网络流

算法设计与分析小班课

2022 年 4 月

注意: 以下内容主要是给课本作一些注解并补充习题, 可能不能准确地代表课程内容和考核的方向, 也不能替代阅读课本、听课、完成作业题、做往年题等活动; 其中的内容没有经过课程主管老师的审核, 也可能存在错误. 不过出现的所有错误都由作者本人负责.

网络流是图论算法, 课本上介绍的基本模型有最大流、最小割以及最小费用流三种. 最直接的几个应用是匹配 (Hall 定理, 带权版本是课本匈牙利算法)、最小点覆盖 (König 定理, 魏斯桐回课讲到) 和不交路径问题 (Menger 定理, 往年题之防空避难), 这些可以回忆集合论与图论中的证明以及网络流算法的求解方法.

1 建模问题

建模的过程就是归约, 套路有

- 前面一段话给出的图论性质及其对偶;
- 建立超级源/汇, 通常是辅助平衡作用;
- 求最小割, 一般用于一些 “选择问题”, 例如王思远课上讲的求最大收益问题;
- 拆点, 可以用于一些和点有关 (比如点权, 或者对点有约束) 的问题;
- 二维选择, 也是一些 “选择问题”, 比如课上与图像分割有关的问题.

由于网络流适用的问题太杂, 变化太多, 所以到底还是要从题目本身入手.

其实, 既然网络流总是能写成线性规划, 所以万金油是: 图和流 $\approx$ 可行解中的 “关系” (匹配? 划分? 顺序?), 可行解对应的约束 $\approx$ 流量平衡, 提高解的性能 $\approx$ 找增广路. 这基本上能解决大多数问题. 应考时如果还是想不出归约方法的话就用线性规划的方法做 (但效率可能会降低).

问题 1 (扩展的网络流) 考虑如下输运问题. 在容量为非负整数的网络 (V, E, c) 中, 每个顶点还有一个需求函数 $d: V \longrightarrow \mathbb{Z}$, 要求流过 v 的净流量恰好是 $d(v)$. 即 $d(v) < 0$ 时, v 是一个发点, 否则 v 是一个收点.

(1) 设计一个算法确定是否有一个可行流, 满足以上发送和需求平衡的约束.

(2) 假设每条边还有流量下界 $\ell: E \longrightarrow \mathbb{Z}_{\geq 0}$, 要求流过边 e 的流量至少为 $\ell(e) \leq c(e)$, 重做上一题.

解 (1) 为了将其归约到原版最大流问题, 我们建立一个超级源 s 和一个超级汇 t. 源 s 连向那些发点, 边容量为 $|d(v)|$, 源 t 则连向那些收点, 容量也是 $|d(v)|$. 注意到如果原问题有可行解的话, 归约的问题对应的流量一定是 $\sum_{v\in V}|d(v)|$; 但归约问题的流量是不能超过 $\sum_{v\in V}|d(v)|$ (考虑 s 出发的扇形割), 因此问题就是问新问题的最大流量是否为 $\sum_{v\in V}|d(v)|$.

(2) 一个自然的想法是将 $\ell(e)$ 减去, 使得问题的限制重新变成不能低于 “零流”. 设想把每条边的流量都先设成 $\ell(e)$, 此时每个顶点的净流量为 $L(v) = \sum_{e_{\text{in}}}\ell(e_{\text{in}}) - \sum_{e_{\text{out}}}\ell(e_{\text{out}})$. 考虑新容量网络, 每个顶点的需求为 $d(v) - L(v)$, 边的容量为 $c(e) - \ell(e)$, 此时问题就变成了解决 (1). 不难看出两个问题的可行解存在一一对应的关系. □

问题 2 (拆点) 设有容量非负的网络 (V, E, c, s, t), 但容量是对顶点的流量限制 $c: V \longrightarrow \mathbb{R}_{\geq 0}$, 要求经过顶点 v 的流入流量不超过 $c(v)$; 边容量无限, 但流量平衡条件仍要得到满足. 设计算法求最大流.

解 每个顶点 v 拆成 $v_{\text{in}}, v_{\text{out}}$, 二者中间的边的容量为 $c(v)$. 在这个情况下最大流–最小割定理指出最大流等于使得 st 不连通的割点的最小容量和. □

2 网络流的补充例题

网络上有很多源于 OI 的题目, 这里就不收入了, 我们以下仍以求解 [KT06; Cor+09] 中的简单习题为主. 下面的部分题目模型略有重复, 可以适当跳过.

2.1 简单的二部图匹配

问题 3 赛克勒考古与艺术博物馆开办了有 n 件藏品的新展览. 展厅可视为二维平面上由平行于 x, y 轴的线段围成的闭合多边形, 这些线段一共有 m 个, 两个端点分别为 $(x_i, y_i), (x_i', y_i')\,(1 \leq i \leq m)$. 现在在这个多边形内部放好了 n 个藏品以及 n 个保安岗位. 假设每个保安只能管理一个藏品, 而且管理的藏品必须在保安的视线范围内——即保安和藏品的连线不能和多边形围墙有交点. 已知可以在 $O(1)$ 时间内询问两条线段是否交叉, 设计一个算法确定能否为 n 个保安位置安排对应负责的藏品, 使得每个藏品都在相应保安的视线范围内.

