18.433 组合最优化

流对偶及其算法

Sept 25，30　　授课教师：Santosh Vempala

1 简介

每条边都规定了方向的图称为*有向图*。一条边允许通过的最大流量称为*容量*，从顶点i到j的边容量记为$c_{i,j}$。

给定一个有向图$G=(V,E)$，*流*是从一个发点（源）$s\in V$到一个收点（汇）$t\in V$的路的总和。*边不交路*的集合指从s到t的路的集合，且其中任意两条路不共边。

设S是包含s但不包含t的一个顶点集，$\bar{S}=V-S$是S在V中的补集。一个$(S,\bar{S})$割的大小指从S到$\bar{S}$的有向边的数目。

定理 1.*(Menger)* $G=(V,E)$有k条s到t的边不交路$\Longleftrightarrow k$是最小有向 s－t 割的大小。

证明.

$(\Rightarrow)$ 充分性是显然的。

$(\Leftarrow)$ 假设结论不成立。取一个最小的反例G，即G的最小割大小为k但不包含k条从s到t的边不交路。因为G是最小的，删去任何一条边得到的图的最小边割会更小。特别地，G不含s的入弧和t的出弧，因为这些弧不会出现在任意的 s－t 割中。我们可以把问题分成以下两种情况：

(i) $\exists e\in E$，既不与s也不与t相邻。由G的最小性可知，e包含在某个最小割$(S,\bar{S})$中。集合$\bar{S}$收缩成一个最小割大小至少为k的图G'，集合S收缩成一个最小割大小至少为k的图G''。因为G是最小反例，G'有从s到收缩了的顶点$\bar{S}$的k条不交路，同时G''有从收缩了的顶点S到t的k条不交路。这两个k条不交路的集合仅在$(S,\bar{S})$割内相交，故它们可以合并为G中从s到t的k条边不交路。这就得出了一个矛盾。

(ii)每条边都与s或t相邻。把中间顶点（除了s和t以外的所有顶点）分成下面两组：

1、 入度大于出度的所有顶点，和s。

2、 出度大于入度的所有顶点，和t。

由定义，从第一组到第二组的边集中一定包含k条边。显然很容易找到从s到t的k条边不交路。 □

2、最大流－最小割定理

$(S,\bar{S})$割中所有从S到$\bar{S}$的弧的容量之和称为$(S,\bar{S})$的*容量*，记为$c(S,\bar{S})$。令f为从s到

t的一个流，下面我们也用f来表示这个流的流值。

引理 2. 令$(S, \bar{S})$为图G中任意一个 s－t 割，则有$f \leq c(S, \bar{S})$。

证明. 我们已经知道$f(s, V) - f(V, s) = f$，且对于任意的顶点$x \in V - \{s, t\}$ 均有$f(x, V) - f(V, x) = 0$成立。由此可知，$f(S, V) - f(V, S) = f$. 另外V是S和$\bar{S}$的并，因此$f(S, \bar{S}) - f(\bar{S}, S) = f$. 从而，

$$f = f(S, \bar{S}) - f(\bar{S}, S) \leq f(S, \bar{S}) \leq c(S, \bar{S}) \qquad \square$$

在有限图中一定有一个最大可行流。找到这个最大值并找出这个流是许多图论和网络问题中一个非常重要的问题。给定图$G = (V, E)$，$s, t \in V$分别为源和汇。取一个从s到t的流f，如何判断它是否为最大流？再来看一下引理 2，可以看出最大流的值不超过最小割容量。所以判断f是否为最大流的一个方法是找最小割，并求其容量，然后比较两个值。一个更好一点的方法是找一条f的可增广路。在上面以s为源、t为汇的图中，f的一条*可增广路*是指路$\{u_0, u_1, \ldots, u_r\}$，这里：

1、$u_0 = s$.

2、如果(u_i, u_{i+1})是一条边，那么$f_{i,i+1} < c_{i,i+1}$。

3、如果(u_{i+1}, u_i)是一条边，那么$f_{i+1,i} > 0$.

