网络流 20+4 题解题报告

water_mi

2019 年 3 月 15 日

目录

考点总结

以下是所涉及到的网络流题目的考点，可能对你有一些启发。

题目名称	考点
pilot	二分图最大匹配
space	最大权闭合子图
dist	只走一次拆点，最小路径覆盖
ball	同上，动态加边增广最大流
table	资源分配类最大流
seq	通过修改边权达成限制的最大流
test	资源分配类最大流
number	二分图最大权独立集
cloth	分类拆点，最小费用最大流
air	限制流量最大费用最大流
galaxy	按时间拆点，动态加边增广最大流
trape	有可经过多次和经过一次限制的流
trans	资源分配类费用流
alloc	二分图最小/大权匹配
balance	环形均分纸牌
robot	资源分配类费用流
range	覆盖类费用流
segment	覆盖类费用流
mars	资源分配类费用流
knight	二分图最大独立集

1 飞行员配对方案问题 pilot

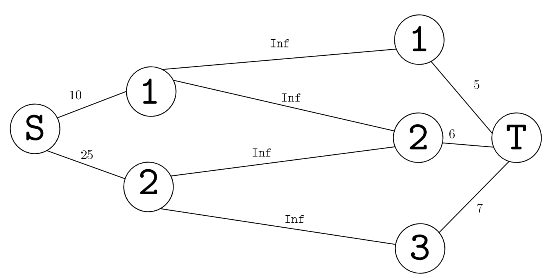
1.1 题解

考虑将二分图最大匹配转化为网络最大流模型，建立一个新的源点和汇点，源点向二分图中左部的点连流量为 1 的边，右部的点向汇点连流量为 1 的点，对于二分图中的一个匹配边，我们对应的从左部点向右部点连流量为 1 的边，跑最大流即可。

2 太空飞行计划问题 space

2.1 题解

考虑建立网络最小割模型，我们从源点向所有的实验连流量为这个实验费用的边，接着对于每一个仪器，我们连流量为仪器费用的边连向汇点，其中源点和汇点为新建点，接着对于每一个实验所需的仪器，我们从实验向仪器连流量为 `Inf` 的边，比如样例：



此时假如我们割掉了一条实验边，则代表我们不选择这个实验，如果割掉一条器材边，则我们购买了这个仪器，则此时最小割的意义就是不选择某些实验和选择某些实验所需的器材，同时保证失去费用的最少，最后答案就是所有实验的代价总和减去最小割。

接着考虑如何输出方案，考虑 *dinic* 在建立分层图时会发生什么，一个点会被分层当且仅当存在至少一条入边的残量不为 0 且它没被分层，对于一个实验，如果它的入边（此时只有一条）有残量，说明做这个实验是会增大净收益的，同时，因为 `Inf` 边的关系，它所需要的器材也会被分层，于是乎，在最后输出方案时，一个被分了层的实验/器材就是我们要选的。

3 最小路径覆盖问题 dist

3.1 题解

将一个点拆成两个点，即一个入点 x 和一个出点 y，因为每一个点都恰好且仅恰好在一条路径上，所以我们考虑建立新的源点向每一个入点连一条流量为 1 的边，每一个出点向新建的汇点连流量为 1 的边，对于一条 DAG 中的边 (u, v)，我们从 x_u 向 y_v 连边，最后计算出来的最大流与点数作差即为最小路径覆盖数。

考虑这样做为什么是对的，我们最开始有 n 个点，每多一条边 (u, v) 时，如果 u 的出度为 0 且 v 的入度也为 0，则我们显然将这两个点所在的路径连接在一起，如此做下去，如果某条边 (u, v') 不满足这个条件，我们考虑当前 u 所连接的点 v 能否找到一个新的入点 u' 与它相连，能找到则修改连接方式，这很显然就是二分图匹配的过程，我们每多一个匹配则少一个联通块，最终有多少个联通块就有多少条路径。