解 这是很简单的二部图匹配问题. 构造 n 个藏品顶点和 n 个保安顶点, 若二者的连线不和多边形围墙有交点, 则连边. 在所得二部图运行网络流算法确定是否有完美匹配即可. 需要的时间为 $O(\max\{n^2m, n^3\})$. □

问题 4 考虑如下移动网络搜索基站的问题. 二维平面上有基站 $b_1, \ldots, b_n$ 和 n 部手机 $p_1, \ldots, p_n$. 给定常数 $\Delta > 0$, 我们称移动网络是**连接良好的**, 如果每部手机都只和一个

Euclid 距离不超过 Δ 的基站连接. 现在假设 p_1 开始沿着一条直线移动 (其他手机均不动) z 个单位, 那么在这个过程中基站的连接就会发生变化. 设计一个 $O(n^3)$ 的算法确定整个过程中移动网络是否能一直保持连接良好, 如果是则给出一个调整方案.

解 首先注意到这是一个变化的二部图完美匹配问题. 变化的次数不会超过 $2n$ 次, 因为 p_1 运动的直线和全部 n 个半径为 Δ 的圆至多有 $2n$ 个交点, 即二部图不会变化超过 $2n$ 次. 暴力算法就是每次都计算一个完美匹配, 得到 $O(n^4)$ 的做法, 不过我们可以用注 23 的方法, 这样每次变化只需要搜一次增广路径, 时间为 $O(n^2)$, 这样算法就是 $O(n^3 + 2n \cdot n^2) = O(n^3)$ 的了. □

2.2 匹配 + 限制

问题 5 *沙漠中有 n 个考察站和 k 个通讯基站, 它们的位置都由 xy 平面上的二维坐标给出. 现在要将每个考察站连接到某个通讯基站. 由于通讯质量缘故, 考察站到连接的基站的 Euclid 距离不能超过 r, 而且每个通讯基站不能连接超过 L 个考察站. 设计一个算法确定是否能完成这个任务, 如能完成则给出一个方案, 否则回答不可能.*

解 在这里, 流量限制就是连接一个基站的考察站数目, 流量平衡条件允许我们构造超级源和超级汇来具体表达这个限制. 也就是说, 构造 n 个考察站结点和 k 个基站结点, 二者如果距离在 r 内, 就连一条容量为 1 的边. 所有的考察站都连到超级源 s, 边容量为 1; 所有的基站都连到超级汇 t, 边容量为 L. 那么只需要问该容量网络是否有流量 n 的流, 由于流量不可能超过 n, 所以求最大流即可 (注意这依赖于最大流一定是整数流). 需要的时间为 $O((n+k)^3)$ (Dinic, 以下不再分析时间复杂度). □

注 6 类似问题: 某地区出现洪水, 有 n 位伤者分别需要在半小时内送到 k 个医院中的一个, 是否存在一个送院方案使得抢救及时而且每个医院接受的患者数都不超过 $\left\lceil \frac{n}{k} \right\rceil$? ♤

问题 7 战时, 某医院某日分别得到了 $s_{\mathrm{A}}, s_{\mathrm{B}}, s_{\mathrm{AB}}$ 和 s_{O} 的供血 (下标代表血型), 此日需要的用血分别为 $d_{\mathrm{A}}, d_{\mathrm{B}}, d_{\mathrm{AB}}$ 和 d_{O}. 假设这些供需量都是整数. 设计一个算法确定该医院能否对血进行分配满足所有用血需求, 给出方案或者回答不可能. (需要用到生物学中的血型知识. 在平常, 医院是严格配型的, 一般不能把 O 型血配给 A 型血, 等等.)

解 其实可以用贪心法求解: 先满足所有 O 型血的受血者, 然后把剩下的 O 型血全部分配给 A、B 型血受血者; 余下没有得到分配的 A、B 型受血者用相应的准确配型满足, 剩下的血全部满足 AB 型受血者; 如果中间任何一步出现血不足, 就报告失败. (这个算法在能准确配型时可能并非准确配型, 如何修改能保持其正确性, 而且使得在能准确配型时完全准确配型?)

用网络流的方法则和上题做法一样, 从源到供血顶点的容量为相应的供应量, 从受血顶点到汇的容量为相应需求量, 中间的连接按血型匹配进行, 容量 ∞, 检查最大流中受血点到汇的边是否都饱和即可, 不过有点高射炮打蚊子的意思. □

问题 8 在接下来的 n 天, 某医院希望在第 i 天恰好有 p_i 位医生来工作 (不能多也不能少). 每位医生 $j\,(1 \le j \le k)$ 都提供了一个可工作的日子集合 $L_j \subseteq [n]$.

(1) 设计一个算法, 给每个医生安排一个工作日的集合 $L'_j \subseteq L_j$, 同时满足医院的要求, 或者报告不可能.

(2) 鉴于人类是天生懒惰的, 提交可工作日子的集合常常使得 (1) 的算法汇报不可能. 现在医院给出一个正整数 c. 请设计一个算法, 给每个医生安排一个工作日的集合 L'_j, 要求 $|L'_j \setminus L_j| \le c$ (即最多强迫医生 "加班" c 天), 同时满足医院的要求.

