网络流

杨思祺
李琦煜
张远鹏

2020/04/03

目录

- 定义
  - 容量网络
  - 最大流
  - 最小割
- 性质
  - 引理7.1
  - 引理7.2
  - 引理7.3
- FF算法
- 辅助网络
- 一点点练习

定义

容量网络

容量网络 $N=<V,E,c,s,t>$: 有向连通图 $N=<V,E>$，记 $n=|V|,m=|E|$，每条边 $<i,j>$ 有非负容量 $c(i,j)$，有源点 s 和汇点 t。

定义

流

若 $f: E \to R^*$ 满足

- 容量限制（流量不超过容量）
  $\forall <i,j> \in E, f(i,j) \le c(i,j)$
- 平衡条件（流入恰等于流出）
  $\forall i \in V - \{s,t\}, \sum_{<j,i> \in E} f(j,i) = \sum_{<i,j> \in E} f(i,j)$

则称 f 是 N 上的一个可行流，流量 $v(f)$ 为源点的净流出量（汇点的净流入量）。
流量最大的可行流即为最大流。

定义

割

对于容量网络 $N =< V, E, c, s, t >$，如果 $A \subset V, \bar{A} = V - A, s \in A, t \in \bar{A}$，
则称 $(A, \bar{A}) = \{< i, j > \mid < i, j >\in E, i \in A, j \in \bar{A}\}$ 为 N 的割集，
$c(A, \bar{A}) = \sum_{<i,j>\in(A,\bar{A})} c(i, j)$ 为其容量。
容量最小的割集即为最小割集。

定义

最大流问题的线性规划形式

$\max v(f) \quad s.t.$

- $f(i,j) \le c(i,j) \quad \forall <i,j> \in E$
- $\sum_{<j,i>\in E} f(j,i) - \sum_{<i,j>\in E} f(i,j) = 0 \quad \forall i \in V - \{s,t\}$
- $v(f) - \sum_{<s,j>\in E} f(s,j) + \sum_{<j,s>\in E} f(j,s) = 0$
- $f(i,j) \ge 0 \quad \forall <i,j> \in E$
- $v(f) \ge 0$

性质

引理7.1

对任意一个可行流 f 和割集 $(A, \bar{A})$，其流量 $v(f) = \sum_{<i,j>\in(A,\bar{A})} f(i,j) - \sum_{<j,i>\in(\bar{A},A)} f(j,i)$。

性质

引理7.2

设容量网络 $N=<V,E,c,s,t>$， f 是 N 上任一可行流，$(A,\bar{A})$ 是任一割集，则 $v(f)\leq c(A,\bar{A})$。

性质

引理7.3

设容量网络 $N=<V,E,c,s,t>$，f 是 N 上可行流，$(A,\bar{A})$ 是割集，若 $v(f)=c(A,\bar{A})$，则 f 为最大流，$(A,\bar{A})$ 为最小割集。

Ford-Fulkerson 算法

定义

设 f 是容量网络 $N =< V, E, c, s, t >$ 上一个可行流，

- 饱和边：流量等于容量的边
- 非饱和边：小于容量的边
- 零流边：流量等于 0 的边
- 非零流边：流量大于 0 的边

Ford-Fulkerson 算法

定义

不考虑边的方向，从点 i 到点 j 的一条边不重复的路径称为 $i-j$ 链，

- 前向边：链中与链方向一致的边
- 后向边：链中与链方向相反的边

如果链中所有前向边都是非饱和边，所有后向边都是非零流边，则链为 $i-j$ 增广链。

Ford-Fulkerson 算法

定理

可行流 f 是最大流的充分必要条件是不存在关于 f 的 $s-t$ 增广链。

- $A:=\{j\in A|\text{存在关于}f\text{的}s-j\text{增广链}\}, s\in A, t\in\bar{A}$
- $\forall <i,j>\in(A,\bar{A}),\ <i,j>$ 饱和
- $\forall <j,i>\in(\bar{A},A),\ <j,i>$ 零流
- 由引理7.1 $v(f)=c(A,\bar{A})$
- 由引理7.3 f 是最大流，$(A,\bar{A})$ 是最小割集

