上下界网络流

一、问题引入

一般的，定义一个网络是一个加权的有向图 $G=(V,E,C)$，E 中的每条弧(u,v)都有一个容量上界 $C(u,v)\geqslant 0$。

如果人为的规定 V 中的两个点 s 和 t，其中 s 没有入度而 t 没有出度；并为 E 中的每条弧(u,v)赋予一个值 $f(u,v)\geqslant 0$，f 满足以下两个条件：

①除 s,t 之外的任意一个点 i 都满足：$\sum_{(u,i)\in E} f(u,i)=\sum_{(i,v)\in E} f(i,v)$；

②任意一条 E 中的弧(u,v)，都满足 $f(u,v)\leqslant C(u,v)$。

则称 f 是 G 的一个**可行流**，称 s 为流的源且 t 是流的汇。前一个条件被称为**流量平衡条件**，而后者则是**容量限制条件**。

而如果一个可行流 f 使原点提供的流量 $\sum_{(s,i)\in E} f(s,i)$ 达到最大，则称 f 是 G 网络的**最大流**。

如果为 G 中的每条边再加入一个容量下界：令 $G=(V,E,B,C)$，$B(u,v)$表示弧(u,v)的容量下界。这样 G 就是一个**容量有上下界的流网络**。类似的定义 G 中的可行流 f：

①除 s,t 之外的任意一个点 i 都满足：$\sum_{(u,i)\in E} f(u,i)=\sum_{(i,v)\in E} f(i,v)$；

②任意一条 E 中的弧(u,v)，都满足 $B(u,v)\leqslant f(u,v)\leqslant C(u,v)$。

同时可以定义 G 中的最大流 f_{max}，对容量有上下界的网络来说，还可以定义这个网络的**最小流** f_{min}：使原点提供流量 $\sum_{(s,i)\in E} f(s,i)$ 达到最小的流。

二、求解可行流

为了最终能够解决问题，不妨来看一个简化版的问题：

[问题 1.1]在一个有上下界的流网络 G 中，不设源和汇，但要求任意一个点 i 都满足流量平衡条件：

$$\sum_{(u,i)\in E} f(u,i)=\sum_{(i,v)\in E} f(i,v)$$

且每条边(u,v)都满足容量限制 $B(u,v)\leqslant f(u,v)\leqslant C(u,v)$的条件下，寻找一个可行流 f，或指出这样的可行流不存在。

不妨称这个问题为**无源汇的可行流**。

[问题 1.1 的解答]仔细分析一下，由于普通最大流中对每条边中流量的约束条件仅仅是 $f(u, v)\geq 0$，而在这个问题中，流量却必须大于等于某一个下界。因此可以想到，设

$$f(u, v) = B(u, v) + g(u, v) \quad (*)$$

其中 $g(u, v)\geq 0$，这样就可以保证 $f(u, v)\geq B(u, v)$；同时为了满足上界限制，有

$$g(u, v)\leq C(u, v) - B(u, v)$$

令 $C'(u, v) = C(u, v) - B(u, v)$，则大致可以将 $g(u, v)$看作无下界流网络 C'中的一个可行流。当然这样直接转化显然是不对的，因为这样仍无法体现“下界”这个条件。将(*)式代入流量平衡条件中，对于任意一个点，有：

$$\sum_{(u,i)\in E}[B(u,i)+g(u,i)]=\sum_{(i,v)\in E}[B(i,v)+g(i,v)]$$

$$\sum_{(i,v)\in E}g(i,v)-\sum_{(u,i)\in E}g(u,i)=\sum_{(u,i)\in E}B(u,i)-\sum_{(i,v)\in E}B(i,v)$$

如果设：

$$M(i)=\sum_{(u,i)\in E}B(u,i)-\sum_{(i,v)\in E}B(i,v)$$

即 $M(i)$为流入结点 i 的下界总和减去流出 i 的下界总和。

如果 $M(i)$非负，那么有：

$$\sum_{(i,v)\in E}g(i,v)=[\sum_{(u,i)\in E}g(u,i)]+M(i) \qquad (1)$$

设一附加源 S_0，则可以令

$$C'(S_0, i) = M(i)。$$

如果 $M(i)$是负数，那么有：

$$[\sum_{(i,v)\in E}g(i,v)]-M(i)=\sum_{(u,i)\in E}g(u,i) \qquad (2)$$

设一附加汇 T_0，令

$$C'(i, T_0) = -M(i)。$$

这里$-M(i)$是正数。

至此，附加图构造完毕。

在这样一个加入附加源和附加汇的流网络 C' 中，如果任意 $g(S_0, i)$或 $g(i, T_0)$都达到满载，那么 C' 中的这一个可行流 g 一定对应原网络 G 中的一个可行流 f；反之 G 中的任意一个可行流 f 都可以对应 C' 中的一个 $g(S_0, i)$或 $g(i, T_0)$都满载的流。

