网络流的一些建模方法

东营市胜利第一中学 姜志豪

摘 要

网络流在信息学竞赛中有着广泛的应用，很多网络流模型的建立方法十分巧妙。本文对一些比较常用的网络流建模方法进行了总结。

1 引言

网络流(network-flows)是一种类比水流的解决问题方法，在信息学竞赛中应用广泛。常见的网络流问题有最大流、最小费用最大流、有上下界的网络流等。网络流问题的巧妙之处往往不在于算法实现过程，而在于网络流的建模方法。本文对一些比较常用的网络流建模方法进行了总结，分为最大流建模、最小割建模、费用流建模和流量平衡思想四部分。希望能够给参加信息学竞赛的同学带来或多或少的帮助。

2 从最大流角度建模

一般来说，从最大流角度进行建模最直观。往往是用一条源点 $S \to$ 汇点 T 的流来表示一种方案。例如用最大流求二分图的最大匹配时，一条 $S \to T$ 的流就表示一个匹配。

2.1 建模举例

例 1. *(士兵占领)*

有一个 $n \times m$ 的棋盘，有的格子是障碍。现在你要选择一些格子来放置一些士兵，一个格子里最多可以放置一个士兵，障碍格里不能放置士兵。我们称这

些士兵占领了整个棋盘，当满足第i行至少放置了 r_i 个士兵, 第j列至少放置了 c_j 个士兵。现在你的任务是使用最少个数的士兵来占领整个棋盘。

$1 \le n, m \le 100$

最坏情况是先在每一行安排上指定数量的士兵，再在每一列安排上指定数量的士兵。这个方案中，每个士兵的贡献是1。

有些士兵可以既对行有贡献，又对列有贡献，贡献是2。这类士兵越多，需要的士兵总数就越少。考虑使这一类士兵尽可能多。第 i 行这类士兵的数量不能超过 r_i ，否则就有士兵对这一行没有贡献。同样，第 i 列这类士兵的数量不能超过 c_i 。考虑建图：

每一行建立一个点 A_i ，与源点 S 相连，容量是 r_i 。每一列建立一个点 B_i ，与汇点 T 相连，容量是 c_i 。

若第 i 行第 j 列可以放置士兵，就从 A_i 向 B_j 连一条容量为1的边。

这样建图后，一个贡献是2的士兵对应着一条 $S \rightarrow T$ 的流。边的容量限制了一个格子最多放置一个士兵，并且每行每列贡献是2的士兵的数量不超出要求。

求最大流，也就是贡献是2的士兵的最大数量，从而求出最小总士兵数。

例 2. *(Dining)*

有 f 种食物和 d 种饮料，每种食物或饮料只能供一头牛享用，且每头牛只享用一种食物和一种饮料。现在有 n 头牛，每头牛都有自己喜欢的食物种类列表和饮料种类列表，问最多能使多少头牛同时享用到自己喜欢的食物和饮料。

$1 \le n, f, d \le 100$

考虑用一条 $S \rightarrow T$ 的流来表示满足一头牛的要求，可以得到建图方式：

每种食物建立一个点 A_i ，与 S 相连，容量是1。每种饮料建立一个点 B_i ，与 T 相连，容量是1。

每头牛建立两个点 C_i 、 D_i ， C_i 与 D_i 之间连一条容量为1的边。

若第 i 头牛喜欢第 j 种食物，就将 C_i 与 A_j 相连，容量是1。

若第 i 头牛喜欢第 j 种饮料，就将 D_i 与 B_j 相连，容量是1。

用两个点来表示一头牛，中间连容量为1的边，是为了限制一头牛只被满足一次。

有一些题目的建模需要对问题进行分析，将问题简化或转化成可以用网络流解决的问题。我们来看下面这个问题。

例 3. *(Collector's Problem)*

*Bob*和他的朋友从糖果包装里收集贴纸。*Bob*和他的朋友总共 n 人。共有 m 种不同的贴纸。

每人手里都有一些（可能有重复的）贴纸，并且只跟别人交换他所没有的贴纸。贴纸总是一对一交换。

*Bob*比这些朋友更聪明，因为他意识到只跟别人交换自己没有的贴纸并不总是最优的。在某些情况下，换来一张重复贴纸更划算。

假设*Bob*的朋友只和*Bob*交换（他们之间不交换），并且这些朋友只会出让手里的重复贴纸来交换他们没有的不同贴纸。你的任务是帮助*Bob*算出他最终可以得到的不同贴纸的最大数量。

$2 \le n \le 10, 5 \le m \le 25$

Bob的朋友只会出让手里的重复贴纸来交换他们没有的不同贴纸。所以，对于Bob的某个朋友Friend，Bob只能把一种Friend没有的贴纸给他，并且一种最多给一次。Friend只会把他手里重复的贴纸给Bob，如果Friend有 i $(i \ge 2)$ 张某种贴纸，他至多给Bob $(i-1)$ 张。

