一种简易的方法求解流量有上下界的网络中网络流问题

安徽 周源

研究命题

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 $\sum_{(s,i)\in E} f_{\min}(s,i)$ 达到最小的流。

在这篇文章中，我们的讨论就将围绕容量有上下界的网络展开，经过一些必要的讨论，最后将得到用于解决这一系列问题的一个较为简单的，有着较好的应用价值的算法。

正文

第一部分

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

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

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。

第二部分

[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。

在一般问题中，如果所有弧上的容量为整数，那么算法一定可以在有限次计算之后停止，而算法的时间复杂度约为[$\log_2$ 流量]次的最大流过程，采用朴素的预流推进算法，执行一次最大流算法约需要 $O(N^3)$的时间，而[$\log_2$ 流量]基本可以看作与[$\log_2 N$]看作同阶。因此这个算法的时间复杂度约为 $O(N^3 \log_2 N)$。

在实际应用中，这个算法是非常有价值的。但从纯理论上说，这并不是一个多项式算法，特别是当流量可能为小数的时候，算法可能会无止境的执行下去。一种解决方法是规定一个较小的误差范围，这样算法的时间复杂度为 $O(N^3 * [\log_2 \text{精度}])$：从理论上说，这也只能算一个近似算法☺。另一种解决方法则是直接将答案离散为整点再进行二分，这样最多有 $O(2^N)$ 个整点，因此算法的时间复杂度为 $O(N^4)$。当然这种方法效率不算很高，而且实现起来很麻烦，在实际竞赛中也并不需要仅仅为追求理论上的“快感”而做出这么大的牺牲。

[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。

此类问题还有一些简易的扩展，如求容量有上下界的流网络 G 中的最小费用最大流之类，其本质思想都是**“二分参变量”**，这里不再赘述。

附录

SGU 中两道流量有下界试题：

p194 Reactor Cooling 和 p176 Flow Construction；

以及两题的程序：p194.dpr 和 p176.dpr。