而让从附加源点流出的弧都满载的可行流，一定是一个从附加源到附加汇的最大流。因此，求原网络 G 中的一个可行流等价于求 C' 中 S_0 至 T_0 的最大流，并判断从源点流出的弧是否满载：如果满载，则[问题 1.1]有解，否则一定无解。

三、求解最大最小流

[问题 1.2]在一个容量有上下界的流网络 G 中，求源点 s 到汇点 t 的一个可行的最大流。

[问题 1.2 的解答]如果从 s 到 t 有一个流量为 a 的可行流 f，那么从 t 到 s 连一条弧(t, s)，其流量下界 $B(t, s) = a$，则这个图一定有一个无源汇的可行流：除了弧(t, s)的容量为 a 外，其余边的容量与 f 相同。

如果从 s 到 t 的最大流量为 a_{max}，那么从 t 到 s 连一条下界 $B(t, s) = a' > a_{max}$ 的弧(t, s)，则从在这个改造后的图中一定没有无源汇的可行流：否则将这个可行流中的弧(t, s)除去，就得到了原图中 s 到 t 的流量为 a' 的流，大于最大流量 a_{max}，产生矛盾。

如果给定一个参数 a，如何判断在 G 中从 s 到 t 是否有一个流量为 a 的可行流呢？综上所述，判断在 G 中是否有 a 的可行流和判断在改造后的图中是否有一个无源汇的可行流完全等价。因此，执行一次普通最大流算法，就可以完成这个任务了。

下面回到[问题 1.2]中来，我们不妨二分枚举这个参数 a，每次改造图后执行一次最大流来判断是否有 s 到 t 的流量为 a 的可行流。这样找到 a 能取到的最大值，也就是 G 图中的最大流 a_{max} 了。

[问题 1.3]在一个容量有上下界的流网络 G 中，求源点 s 到汇点 t 的一个可行的最小流。

[问题 1.3 的解答]与[问题 1.2]类似，只是在加入弧(t, s)后二分(t, s)的容量上界 $C(t, s)$即可。

上述算法的时间复杂度均为 O(单次网络流复杂度 * log 最大流量)。

实际上还存在更加简便的方法：

首先有最大最小流的必要条件是原图要有一可行流。因此我们先判断是否有可行流，若无可行流则无解；否则，我们就得到一个循环可行流。

事实上，可行流的循环流量就是流过(t, s)边的流量，即(t, s)边反向边的容量。设可行流

流量为 x.

求最大流时在残余网络中可能还有新的增广路，因此以原图的 s, t 为源汇，再跑一遍最大流，设最大流为 y 则 ans = y.（原来 (t,s) 这条边的流量 x 现在会被退回去）。

对于最小流，考虑可行流可能多流了 s 到 t 的部分流量，我们从 t 到 s 跑最大流退流回去，注意到直接退流可能会使流不满足下界，我们将第一次跑流时超级源点 S 和超级汇点 T 连出的边及其反向边都删掉，并且再删掉(t,s)这条边，这样保证流不会顺着这些边退回去导致不合法。设最大流为 y 则 ans = x - y.

还有一种最小流求法：照求无源汇可行流的方法，建附加源点 S 与汇点 T，求一遍 S->T 的最大流. 但是注意这一遍不加原汇点 t 到原源点 s 的这条边，即不使之改为无源汇的网络去求解. 求完后，再加上那条汇点 t 到源点 s 上限 INF 下限 0 的边. 因为这条边下限为 0，所以 S,T 无影响. 再直接求一遍 S->T 的最大流. 若 S 出去的边全满流，T->S 边上的流量即为答案原图最小流，否则若不全满流即无解。第一次做最大流是把那些能流掉的循环流都流满，加上(t,s)这条边后，第二次最大流流过(t,s)的流量就尽可能小了。

练习题

ZOJ 2314 ; POJ 2396 ; ZOJ 3229 ; ZOJ 3496 ; POJ 3801 ; HDU 3157

上述内容摘自：

1.一种简易的方法求解流量有上下界的网络中网络流问题，周源