为什么要拆点变成二分图：对于左部 x 中的点，它的意义是一条边的起点，对于右部 y 中的点，它的意义是一条边的终点，一条边 (x_u, y_v) 就相当于我们有一种决策是可以将终点 v 和起点 u 连接起来，这样就可以模拟出上面所说的过程。

输出路径的话应该不难实现（记一下后继即可，不需要并查集）。

4 魔术球问题 ball

4.1 题解

类似于最小路径覆盖，我们只需要保证最小路径条数小于等于 n 即可，考虑一个点一个点的加入，每次连上权值加和为完全平方数的点，按顺序加点有一个好处，我们在动态加边的时候，我们只需在原有的残量网络上新加边即可，并不需要对其他之前加入的边修改，输出路径的话也同最小路径覆盖问题。

我比较懒，我在求得答案后（因为一个一个点的加需要减一）又重新建图跑了一次最大流找后继。

5 圆桌聚餐问题 table

5.1 题解

新建一个源点向每一个单位连流量为代表数的边，即保证这个单位有这些人；每一个圆桌向汇点连流量为容量的边，最后这条边残量为多少则代表剩下几个空位；对于每一个单位，向所有的圆桌连流量为 1 的边，表示限制不在同一桌聚餐，最后如果这张图的最大流等于人数就表示问题有解，输出方案的话就对于每一条单位到圆桌的边，如果残量为 0 则表示这个单位分配了一个人到这条边对应的圆桌。

6 最长递增子序列问题 seq

6.1 题解

这题应该叫最长不下降子序列才对

先动态规划处理出 $f[i]$ 表示以 i 为第一个元素的最长不下降子序列有多长，最后答案就是 $m = max(f[i])$

接下来解决如何用网络流来接决求方案数的问题，考虑通过拆点的方式来保证每一个元素只会出现在一个序列之中，对于点 i 拆成 i 和 $i+n$，由点 $i+n$ 来决定它之后选择哪一个元素，由点 i 来接受从某一个点 $j+n$ 或源点流过来的流量，显然需要建立一条由 $i \to i+n$ 的流量为 1 的边。

考虑如何建图，从源点向一个满足 $f[i] = m$ 的点 i 连流量为 1 的边，这样来保证所找出来的序列是最长的，然后考虑哪些点 j 可以当做点 i 的后继，不难发现当且仅当 $j > i, a[j] \geq a[i], f[j] + 1 = f[i]$ 时，j 为点 i 的后继，由 $i+n$ 向 j 连边，最后考虑统计答案，若 $f[i] = 1$，则由 $i+n$ 向 T 连边，表示这是一种可行的取出方案。

对于第三个问题, 我们只需修改从 $1 \to 1{+}n, n \to n{+}n, S \to 1, n{+}n \to T$ 这些边中存在的边为 Inf 后，再跑一次网络流即可，这样做就相当于我们限制它可以被多选，而且此时 1 只有可能是一个序列的开头，n 只有可能是一个序列的结尾，所以需要特判 $n = 1$ 的情况，否则会有无数的流量直接从 $S \to T$，此时问题三的答案即问题二的答案。

7 试题库问题 test

7.1 题解

与圆桌聚餐问题一样的套路，考虑对于每一个类型，我们连流量为需要选出的该类型题目个数的边到 T，对于每一个试题，我们从 S 连流量为 1 的边到它，然后连流量为 1 的边到可以选择的题目类型，此时我们就保证了这个试题只能属于一种类型，如果最大流等于 m，即满足要求。

输出选择方案的话同样是找出残量网络中的 0 流量边，不多解释。

8 方格取数问题 number

8.1 题解

显然这个 $n*m$ 的网格可以转化成一张二分图，我们所要求的就是这张图的最大权独立集，先对所有的点进行黑白染色，然后从源点向黑点连流量为权值的点，白点向汇点连流量为权值的的边，然后白点向相邻的黑点连流量为 Inf 的边，最后总权值减去最小割就是最大权独立集。

