of the directed edge (u,v). Define a *feasible routing* as a flow that satisfies

- (Capacity Constraint) For each $(u,v) \in E$, we have $0 \leqslant f(u,v) \leqslant c(u,v)$
- (Supply and Demand Constraint) For each vertex $v \in V$, we have

$$\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) = d_v$$

Note that this is a feasibility problem, and the answer is simply yes or no, whereas the max-flow is an optimization problem, where the answer is a number (max flow value).

(a) Let S denote the supply nodes and T the demand nodes. Define the total demand as $\sum_{u \in T} d_u$ and total supply as $\sum_{u \in S} -d_u$. Is there a feasible routing if total demand is not equal to total supply?

(b) Provide a polynomial-time algorithm to determine whether there is a feasible routing, given the graph, edge capacities, and node demand values. **Please give a 3-part solution.** ◀

Answer. (a) 不存在可行流通使得 $\sum_{u\in T} d_u \neq \sum_{u\in S} -d_u$. 对任意的可行流通 f, 我们有:

$$\sum_{v\in V} d_v = \sum_{v\in V}\left(\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e)\right)$$

注意到对于 $\forall e=(u,v)\in E$, 在上式 in/out 求和中各出现了一次, 相互抵消, 因此上式等于 0, 即:

$$\sum_{v\in V} d_v = \sum_{u\in T} d_u + \sum_{u\in S} d_u = 0 \implies \sum_{u\in T} d_u = \sum_{u\in S} -d_u$$

证毕.

(b) 将这一问题转化为最大流问题. 考虑一个新图 $G'=(V',E')$, $V'=V\cup\{s^*,t^*\}$, 其中 s^*,t^* 为超级源点和超级汇点. 对 $\forall s\in S$, 向 E 中添加容量为 $-d_s$ 的边 (s^*,s), 对 $\forall t\in T$, 添加容量为 d_t 的边 (t,t^*), 这样得到了 E'. 考虑如下的算法:

Algorithm 2 Feasible Routing

```
Input: Graph $G=(V,E)$, edge capacities $c(u,v)$, node demand values $d_v$
Output: Does there exist a feasible routing?
Construct the new graph $G'=(V',E')$ as described above
Run Ford-Fulkerson algorithm on $G'$ with $(s^*,t^*)$ to find the max flow $f$
if $|f| = \sum_{u\in T} d_u$ then
    return True
else
    return False
```

时间复杂度为 $O(|V|\cdot|E|^2)$(也可以使用更快的解网络流的算法). 下面证明其正确性:

即证明: G 中存在一个带需求 $\{d_v\}$ 的可行流通 $\iff$ G' 的 (s^*,t^*) 最大流有值 $\sum_{u\in T} d_u$. 注意到

$$|f| = \sum_{u\in T} d_u = \sum_{u\in S} -d_u \iff \forall s\in S, f(s^*,s)=c(s^*,s)=-d_s \wedge$$
$$\forall t\in T, f(t,t^*)=c(t,t^*)=d_t$$

- $\Rightarrow$: 若 G 中存在一个可行流通, 那么令 G' 中 s^*,t^* 新添加的边的流量恰为容量, 容易验证容量限制和每个节点的平衡条件是成立的. 且 $|f|=\sum_{u\in T} d_u$, 显然 f 为 G' 的最大流.
- $\Leftarrow$: 若 G' 的最大流有值 $\sum_{u\in T} d_u$, 那么

  $$\forall s\in S, f(s^*,s)=c(s^*,s)=-d_s \ \wedge \forall t\in T, f(t,t^*)=c(t,t^*)=d_t$$

  此时令一个流通 $f'(e)=f(e), \forall e\in E$. 容易验证 f' 满足 G 的供需和容量限制, 因此 f' 为 G 的可行流通.

证毕. ◁