可以看出，路中的每一个顶点（除了s和t以外）上网络流一定为 0。如果$u_r = t$,那上述的增广路称为*流增广路*或*f－增广路*，可以用来增大流值。

这就引导我们发现了一个寻找最大流的算法，它与我们前面用到的寻找最大匹配的算法非常相似。

算法 I

{

1) 找一条 f－增广路。
2) 增广初始流。
3) 重复上面两步。

}

这个算法留给我们几个问题。寻找可增广路的最好方法是什么？这个过程是否有限？不存在增广路时f是最大流吗？下面我们倒着来回答上面几个问题。

定理 3. f是最大流$\Longleftrightarrow$不存在流增广路。

证明.

$(\Rightarrow)$ 显然，存在流增广路时f不可能是最大流。

$(\Leftarrow)$ 作$A_f = \{u \in V :$存在一条从s到u的可增广路$\}$。注意到不存在流增广路，从而$t \notin A_f$。考虑$(A_f, \bar{A}_f)$割：取边(i, j)，其中$i \in A_f$，$j \in \bar{A}_f$，则必有$f_{i,j} = c_{i,j}$，

否则我们可以扩大（grow）A_f。同理，$f_{j,i} = 0$。也就是说通过这个割的流量是$\sum c_{i,j} - 0.$而我们的传送量不可能超过割集容量，所以当$t \notin A_f$时f是最大流。 □

定理 4. *（最大流－最小割定理）*最大流的流值等于最小割的容量。

证明. 设f为最大流值，那么流f没有流增广路。既然它没有流增广路，那么图中一定包含一个具有容量f的割A_f。又因为任何一个割的容量都不小于f，所以结论得证。 □

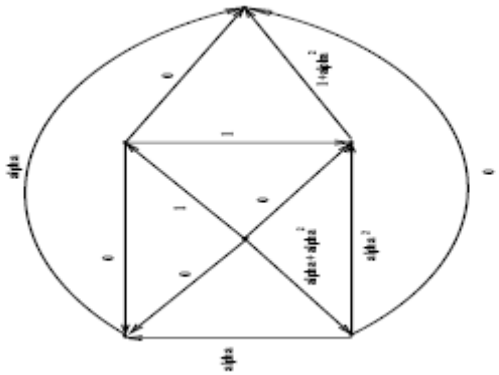

图 1：初始流

3 算法效率

有待回答的第二个问题是寻找流增广路问题的算法是否有限。答案是，不一定。看图 1，找到不恰当的增广路集合会导致无限次的增广，且仍然得不到最大流。图 1 中的边具有极大容量，且标出了流值为$1 + \alpha + \alpha^2$的初始流，其中α是$1 - \alpha - \alpha^3 = 0$的根。

为了找到图 1 的一条增广路，选择一条从s到t的路。仅考虑与路的方向相反的边，找到这条路上的当前最小流值（不用担心那些与路同向的边，因为这些边的容量极大），用这个值进行增广。

例如图 2，我们可以用α来增大这个流。接下来的情况是类似的，只不过找到的是容量为α^2的增广路。再接着我们沿具有容量 α^3的路来增加流。这个过程可以无限进行下去，另外，得到的流值的极限为$(1 + \alpha + \alpha^2 + \sum_{r \geq 1} \alpha^r)$，它是有界的，但是最大流可以是任意值。

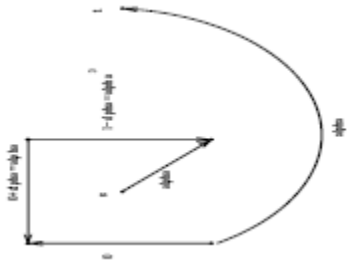

图 2：增加α

图 3：剩余图

4 一个弱多项式算法

有一个更好的方法来选择随机流并增加它，我们可以做一个剩余图。*剩余图*把原来的每条边变为两条不同的边，记为 Res(G)。例如容量为$c_{i,j}$流量为$f_{i,j}$的边(i,j)，在 Res(G)中变为具有方向相反的两条边：容量为$c_{i,j}-f_{i,j}$的边(i,j)和容量为$f_{i,j}$的边(j,i)，如果两条中有一条边的容量为 0，那么从 Res(G)中删掉这条边。如果图G的两个顶点间已经存在两条方向相反的边，那么 Res(G)的生成就变得较为复杂（见图 3）。