解 (1) 用前面两个题的方法就可以 (每天代表的顶点到汇点的边的容量有 p_i 限制), 但是实际上是平凡的 (先让每位医生都天天工作, 然后除掉多的医生). 对于 (2) 允许超出限制, 我们只要添加顶点来 "吸收" 这些限制就可以, 即对于每个代表医生的顶点 v_j, 添加一个顶点 u_j 和容量为 c 的边 (s, u_j). 对每个 j 不想工作的日子 i, 添加边 (u_j, w_i), 容量为 1. 再次问是否有流量为 $\sum_i p_i$ 的最大流即可. □

注 9 类似问题: 有 m 个气象气球和 n 个大气测量任务. 这些任务的集合为 S, 气象气球 $i\,(1 \le i \le m)$ 只能测量任务集 $S_i \subseteq S$. 每个任务至少需要有 k 个气象气球测量. 设计一个算法确定是否可能, 并在可能时给出一个方案. 进一步, 如果每个气象气球都有一个品牌属性, 一个任务不能由全部来自于一个品牌的气球测量, 重做前一个问题. (后一个问题用三层顶点: 气球、品牌和任务.) ♤

问题 10 一群卷王 $p_1, \ldots, p_k$ (全体记为集合 S) 住在一起, 在接下来的第 i 天中, 他们中的一部分 $S_i \subseteq S$ 要开一部车去学校 (假设车足够大). 鉴于他们都希望夜以继日地卷, 因此他们都不想驾驶, 而是让别人开车, 自己坐在车上写算法设计与分析的作业和大作业. 为了让他们获得公平地获得卷的机会, 我们设想每天均匀随机地让 S_i 中的某个人开车, 那么 $j \in [k]$ 的期望开车次数是 $\Delta_j = \sum_{i: p_j \in S_i} \frac{1}{|S_i|}$. 注意 Δ_j 不一定是整数. 现在希望给出一个 d 天的驾驶员安排 $p_{i_1}, \ldots, p_{i_d}$, 使得 $p_{i_t} \in S_t\,(1 \le t \le d)$ 而且 p_j 的开车次数不超过 $\lceil \Delta_j \rceil$. 证明: 这总是可能的.

证明 此题主要是温馨提示一下整数流问题. 用上面题目的办法建网络, 卷王 p_j 到每个 $p_j \in S_t$ 的 S_t 顶点都连接容量为 1 的边, 卷王顶点到汇点的容量为 $\lceil \Delta_j \rceil$, 每天对应的顶点 S_t 到源点的容量为 1. 因为给每个 S_t 相连的中间边 $1/|S_t|$ 的流量是一个可行流, 其流量为 $\sum_j \Delta_j$, 最大流的流量 $f \ge \sum_j \Delta_j$. 而**整数容量网络的最大流一定是整数流**, 所以一定有解. □

问题 11 考虑一个 $(n+1) \times (m+1)$ 的表格. 最后一行和最后一列分别为每行的和以及每列的和, (n, m) 元表示表格中所有数的和 (如下例子所示).

x_{11}	x_{12}	x_{13}	$= x_{11} + x_{12} + x_{13}$
x_{21}	x_{22}	x_{23}	$= x_{21} + x_{22} + x_{23}$
$= x_{11} + x_{21}$	$= x_{12} + x_{22}$	$= x_{13} + x_{23}$	$= 6$ 个数全部求和

假设每行的和以及每列的和都是整数, 但是其他数不保证是整数. 希望把每个不是整数的值 α 四舍五入到 $\lceil \alpha \rceil$ 或者 $\lfloor \alpha \rfloor$, 但是要求行和以及列和都不能改变. 证明: 一定能完成这件事. (提示: 只需解决 $\alpha - \lfloor \alpha \rfloor$ 的情形.)

解 本质上只需要解决小数部分是舍还是入, 故按提示取 $\alpha - \lfloor \alpha \rfloor$ 知可不妨设 $\alpha \in [0, 1]$. 对于每一行 i 和每一列 j 都给一个顶点. 流量约束就是求和的约束, 即 s 到行 i (resp. 列 j) 对应的顶点的流量是行和 (resp. 列和), 对于每个不为零的 (i, j) 元, 用容量为 1 的连接对应 i 行 j 列的顶点 (是否饱和代表该元素是否上入). 注意原表格就是一个可行流, 因此一定存在流量恰好是整个表格数的总和的最大流. □

问题 12 考虑如下推广的调度问题. n 项工作中, 每项工作 $i \in [n]$ 都有一个可以开始被处理的时间 a_i, 一个 ddl d_i 和需要花费的时间 ℓ_i (设 $0 < \ell_i \le d_i - a_i$). 有 k 台机器, 机器 j 可以在 $[t_j, t_j']$ 的时间内工作. 一台机器只能同时处理一台任务, 不过任务中途可以被暂停, 之后继续. 请设计算法确定是否能将这些工作和机器合理调度, 使得它们都能成功完成, 并在可能时给出一个方案.

解 这仍然是一个指派/匹配的问题, 就是每个任务在何时间段, 在何机器上加工. 但是这个匹配的状态会随着时间的变化而变化, 所以可以先把它们固定下来. 我们把区间 $[\min_i a_i, \max_i d_i]$ 用 $(a_i)_{1\le i\le n}, (d_i)_{1\le i\le n}, (t_j, t_j')_{1\le j\le k}$ 划分为一个个静态的时间段 $I_1, \dots, I_r$. 在这些时间段中, 有待处理的工作和可用的机器都是不变的, 而且那些上机器的工作的顺序也是无关紧要的, 只需要决定在机器上处理多少时间.