那么，Bob的朋友的作用是，将Bob手中的贴纸 X 变成另一种贴纸 Y。

可以进行如下建图：

对每种贴纸 i 建立点 A_i。源点 S 向 A_i 连边，容量为Bob拥有的贴纸 i 的数量。A_i 向汇点 T 连边，容量为1。

对Bob的每个朋友 j 建立点 B_j。若朋友 j 没有贴纸 i，就从 A_i 向 B_j 连边，容量为1。若朋友 j 有 k $(k \ge 2)$ 张贴纸 i，就从 B_j 向 A_i 连边，容量为 $k-1$。

我们用 A 表示点 A_i 的集合，用 B 表示点 B_i 的集合。

一条 $S \to T$ 的流，是先从 S 到 A，表示Bob最初拥有的某种贴纸。然后经过若干次（可能是0次）到 B 再到 A 的过程，表示的是和朋友进行了交换。最后从 A 到 T，表示交换结束后Bob手中的贴纸的种类。

2.2 小结

最大流构图的特点是直观容易理解。$S \to T$ 的流，有着实际的意义，表示方案或操作方式。

不过，最大流问题的变化也非常多。有些时候，需要认真分析问题，发现问题的实质，将问题简化或转化，才能够得出网络流模型。

3 从最小割角度建模

3.1 用容量为正无穷的边表示冲突

例 4. *(NOI2006, 最大获利)*

新的技术正冲击着手机通讯市场，对于各大运营商来说，这既是机遇，更是挑战。*THU*集团旗下的*CS&T*通讯公司在新一代通讯技术血战的前夜，需要做太多的准备工作，仅就站址选择一项，就需要完成前期市场研究、站址勘测、最优化等项目。在前期市场调查和站址勘测之后，公司得到了一共 n 个可以作为通讯信号中转站的地址，而由于这些地址的地理位置差异，在不同的地方建造通讯中转站需要投入的成本也是不一样的，所幸在前期调查之后这些都是已知数据：建立第 i 个通讯中转站需要的成本为 p_i。另外公司调查得出了所有期望中的用户群，一共 m 个。关于第 i 个用户群的信息概括为 a_i , b_i 和 c_i：这些用户会使用中转站 a_i 和中转站 b_i 进行通讯，公司可以获益 c_i。*THU*集团的*CS&T*公司可以有选择的建立一些中转站（投入成本），为一些用户提供服务并获得收益（获益之和）。那么如何选择最终建立的中转站才能让公司的净获利最大呢？（净获利 = 获益之和 − 投入成本之和）

$1 \le n \le 5000, 1 \le m \le 50000$

将中转站、用户群看成点。建立点 A_i 表示中转站 i，从源点 S 向 A_i 连接容量为 p_i 的边，割这条边表示建立中转站 i，需要 p_i 的费用。建立点 B_i，表示第 i 个用户群，从 B_i 向汇点 T 连接容量为 c_i 的边，割这条边表示不满足这个用户群的要求，损失了 c_i 的收益。

若第 i 个用户群会使用中转站 j，那么 S 向 A_j 连接的边不能与 B_i 向 T 连接的边同时保留，所以可以从 A_j 向 B_i 连接一条容量为正无穷的边，这样就限制了那两条边不会同时保留。

所有用户群的获益之和减去最小割就是最大净获利。

在这道题目中，借助容量为正无穷的边，使冲突的收益无法同时保留。容量为正无穷的边还可以限制其他的冲突，我们看一下下面这道题目。

例 5. *(Codechef Dec14, Course Selection)*

铃现在正在大学学习。

课业计划共包含 n 项课程，每项课程都需要在 m 个学期里的某一个完成。

一些课程有前置课程， a 是 b 的前置课程表示课程 a 的完成的学期要在课程 b 完成的学期的前面。共有 k 组课程顺序要求。

对于课程 i ，在不同学期完成的得分不同。令 $x_{i,j}$ 表示课程 i 在学期 j 完成的得分。若 $x_{i,j}$ 为 -1 ，表示学期 j 中没有开设课程 i 。

计算铃各个课程分数的平均值的最大值。

$1 \le n, m \le 100, 0 \le k \le 100, -1 \le x_{i,j} \le 100$

各课程分数的平均值，就是各课程分数之和除以课程数。所以就是要最大化课程分数之和。

得分最大就是扣分最小，可以假设满分是100分，最小化扣分之和。用 $y_{i,j}$ 表示课程 i 在学期 j 完成的扣分。 $x_{i,j} = -1$ 时， $y_{i,j}$ 的值是正无穷，否则 $y_{i,j} = 100 - x_{i,j}$ 。