论断 1. G中的增广路对应于 Res(G)中的有向 s－t 路。
由此我们就得出了一个新的算法，但与我们前面用到的方法类似。
算法 II
{
1) 在 Res(G)中找一条 s－t 路。
2) 增大流。
3) 重复上面两步。
}
这个算法存在的问题，与前面一样，仍然依赖于初始增广路的选择。如何选择一条恰当的增广路呢？一个好的方法是选择有最大容量的路。这是一个有限的搜索过程，其证明留做作业。

论断 2. 一个流f至多可以被分解成m条从s到t的路，不包含圈。
论断的证明. 每次创建一条路，就从图中删去路上的限制边。所谓限制边，是指流量达到容量以致沿这条路不能再增加流量的边。很容易地得出最多有m条路。 □

定理 5. 存在一条路P，其容量满足$c(P) \geq \frac{f^*-f}{m}$，其中$f$是当前流值，$f^*$是最优流值。

证明. 给定图G，设f^*为其最大流，f为当前流。在$\mathrm{Res}(G)$中可以给当前流增加流量$f^* - f$而成为最大流，所以，在$\mathrm{Res}(G)$中存在一条路P容量满足$c(P) \geq \frac{f^*-f}{m}$。 □

接下来看一下这个算法的运行时间。找到一个最大容量路的复杂性为$O(m)$。再考虑增广集，每次至少增加流量$\frac{f^*-f}{2m}$，所以至多需增广$2m$次。经过这些增广后，剩余的流流值最大为$\frac{f^*-f}{2}$，因为当前最大容量路的容量小于$\frac{f^*-f}{2m}$。可以看出剩余的这一半至多需要增广$2m$次（至多需要增广$2m$次流量剩为一半 ），由此得出总的增广次数最多为$2m\log(nU)$

所以，上述算法执行时间为$O(m^2\log(nU))$，是一个弱多项式算法。

5 一个强多项式算法

另一个选择增广路的方法是找最短增广路。从发点开始，找出所有一步可达的顶点，即仅经过一条边，这些顶点称为第 1 层顶点。同样的方法可以找出第 2 层、第 3 层顶点… 找出到收点的最小层数（最短路），并尽可能的增加其流值。用$d(i)$表示结点i所在的层数，那么若边(i,j)在从s到t的最短路上，则必有$d(j) = d(i) + 1$。
下面的观察留做作业。
观察 算法进行的过程中，最短增广路的长度是非降的。
引理 6. 一条边是最小容量边的总次数为$O(n)$。

证明. 设容量为β的边(i,j)是增广路中的最小容量边。流值增加β，然后$\mathrm{Res}(G)$中就不再含边(i,j)。在这次增广之前，比如说在τ时刻，一定有$d(j) = d(i) + 1$，因为我们选择的是最短增广路。但现在，下一次这条边还要被用到，那边(i,j)必须仍然在$\mathrm{Res}(G)$中。要出现这种情况，我们一定是同时对某一条含边(j,i)的路进行了增广。在这个时刻，比如说在τ'时刻，有$d'(i) = d'(j) + 1$成立，因此$d'(i) \geq d(i) + 2$。而$d(i)$的最大值为n，从而定理得证。还要注意到在所有顶点上 d(i)总的增加小于n^2，因此增广的总次数为$O(n^2)$。 □

又因为找出一条最短增广路的执行时间为$O(m)$，所以整个算法的复杂性为$O(mn^2)$，是一个强多项式算法。