所以, 流量就是处理时间, 接下来就很简单了. 对每个工作 j 建立一个顶点 v_j, 以及时间段 I_ℓ $(1 \le \ell \le r)$ 也各给顶点. 若 $I_\ell \subseteq [a_j, d_j]$, 就给 v_j 和 I_ℓ 连接容量为 $|I_\ell|$ (可用时间) 的边; 源到每个 v_j 都有容量 ℓ_j 的边; 每个 I_ℓ 到汇都有容量为 $n_\ell|I_\ell|$ 的边, 这里 n_ℓ 是指 I_ℓ 中可用的机器数目. 最后计算是否有流量为 $\sum_i \ell_i$ 的最大流即可. 不过注意通过最大流导出方案的时候要防止两个机器同时处理一份工作——但因为在一个时间段内指派顺序和工作顺序是无关的, 只要总加工时间足够, 就总是可以重新调整得到可行解. □

2.3 推广的容量网络

问题 13 某网站的访客有不同的特征, 具有某个特征的访客的集合为 G_ℓ $(1 \le \ell \le k)$. 注意这些集合的交集可能不为空. 该网站和 m 位广告主签了合同, 每位广告主 i $(1 \le i \le m)$ 都有一个偏好 $X_i \subseteq \{G_1, \dots, G_k\}$ 和 r_i:

(i) 广告只能展示给 X_i 中的用户群体;

(ii) 每分钟广告至少要展示给 r_i 个属于 X_i 中某个用户群体的访客.

某分钟, 该网站有 n 位访客, 而且知道访客 $j \in [n]$ 属于的用户群体是 $U_j \subseteq \{G_1, \dots, G_k\}$. 请为网站管理员设计一个广告展示算法, 确定在这一分钟是否能满足所有广告主的要求, 并在能满足时给出一个方案. (每个用户只能看一个广告.)

解 只需要确定恰好达成 $\sum_i r_i$ 次的广告展示方案是否可能即可. 利用例题 1 的模型, 令 $v_1, \dots, v_n$ 为访客顶点, $w_1, \dots, w_m$ 为广告主顶点, 边 (v_j, w_i) 表示广告主 i 对用户 j 展示

广告感兴趣, 这些边容量为 1, 表示用户只能看一个广告. 对于广告主 j, 它想要播放广告的次数相当于下界, 因此 w_i 到超级汇点有边, 其容量下界为 r_i. v_j 到超级源点的边的容量为 1. 源点和汇点的需求分别为 $-\sum_i r_i$ 和 $\sum_i r_i$, 其他点的需求均为 0. 不难看出问题和该容量网络是否有可行流是完全等价的. □

注 14 类似问题: 近来某学校 WiFi 信号质量堪忧, 计算中心挑选了 n 位同学的 n 台笔记本电脑对 n 个信号发射点做测试. 这些发射点的集合记为 S, 并设笔记本电脑 $\ell\,(1 \le \ell \le n)$ 在信号发射点集合 S_ℓ 的范围内. 设每个 $S_\ell \neq \varnothing$ 而且每个发射点都至少有一台笔记本电脑在范围内. 计算中心人员希望进行 k 次测试, 每次测试让某台电脑 ℓ 连接某个发射点 $p \in S$ (要求 $p \in S_\ell$), 使得全部的测试能让每台笔记本电脑都进行过至少一次测试, 且每个发射点也被测试一次. 设计一个算法确定 k 次是否能完成任务, 可能时给出方案. (办法: 超级源和超级汇的需求分别为 $-k, k$.) ♤

问题 15 2202 年, 某大学的 ICS 课程引入了 office hour 制度. 在时间 $I_1, \dots, I_k$ 的一部分中, ICS 将举行 office hour 并将请助教负责答疑, 每次 office hour 只能有一位助教. 每位助教告诉教务他/她在 $I_1, \dots, I_k$ 这些时间段的空闲情况 (是否能来答疑). 为了促使助教们努力工作, 教务要求每个助教参加的 office hour 次数合计在 $[a, b]$ 之间, 而且举行的 office hour 次数恰好为 c 次. 另外, 由于在 lab 快要到 ddl 的时候同学们会比较需要答疑, 因此教务还把 $\{I_1, \dots, I_k\}$ 划分为一些时间段 $S_1, \dots, S_\ell$, 并用 $d(S_i)$ 表示时间段 S_i 至少需要的答疑次数. 请设计一个算法确定教务的要求能否被满足, 可能时给出方案.

解 还是利用例题 1 的模型. 我们为每位助教 $ta_1, \dots, ta_m$, 时间 $I_1, \dots, I_k$ 和要求有 office hour 的时间段 $S_1, \dots, S_\ell$ 各自建立顶点. 源点 s 到助教连边, 容量为 $[a, b]$, 助教到各自合适的时间连边, 容量为 1, 时间到对应的时间段 S_i 连边, 容量为 1. 每个时间段 S_i 到汇点连边, 容量为 $[d(S_i), c]$. 最后, 因为总次数为 c 次, 所以源和汇的需求分别为 $-c, c$. □

2.4 多个点/边的构造

问题 16 在二维平面上有 n 个已经指定坐标的路由器 $1, 2, \dots, n$, 现在它们要组成一个带备份的网络. 要求每个路由器至少以 k 个路由器作为备份 (比如自己即将失效的时候给这些备份发送消息), 而每个路由器被作为备份的次数不能超过 b.

(1) 若要求备份路由器必须在 d 米的范围内, 设计一个算法确定是否可能, 并在可能时给出一个方案.

(2) 更现实地, 假设信号的是随着距离的增加的衰减的, 距离路由器 x 米的信号会衰减到 $f(x)$ 的水平. 现在给出常数 $p_1 \ge \cdots \ge p_k$, 要求每个路由器的备份中, 距离第 $j\,(1 \le j \le k)$ 近的路由器的信号满足 $f(d_j) \ge p_j$. 重做上一问.