可以进行如下建图：

对于课程 i 、学期 j ，建立点 (i, j) 。

从源点 S 连向 $(i, 1)$ 一条边，容量为 $y_{i,1}$ 。

对于 $2 \le j \le m$ ，从 $(i, j-1)$ 连向 (i, j) 一条边，容量为 $y_{i,j}$ 。

从 (i, m) 连向汇点 T 一条边，容量为正无穷。

割去连入 (i, j) 的边，表示课程 i 在学期 j 学习。

这样求出的最小割，就是不考虑前置课程要求的情况下的最小扣分。

若 a 是 b 的前置课程，那么 a 的割边位置要在 b 的割边位置的前面。

从 S 向 $(b, 1)$ 连一条容量为正无穷的边。

对于 $2 \le j \le m$ ，从 $(a, j-1)$ 向 (b, j) 连一条容量为正无穷的边。

这样建图，就能使方案满足所有前置课程的要求。

容量为正无穷的边不会出现在最小割中。所以可以借助容量为正无穷的边，限制“ S 与边的起点连通”、“边的终点与 T 连通”这两个条件不会同时满足。从而使问题中的一些冲突情况在网络流图中表现出来。

3.2 从两点关系的角度进行最小割建模

例 6. *(happiness)*

高一一班的座位表是个 $n \times m$ 的矩阵，经过一个学期的相处，每个同学和前后左右相邻的同学互相成为了好朋友。

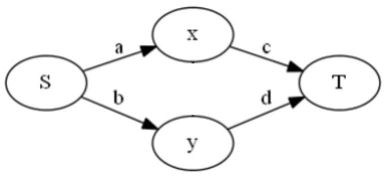
这学期要分文理科了，每个同学选择理科的喜悦值是 p_i，选择文科的喜悦值是 q_i。

一对好朋友如果同时选理科，他们又将共同获得喜悦值 v_{i1}。一对好朋友如果同时选文科，他们又将共同获得喜悦值 v_{i2}。

如何分配可以使得全班的喜悦值总和最大。

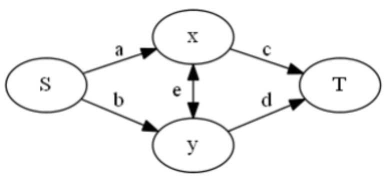
$1 \le n, m \le 100$

对每个同学建立一个点 x，点 x 向源点 S 连一条边，向汇点 T 连一条边，分别表示选择文科或理科。如下图所示。



每个同学选择文理科的收益可以最后再加进来，所以暂且不考虑。

若两个同学 x、y 相邻，并且他们没有选择相同的科类，就会少获得收益，或者认为是有损失。我们可以在 x、y 之间连边，来表示这种损失。



如上图，我们假设保留与 S 相连的边表示理科，保留与 T 相连的边表示文科。用 v_1 表示 x、y 同时选择理科的收益，用 v_2 表示 x、y 同时选择文科的收

益。那么，x、y的选择共有四种情况，情况及损失如下。

$$a + b = v_1 \quad (x\text{、}y\text{都选择文科}) \tag{1}$$
$$c + d = v_2 \quad (x\text{、}y\text{都选择理科}) \tag{2}$$
$$a + d + e = v_1 + v_2 \quad (x\text{选择文科，}y\text{选择理科}) \tag{3}$$
$$b + c + e = v_1 + v_2 \quad (x\text{选择理科，}y\text{选择文科}) \tag{4}$$

$$(3) + (4) - (1) - (2) = 2 \cdot e = v_1 + v_2$$
$$\therefore e = \frac{v_1 + v_2}{2}$$
$$a = b = \frac{v_1}{2}$$
$$c = d = \frac{v_2}{2}$$

解出边的容量后，再考虑上每个同学选择文理科的收益，可以这样来建图：

对于每个同学，建立一个点。从S向这个点连一条边，容量是这个同学选理科的收益，再加上他与所有相邻同学都选理科的共同收益和的一半。从这个点向T连一条边，容量是这个同学选文科的收益，再加上他与所有相邻同学都选文科的共同收益和的一半。