9 餐巾计划问题 cloth

9.1 题解

因为在一天中我们有两个决策：洗和买，所以考虑拆点，将第 i 天拆成点 i 与 $i+n$，前一个点表示这一天结束后我们洗的决策，后一个点表示我们今天能用多少餐巾，所以考虑如下连边方式：

1. 从源点开始：向点 i 连流量为 $r[i]$，费用为 0 的边，表示今天结束后，我们有 $r[i]$ 块餐巾没洗；向点 $i+n$ 连流量为 Inf，费用为 p 的边，表示买的决策。

2. 对于点 i：这个点是没洗的餐巾，从它开始向 $i+1$ 连流量为 Inf，费用为 0 的边，表示留存一些不洗的决策；向点 $i+n+M$ 连流量为 Inf，费用为 F 的边表示快洗决策；向点 $i+n+N$ 连流量为 Inf，费用为 S 的边，表示慢洗的决策。

最后从点 $i+n$ 向汇点 T 连流量为 $r[i]$，费用为 0 的边，输出最小费用即可

10 航空路线问题 air

10.1 题解

考虑拆点后跑最大流，将每一座城市 i 拆成两个点 $i, i+n$，然后从左到右连边，即对于一条航线 (u,v) 连边流量为 1 的边 $u+n \rightarrow v$。

与传统只经过一次每个点的最大流不同，我们需要找出两条题目所需的路径（不妨把来回各一条看作从最西段出发的两条路径），所以，我们在连以下边的时候要把流量改成 2，我们巧妙的转化成了最左端的城市和最右端的城市可以经过两次：

$$S \rightarrow 1, 1 \rightarrow 1+n, n \rightarrow n+n, n+n \rightarrow T$$

输出路径的话同样是找残量为 0 的边。

11 星际转移问题 galaxy

11.1 题解

直接做的话可能没什么好的思路，所以我们考虑转化问题的模型，假设有一个时间 T，我们只需要考虑能不能使最多到达的月球的人数大于等于 k，所以我们可以二分后跑一个最大流来判断最大流是否大于等于 k。

具体的建图方式：我们将每个位置（地球、月球、空间站）按照时间拆点，接下来连边如果没有特殊说明流量均为 Inf，源点对每个时刻的地球连边，每个时刻的月亮对汇点连边，每一个位置的上一个时刻向此时刻连边（代表留下来的决策），接着对于每一个太空船我们把它在上一时刻到达的空间站（上一时刻到达的空间站所拆出的上一时刻的点）向此时刻到达的空间站连流量为它可载人数的边，跑最大流即可。

但是如果二分的话，我们每次都需要将整张网络推翻重建，如果你做过魔术球问题的话你就知道存在一种效率更高的做法，我们可以在残量网络上动态加边（利用了最大流会调整残量网络的特性），所以我们可以不断增加 T 直到最大流大于等于 k。

12　数字梯形问题 trape

12.1　题解

由于我们需要用流来限制经过次数，所以我们考虑附加一个费用，即求最大费用

先考虑问题一，考虑拆点（一个点拆成 x, y），接下所有的边的流量都为 1（因为无论是边还是点都只能经过一次），$x[i][j] \to y[i][j]$ 费用为 0、$y[i][j] \to x[i+1][j]$ 费用为 $p[i+1][j]$、$y[i][j] \to x[i+1][j+1]$ 费用为 $p[i+1][j+1]$、S 向第一行的点（$x[1][j]$）费用为 $p[1][j]$，最后一行的点（$y[n][j]$）向 T 费用为 0。

因为费用只与答案有关，所以接下来的过程不会修改的费用（但是不用拆点了，任何不在第一行的点可以经过无数次）。问题二只需要在问题一的见图方式上修改最后一行的点到 T 的流量为 Inf（此时最后的边是可以相交的，实际上这条边并不存在）。问题三（边可以交）则修改除 S 到第一行的点以外的边的流量为 Inf，因为第一行每一个点只能发出一条路径。