解 相比于例题 5, 这里路由器同时具有两个角色, 因此拆成两个点——备份和被作为备份是有益的. (1) 每个路由器 i 对应点 v_i, w_i. 源到 v_i 各有容量为 k 的边, w_i 到汇各有容量为 b 的边. 若 i, j 互在 d 米范围内, 则连接 $(v_i, w_j), (v_j, w_i)$. 询问该网络的最大流是否饱和全部 sv_i 边即可.

(2) 如法炮制, 这时要决定的是 i 的第 j 近的备份路由器. 我们首先预处理出可以作为 i 的第 j 近的备份路由器的集合 $X_{ij} = \{\ell : f(d_\ell) \ge p_j\} \setminus \{i\}$. 用 $(u_{ij})_{1\le i\le n, 1\le j\le k}$ 来表示选择, 源 s 到这些点均有容量 1 的边; $(v_i)_{1\le i\le n}$ 表示被备份的路由器, 这些点到汇 t 都有容量为 b 的边. 注意现在还不能直接连接 u_{ij} 和 X_{ij} 中对应的 v 顶点, 因为 $X_{i,j+1} \subseteq X_{ij}$, 所以我们要防止一个路由器被选择多次. 为此添加顶点 $(w_{ij})_{1\le i\le n, 1\le j\le k}$, 它们和对应的 u_{ij} 有容量为 1 的边, 而 w_{ij} 和 v_ℓ 有一条容量为 1 的边, 当且仅当 $\ell \in X_{ij}$. 这样再计算网络的最大流是否饱和全部 su_{ij} 边就可以. □

问题 17 航空公司有 k 架飞机, 每天有 m 个航班 $1, 2, \dots, m$ 需要服务. 一般而言, 不需要使用 m 架飞机来执行这些任务, 因为有一些航班是具有接续性质的. 比如若某航班早上八点从北京起飞, 十点到达上海, 而十二点从上海到香港又有航班, 则一架飞机就够了. 假设已经给出了航班可接续的对 $S \subseteq [m] \times [m]$, 即 $(i, j) \in S$ 表示飞完 i 后同一架飞机可以飞 j; 接续对对应的图没有环. 请设计一个算法确定是否能用 k 架飞机完成全部 m 个航班, 如果可行则给出一个方案. (任何一架飞机的初始/结束状态可以在任何一个机场.)

解 可以设想一架飞机的轨迹就像是从某一个航班开始, 按一定的边的轨迹到达最后一个航班, 所以我们可以给每个航班一对点 (u_i, v_i) 并连接一条流量必须为 1 的边 (上下界均为 1), 确保航班恰有一架飞机执行. 而对于可接续的 $(i, j) \in S$, 连接顶点 (v_i, u_j), 容量为 1. 最后, 源点 s 连向全部的 u_i, 边容量均为 1, 表示飞机的起始航班, 而所有的 v_i 连向汇点 t, 边容量均为 1, 表示飞机的结束航班. 因为全部的飞机都要安排 (算上无任务的), s, t 的需求分别是 $-k, k$, 注意飞机可能用不完, 所以 st 之间还需要连接容量为 k 的边. 很容易从最大流中构造出安排的方案 (注意最大流中, 除了 st 边之外, 所有边的流量都必须是 1, 因为无环, 按照流量往前找航班安排直到结束就可以). □

注 18 类似问题 1: 仓库里有 n 个长方体箱子, 箱子 i 的尺寸为 (i_1, i_2, i_3), 输入确保 $i_1, i_2, i_3 \in \left(\frac{1}{2}, 1\right)$. 现在希望把一些箱子装在另外的箱子里 (可以套娃), 使得装完之后可见的箱子数最少, 请设计算法. (注意尺寸限制, 所以不可能存在两个箱子同时不套娃地装在一个箱子里, 这样只需要预处理可以装下的关系, 需要不超过 $3! \times n^2$ 时间.) ♤

注 19 类似问题 2: 给定 $n, \ell \in \mathbb{Z}_{\ge 1}$ 以及子集 $S_1, \dots, S_k \subseteq [n]$, 设计一个算法求出满足如下条件的 $[n]$ 的 ℓ 个排列 $\sigma_1, \dots, \sigma_\ell$: 每个 S_i $(1 \le i \le k)$ 中元素经过适当排列之后是某个 σ_j $(1 \le j \le \ell)$ 的前缀. (可以设想飞机的轨迹 $\approx$ 一个排列, 可以接续的航班 $\approx S_i \subseteq S_j$.) ♤

2.5 转成最小割/负容量

问题 20 某公司有 n 份代码 $1, 2, \dots, n$ 有待升级. 升级代码 i 的收益是 b_i. 但是这些代码之间存在耦合, 如果 i, j 没有全被升级或者全未被升级, 则另外需要 $x_{ij} \ge 0$ 的维护费用.

(1) 若 $b_i \ge 0$, 但代码 1 因为实在太糟糕, 无法进行升级 (加这个条件是因为单有 $b_i \ge 0$, 最优解就是全部升级). 请设计一个升级方案, 使得收益减去费用最大.

(2) 若 b_i 可以为负数, 但代码 1 可以升级, 重做前一问.