最大流最小割集定理：最大流等于最小割。

Ford-Fulkerson 算法

基本思想

如果关于可行流 f 有 $s-t$ 增广链，令 δ 为所有前向边容量与流量之差及所有后向边流量中的最小值，有 $\delta > 0$，可以得到新的可行流 f'：

- $f'(i,j) = f(i,j) + \delta$ 如果 $< i, j >$ 是前向边
- $f'(j,i) = f(j,i) - \delta$ 如果 $< j, i >$ 是后向边
- $f'(i,j) = f(i,j)$ 其它（不在增广链上）

且 $v(f') = v(f) + \delta$。

Ford-Fulkerson 算法

基本思想

- 如果存在增广链，则修改可行流如上。
- 如果不存在增广链，则当前流即为最大流。

Ford-Fulkerson 算法

标号法寻找增广链

标号形式：$(\pm i, \delta_i)$

伪代码略（BFS记录路径最值及祖先结点）

Ford-Fulkerson 算法

运行时间和有限终止性

- 所有容量为正整数
    - 至多更新 C 轮
    - 每轮 $O(m)$ 完成
    - 总时间复杂度 $O(mC)$
    - 有限步内终止
- 所有容量为有精度限制的实数
    - 有限步内终止
- 所有容量为无理数（理论上无精度限制）
    - 不能在有限步内终止

辅助网络

定义

设 f 是容量网络 $N =< V, E, c, s, t >$ 上的可行流，定义关于 f 的辅助网络 $N(f) =< V, E(f), ac, s, t >$:

- $E^+(f) = \{< i, j > \mid < i, j >\in E, f(i, j) < c(i, j)\}$
- $E^-(f) = \{< j, i > \mid < i, j >\in E, f(i, j) > 0\}$
- $E(f) = E^+(f) \cup E^-(f)$
- $ac(i, j) = c(i, j) - f(i, j) \quad < i, j >\in E^+(f)$
- $ac(i, j) = f(j, i) \quad < i, j >\in E^-(f)$

辅助网络

引理7.4

设 N 的最大流量为 v^*，f 是可行流，则 $N(f)$ 的最大流量为 $v^* - v(f)$。

- $ac(A, \bar{A}) = \ldots = c(A, \bar{A}) - v(f) \quad \forall (A, \bar{A})$
- $N(f)$ 的最小割集容量为 N 的最小割集容量减 $v(f)$
- 由最大流最小割集定理， $N(f)$ 的最大流流量为 N 的最大流流量减 $v(f)$

辅助网络

定义

设 f 是 N 上一个可行流，g 是 $N(f)$ 上一个可行流，定义 $f' = f + g$ 如下

$\forall < i,j > \in E, f'(i,j) = f(i,j) + g(i,j) - g(j,i)$

其中 g 缺省值为 0。

辅助网络

引理7.5

设 f 是 N 上一个可行流，g 是 $N(f)$ 上一个可行流，则 $f' := f + g$ 也是 N 上可行流，且 $v(f') = v(f) + v(g)$。

辅助网络

分层辅助网络

对 $N(f)$ 进行BFS并按深度分层，去除冗余的边和冗余的点。

- $V(f) = \cup_{k=0}^{d} V_k(f)$
- $AE(f) = \cup_{k=0}^{d-1} < i, j > | < i, j > \in E(f) \wedge i \in V_k(f), j \in V_{k+1}(f)$
- $V_k(f) = \{i \in V | d(i) = k\} \quad 0 \leq k \leq d-1$
- $V_d(f) = \{t\}$

这是为 Dinic 算法做的铺垫

- 寻找最短的增广链
- 一次标号多次增广

一点点练习

带中间点容量的最大流问题

带中间点容量的容量网络 $N =< V, E, c, s, t >$，其中的 $c : E \cup (V - \{s, t\}) \to R^*$，即每条边 $< i, j >$ 有容量限制 $c(i, j)$，且每个中间点也有容量限制 $c(u)$，应当满足 $\forall u \in V - \{s, t\}, \sum_{<v,u>\in E} f(v, u) \leq c(u)$。请将带中间点容量的最大流问题转化为标准的最大流问题。

二分图最大匹配问题

对于给定的二分图 $G =< V, E >$，请尝试

- 建立模型用最大流表示其最大匹配
- 建立模型用线性规划表示其最大匹配