13　运输问题 trans

13.1　题解

考虑费用流，S 向每一个仓库 i 连流量为 $a[i]$ 费用为 0 的边；每一个商店向 T 连流量为 $b[i]$ 的边，此时每一个仓库 i 向商店 j 连流量为 Inf，费用为 $C[i][j]$ 的边，跑一遍最小/大费用即可

14　分配问题 alloc

14.1　题解

显然是一个二分图最小/大权匹配，KM 或者费用流，这里介绍费用流建图：

人为二分图左部，工作为二分图右部，源点向左部点，右部点向汇点连流量为 1，费用为 0 的边，之后左部点 i 向右部点 j 连流量为 1 费用为 $C[i][j]$ 的边，这样子保证弄出来的是匹配。

15 负载平衡问题 balance

15.1 题解

这一题属于一种名为"环形均分纸牌"的类问题，可以利用中位数定理做到线性出解（详见一道叫做"糖果传递"的题），这里仅介绍网络流做法。

考虑费用流，设 ave 为整个数列的平均数，源点向每个仓库连流量为仓库库存，费用为 0 的边表示初始态，仓库向汇点连流量为 ave，费用为 0 的边，表示最终态，对于每一个点 i，向 $i-1$ 和 $i+1$ 连流量为 Inf，费用为 1 的边表示搬运的决策，求最小费用即可。

16 深海机器人问题 robot

16.1 题解

考虑费用流，对于原图中的一条边 (u,v,w)，我们从 u 到 v 连两条边，一条流量为 1，费用为 w 的和流量为 Inf，费用为 0 的边，一条供给采集决策，一条供给通过决策。之后对于每一个起点，源点向起点连流量为出发机器人个数，费用为 0，每一个终点向源点连流量为到达机器人个数，费用为 0 的边，求最大费用即可。

17 最长 k 可重区间集问题 range

17.1 题解

费用流，考虑对点离散化（最后有 m 个点），对于离散化后的每一个区间 $l[i],r[i]$，我们从 $l[i]$ 到 $r[i]$ 连流量为 1，费用为离散前的 $r[i]-l[i]$ 的边表示选择这个区间的决策，其他的话我们对于每一个点 i，向 $i+1$ 连流量为 k（或 Inf），费用为 0 的边，考虑 k 的限制，我们只需从源点向最左侧的点连流量为 k，费用为 0 的边即可，跑一趟最大费用流即可。

18 最长 k 可重线段集问题 segment

18.1 题解

同上一题，唯一的区别在于可能存在垂直于 x 轴的线段，此时我们建图会导致出现自环，所以考虑拆点，将点 i 拆成入点 i 和出点 i'，对于一个区间 (x, y)（假设你已经转化成了上面的问题），若 $x = y$，则从 x 向 y' 连边，否则从 x' 向 y 连边。

具体来说，我的转化方法是：先读入所有的区间，然后把端点 x,y 扩大一倍，如果 $x = y$ 则 $y + 1$，否则 $x + 1$，然后再把此时的 x,y 离散化，就变成了最长 k 可重区间集问题了。

19 火星探险问题 mars

19.1 题解

有些类似于深海机器人问题，但边的决策转化成了点的，所以我们考虑拆点，在点间连边，如果它不是一个障碍，则连流量为 Inf，费用为 0 的边，此外，如果它还是一个石料，则连流量为 1，费用也为 1 的边（限采集一次）。

由于此时建图方式与大部分需要输出方案的题目的建图不同，但我们还是根据残量网络来判断，对于一条边，如果它的反向边有流量，则代表这条边被经过了，通过它之后把反向边流量减去一，这样的话我们就可以很好地输出方案了。

20 骑士共存问题 knight

20.1 题解

很显然是一道二分图上的最大独立集，先进行黑白染色，然后根据最大独立集等于总点数减去最小点覆盖等于总点数减去最大匹配数，即在网络流意义上就是可选的点数减去最小割，连边之后（可以参考飞行员配对方案）求一个最小割即可，最大独立集就是可选点数减去最小割。

剩下的四道题的主流算法不是网络流或者网络流做法为假，这里介绍主流做法：

21 未完待续