解 (1) 选择问题可以建模为一个割两边的顶点. 我们把 v_1 作为不升级的选择, 并制造一个源点 s 作为升级的选择. 对每个顶点 v_i, 从 s 连一条容量为 b_i 的边; 剩下的点 v_i, v_j 两两连接 x_{ij} 容量的两条边 (一对反向边, $x_{ij} = 0$ 则不连边). 现在考虑 sv_1 割 $(X, \overline{X})$ 的容量, 注意 $s \in X, v_1 \in \overline{X}$, 它是

$$c(X, \overline{X}) = \sum_{i \in \overline{X}} b_i + \sum_{i \in X, j \in \overline{X}} p_{ij} = 常数 - \sum_{i \in X} b_i + \sum_{i \in X, j \in \overline{X}} p_{ij},$$

因此求最小割相当于最大化 $\sum_{i \in X} b_i - \sum_{i \in X, j \in \overline{X}} p_{ij}$, 这就解决了问题.

(2) 也完全类似, 不过因为有负的收益, 所以可以先平移一下, 取 $r > \max_i |b_i|$ 并令 $\tilde{b}_i = r + b_i$. 现在构造以 $s, t, v_1, \ldots, v_n$ 的容量网络, 边 $(s, v_i), (v_i, s)$ 的容量为 r (表示不升级), 边 $(t, v_i), (v_i, t)$ 的容量为 $\tilde{b}_i$ (表示升级), 而点 v_i, v_j 两两连接 x_{ij} 容量的两条边 (一对反向边, $x_{ij} = 0$ 则不连边). 现在考虑 st 割 $(X, \overline{X})$ 的容量, 它是

$$c(X, \overline{X}) = \sum_{i \in \overline{X}} r + \sum_{i \in X} \tilde{b}_i + \sum_{i \in X, j \in \overline{X}} p_{ij} = 常数 - \sum_{i \in X} r - \sum_{i \in \overline{X}} \tilde{b}_i + \sum_{i \in X, j \in \overline{X}} p_{ij},$$

因此求最小割相当于最大化 $\sum_{i \in X} r + \sum_{i \in \overline{X}} \tilde{b}_i - \sum_{i \in X, j \in \overline{X}} p_{ij} = 常数 + \sum_{i \in X} b_i - \sum_{i \in X, j \in \overline{X}} p_{ij}$, 这就解决了问题. □

注 21 一般地说, 最大流算法不能很简单地扩展到容量为负数的情况中; 此题满足一些条件才使其可以. 参看例题 27. ♤

3 理论性较强的问题

下面的问题通常讲来不会在期末考试中主要考察, 但有意思, 供参考.

问题 22 假设在一个无环的整数容量网络 (V, E, c, s, t) 上有一个最大流 f. 该流是整数流, 而且每条边的上的流量都是正数. 现在任取一条边 $e^* \in E$ 并将其容量减 1. 决定如何在 $O(|V| + |E|)$ 时间内给出新的最大流.

解 因为是整数流, 因此用很朴素的方法找增广路径就可以. 不妨设 e^* 是饱和边, 否则 f 就是新的最大流. 设 $e^* = (v, w)$, 现在找一条 sv 和 wt 路径使得二者上的流量都是正数 (根据平衡条件一定能找到), 因为容量网络无环, 所以这两条路径不会出现重合. 在 $svwt$ 上每条边的流量都减少 1. 对于新流, 我们利用 FF 算法的办法搜索一条增广路径, 如果能增广就把增广得到的新流作为最大流; 否则减少的流就是最大流——因为这是整数流, 若增广则增广为 1, 而新的最大流的流量不会超过 f 的流量. □

注 23 类似问题: 考虑如下从部分匹配到完美匹配的转换问题. 假设二部图 (X, Y) 中 $X = (x_1, \ldots, x_n), Y = (y_1, \ldots, y_n)$, 现在已经有一个部分的 “匹配” 使得某 $n - 2$ 对顶点得

到完美匹配, 但对于 x_i, x_j 和 y_d, y_ℓ 来说, x_i, x_j 都连到了 y_d, 但 y_ℓ 未被匹配. 请基于这个"匹配", 在 $O(n^2)$ 时间内计算出原图的一个完美匹配, 或者回答完美匹配不存在. (做法: 删掉 $\{x_j, y_k\}$, 然后在匹配的基础上找增广路径.) ♤

问题 24 考虑一个简单容量网络 (V, E, s, t), 假设其上的最大流的流量是 f. 设边集 $F \subseteq E$ 满足 $|F| = k$, 及 $(V, E \setminus F, s, t)$ 的最大流的流量为 f'. 求解 $\max_{F\subseteq E,|F|=k} f - f'$.

解 从要去掉边可想到最大流–最小割定理. 对于 f 对应的最小割, 我们把其中的 k 条边去掉, 那么根据弱对偶性新的最大流量不会超过 $f - k$. 另一方面, 任意去掉 k 条边后的最小割的大小不会小于 $f - k$. 实际上, 任取 $F \subseteq E$, 任何一个 $(V, E \setminus F)$ 中的割中至少有 $f - k$ 条边 (因为 f 是原网络的最小割). 所以 $\max_{F\subseteq E,|F|=k} f - f' = k$, 构造方法已经给出. □

问题 25 两位同学 s, t 在一个网络中通信 (这个网络每条边的容量都是 1). 一个聪明的黑客切断了网络中的 k 条光缆, 使得 s, t 不通了. 已知该黑客切断的是一个最小割, 现在 s 同学希望通过 `ping` 这个命令来确定哪些顶点已经不可达了. 例如 `ping`(t) 一定是不通的, 而 `ping`(s) 会报告一条从 s 出发返回 s 的路径. 请设计一个算法, 只调用 `ping` $O(k \log n)$ 次, 使得 s 同学能知道全部不可达的顶点.

解 首先运行最大流算法得到 st 之间的 k 条不相交的路径 $P_1, \dots, P_k$. 显然黑客肯定把每条路径中的一条边 (且恰好只有一条) 切断了, 于是在每条 P_i $(1 \le i \le k)$ 上二分可得一条边 (v_i, w_i), 其中 v_i 能 `ping` 通但是 w_i 不行. 总体需要 $O(k \log n)$ 时间. 删掉这些边之后进行广度优先搜索即得到了全部可达顶点. □

问题 26 设矩阵 $M = (m_{ij}) \in \{0, 1\}^{n\times n}$. 称 M 是**可交换的**, 如果能适当地交换 M 的行 (或列), 使得其对角元素均为 1. 请设计一个算法确定给定的 M 是否可交换.

解 构造二部图 $(\{x_1, \dots, x_n\}, \{y_1, \dots, y_n\})$ 并令 (x_i, y_j) 有边当且仅当 $m_{ij} = 1$. 只需要证明这个二部图有完美匹配当且仅当 M 可交换. 若完美匹配将 x_i 匹配到 $y_{\sigma(i)}$, 则 σ 是双射, 而 $m_{i,\sigma(i)} = 1$, 所以只需要按 σ 重排 M 的行就可以. 若 M 是可交换的, 假设行 i 换到 $\sigma(i)$ 而列 j 换到了 $\rho(j)$, 则 $(x_{\sigma^{-1}(i)}, y_{\rho^{-1}(i)})$ $(1 \le i \le n)$ 就是想要的完美匹配. □

问题 27 在标准的网络流问题中, 容量是非负实数. 现在我们取消这一条件, 并希望求出**最小割** (不管最大流).

(1) 解释为什么在这情况下标准的网络流方法不再可用.
(2) 假设只有和 s, t 相连的边的容量可能是负数, 证明可以用标准的网络流方法解决问题.

解 (1) 此时最大流–最小割定理不一定成立, 注意课本引理 7.1 到 7.2 的推导是依赖于流量非负的. 另外, 如果我们能多项式时间内求解带负容量的最小割, 那么就能在多项式时间内求解最大割——对于一个最大割问题, 将所有的容量取负即完成归约. 但是最大割的求解是 NP-难的.

(2) 无妨设每个除 s,t 的顶点 v 都拥有 sv 和 vt 边 (不然补上容量为 0 的边. 虽然我们允许流量为负, 但是在原网络中最小割和最大流的原有联系已经消失了). 如果 $c(s,v),c(v,t)$ 都是非负实数则取 $d(v)=0$, 否则取 $d(v)=\max\{|c(s,v)|,|c(v,t)|\}$, 并将 $d(v)$ 加到各自的容量上. 若原网络的某个割的容量为 $c(X,\overline{X})$, 因为 s,t 一定被分开, 所以不论 $v\in X$ 还是 $v\in\overline{X}$, 这个割在新网络上的容量是 $c(X,\overline{X})+\sum_{v\in V}d(v)$, 因此两边的最小割是一一对应的. □

4 附: 网络流与对偶性

最大流和最小割在线性规划中是互为对偶的问题, 以下用强对偶性给出最大流–最小割定理的证明.

设所说的容量网络为 $(N=(V,E),c,s,t)$, 顶点编号 $1,\dots,n=|N|$; 并记 x_{ij} 表示网络流中, ij 边上的流量, 相应的 c_{ij} 表示该边的容量. 最大流问题可以写为线性规划问题:

$$\begin{aligned}
\max_{\boldsymbol{x}}\quad & \sum_{j:(s,j)\in E(N)} x_{sj},\\
\text{s.t.}\quad & \sum_{i:(i,k)\in E(N)} x_{ik}-\sum_{j:(k,j)\in E(N)} x_{kj}=0\quad (\forall k\in[n]\setminus\{s,t\}),\\
& 0\le x_{ij}\le c_{ij}\quad (\forall (i,j)\in E(N)).
\end{aligned}$$

对该问题取对偶得到

$$\begin{aligned}
\min_{\boldsymbol{y}}\quad & \sum_{(i,j)\in E(N)} c_{ij}y_{ij},\\
\text{s.t.}\quad & u_j+y_{sj}\ge 1\quad (j\in[n]\setminus\{s,t\}),\\
& -u_i+u_j+y_{ij}\ge 0\quad (i,j\in[n]\setminus\{s,t\}),\\
& -u_i+y_{it}\ge 0\quad (i\in[n]\setminus\{s,t\}),\\
& y_{ij}\ge 0\quad (\forall (i,j)\in E(N)).
\end{aligned}$$

额外引入 $u_s=1,u_t=0$, 对偶问题进一步简写为

$$\begin{aligned}
\min_{\boldsymbol{y}}\quad & \sum_{(i,j)\in E(N)} c_{ij}y_{ij},\\
\text{s.t.}\quad & -u_i+u_j+y_{ij}\ge 0\quad (\forall (i,j)\in E(N)),\\
& u_s=1,u_t=0,\\
& y_{ij}\ge 0\quad (\forall (i,j)\in E(N)).
\end{aligned}\tag{4.1}$$

若把 y_{ij} 看成边的选择, 则这对应了一个割. 依据强对偶性只需要证明这个问题的最优解一定对应一个合法的割.

(以下证明最优解是整数解, 不感兴趣的同学可以不再往下看了.) 为此用一些小技巧来圈定 $\boldsymbol{u}$ 的取值范围. 上面关于 y_{ij} 的约束条件即 $y_{ij} \geq \max\{u_i - j_j, 0\}$, 因为 $c_{ij} \geq 0$, 故希望 y_{ij} 尽可能小, 所以其最优解和下面的拟线性问题的最优解一模一样:

$$\min_{\boldsymbol{u} \in \mathbb{R}^{|G|}, u_s = 1, u_t = 0} \sum_{(i,j) \in E(N)} c_{ij} \max\{u_i - u_j, 0\}.$$

进一步考虑某个变量 u_k, 和它关联的项有

$$\sum_{(i,k) \in E(N)} c_{ik} \max\{u_i - u_k, 0\} + \sum_{(k,j) \in E(N)} c_{kj} \max\{u_k - u_j, 0\}.$$

我们指出, 对于任意的 $(i,k) \in E(N)$, 都存在一个最优解, 使得 $u_i \geq u_k$. 事实上, 如果最优解中 $u_i < u_k$, 那么我们可以减小 u_k 至 u_i, 这过程中 $\max\{u_i - u_k, 0\}$ 恒为 0, 而 $\max\{u_i - u_j, 0\} \leq \max\{u_k - u_j, 0\}$ ($\forall (k,j) \in E(N)$), 因此最优性保持不变. 同理可证对于任意的 $(k,j) \in E(N)$, 都存在一个最优解, 使得 $u_j \leq u_k$. 有了这一有用的观察, 结合 $u_s = 1, u_t = 0$ 和归纳法不难得知, 存在一个最优解, 使得 $\mathbf{1} \geq \boldsymbol{u} \geq \mathbf{0}$.

总结一下, 式 (4.1) 所说的最大流问题的线性规划形式之对偶中, 一定有一个最优解满足 $\mathbf{1} \geq \boldsymbol{u} \geq \mathbf{0}$. 接下来我们再证明式 (4.1) 的一切最优解都一定是整数向量.

假设最优解的排列形如 $\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} \end{bmatrix}$, 把约束矩阵的列理解为对应于顶点 (u_i) 和边 (y_{ij}). 又, 除了 $u_s = 1, u_t = 0$ 外的每个约束对应于一条边, 故大部分行可以视为边, 则约束矩阵形如

$$\begin{bmatrix} & & M(G)^\top & & & I_{e(G)} \\ 0 & \cdots & \underbrace{1}_{\text{第 } s \text{ 个}} & \cdots & 0 & \mathbf{0}^\top \\ 0 & \underbrace{1}_{\text{第 } t \text{ 个}} & \cdots & \cdots & 0 & \mathbf{0}^\top \end{bmatrix}.$$

其中 $M(G)$ 表示有向图 G 的关联矩阵. 对于该约束矩阵, 我们还是用归纳法证明它是全幺模矩阵. 显然其 1 阶子矩阵的行列式都是 $\pm 1, 0$ 之一. 假设断言对小于 k 阶的子矩阵都成立, 对于 k 阶子矩阵, 如果它选定了 $I_{e(G)}$ 中的列, 或者选定了最后两行, 因为这些行列中至多有一个 1, 所以由归纳假设命题立刻成立. 否则, 行列只能在 $M(G)^\top$ 中取, 所以 $M(G)$ 是全幺模的, 在这些行列中选取子矩阵, 其行列式自然也是 $\pm 1, 0$ 之一.

从而, 式 (4.1) 一定有一个最优解, 其分量要么是 0 要么是 1. 今记 $S = \{i \in V(G) : u_i = 1\}, T = \{j \in V(G) : u_j = 0\}$, 那么 (S, T) 对应一个 s,t-割. 这目标函数刚好就是最小化这个割的容量. 最后依线性规划的强对偶性知定理证毕.

参考文献

网络流问题是 1962 年 Ford–Fulkerson 首先提出和研究的. 不过很有意思的是 [Sch02], 根据一份解密的美国空军文档, 当时最小割问题提出的动机和应用是破坏苏联的铁路网——如何以最小代价切断他们的运输网络? 除了课本上提到的求解网络流的几个算法之外, 预流推进这个更高效的算法也是很重要的, 这在 [Cor+09, §26.4 节] 有讲述. 网络流至今还是一个有意思的课题 [Che+22].

以上并未对图像分割的问题给出其他例子, 对此可以参考 [KT06, 第 7 章及其习题 35、36]. 另外, 最小费用流这个很重要的模型也没有涉及, 这需要参看 [Cor+09]. 最小割的唯一性是有意思的话题, 这在 [KT06, 第 7 章习题 45、46] 中有应用. 对于更多网络流的建模问题, 我们还可以在网络上找到许多来源于 OJ 的新奇问题.

[Che+22] L. Chen et al. “Maximum flow and minimum-cost flow in almost-linear time”. 刊于: *arXiv preprint arXiv:2203.00671* (2022).

[Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT press, 2009. ISBN: 978-0-26-203384-8.

[KT06] J. Kleinberg and É. Tardos. *Algorithm Design*. Pearson Education, 2006. ISBN: 978-0-32-129535-4.

[Sch02] A. Schrijver. “On the history of the transportation and maximum flow problems”. 刊于: *Mathematical Programming* 91.3 (2002), pp. 437–445.

编写: WC
E-mail: wchang@pku.edu.cn