对于相邻两个同学x、y，在x、y之间连一条双向边，容量是他们都选文科的共同收益与他们都选理科的共同收益的平均数。

所有收益的和减去最小割就是最大喜悦值之和。

这个问题中，收益至多涉及两个人。可以先考虑两个人之间的关系，然后把所有关系综合起来，进行建图。

例 7. *(Google Code Jam 2008 Final E, The Year of Code Jam)*

*Sphinny*正在看新一年的日程表。*Sphinny*生活的世界与我们的世界有所不同，那个世界里一年有n个月，每个月恰有m天。

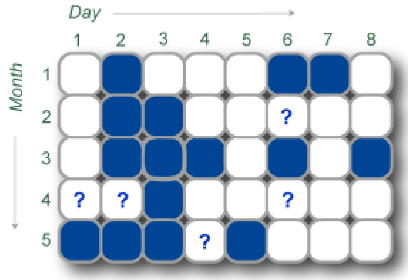
她将这一年的每一天都用以下三种方式之一在日程表上打标记：

*1.*白色：这一天她将不参加竞赛。

*2.*蓝色：这一天她将参加一场竞赛。

3.问号：这一天有预定好的竞赛，但她还没有决定好是否参加。

下面的图片是一张5个月，每个月有8天的日程表。



Sphinny想最大化所有竞赛的喜悦值之和。一场竞赛的喜悦值的计算方式是：

1.初始为4。

2.每有相邻（有公共边）的一天也要参赛，喜悦值减1。

Sphinny决定把每一个问号标记都改为白色标记或蓝色标记，并最大化总喜悦值。帮助Sphinny求出总喜悦值最大是多少。

$1 \le n, m \le 50$

这个题目和上一个题目类似，我们从两点之间的关系入手，建立最小割模型。

初始化 ans 为所有问号标记都为白色标记时的喜悦值，再将 ans 加上问号标记个数的4倍。那么：

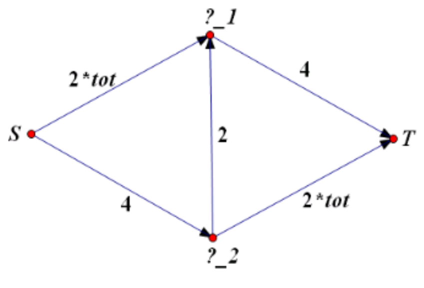
1.问号标记变为白色标记， ans 要减4。

2.问号标记变为蓝色标记，若它周围有 tot 个一开始就是蓝色的标记， ans 要减去 $2 \cdot tot$ 。

3.两个相邻的问号标记 ?_1 和 ?_2 都变为蓝色标记， ans 要减去2。

我们尝试像上一个题目一样求边的容量，但难以求出。

可以使用下图这样的建图方式。



其中，?_1 保留与 S 的连边，表示变成白色标记；保留与 T 的连边，表示变成蓝色标记。?_2 保留与 S 的连边，表示变成蓝色标记；保留与 T 的连边，表示变成白色标记。只有相邻两个问号标记都变为蓝色标记时，连接这两个问号标记的边才会被割掉，使 ans 减小2。

将日程表黑白染色，相邻的格子颜色不同，就会形成二分图。一种点用 ?_1 的连接方式，另一种点用 ?_2 的连接方式。

ans 减去最小割即为最大总喜悦值。

3.3 小结

最小割可以解决一些存在收益冲突的收益最大化问题。在网络流图中，任何一条 $S \to T$ 的路径上，都需要割掉至少一条边。利用这个性质，可以使收益的冲突在网络流模型中得到体现。

4 从费用流角度建模

给网络流增加一个因素：费用。这就变成了费用流问题。

费用流相比于最大流，能够解决更多的问题。

4.1 建模举例

例 8. *(序列)*

给出一个长度为 n 的正整数序列 A_i。选出一个子序列，使得原序列的任意一个长度为 m 的连续子序列中，被选出的元素数量不超过 k 个。

最大化选出的子序列中元素的和。

$1 \le n \le 1000, 1 \le m, k \le 100$

一个长度为 m 的连续子序列中至多选择 k 个元素。可以将问题转化一下，不是选择1次，而是选择k次，但在同一次选择中，任意一个长度为 m 的连续子序列中至多选择一个元素。

不难证明，原问题的一种合法方案，在转化后的问题中有等价的合法方案。转化后的问题中的一种合法方案，在原问题中也合法。

转化后的问题就比较好解决了，可以这样来建图：

建立源点 S 、汇点 T 。对序列中的第 i 个元素建立一个点 P_i 。

S 向 P_1 连一条容量为 k ，费用为0的边。

$P_i\ (1 \le i < n)$ 向 P_{i+1} 连一条容量为 k ，费用为0的边。

P_n 向 T 连一条容量为 k ，费用为0的边。

$P_i\ (1 \le i \le n-m)$ 向 P_{i+m} 连一条容量为1，费用为 A_i 的边。

$P_i\ (n-m+1 \le i \le n)$ 向 T 连一条容量为1，费用为 A_i 的边。

求最大费用最大流，最大费用就是最大元素和。

这道题目需要将原问题转化成更容易进行建模的问题，然后用网络流解决。

例 9. *(WC 2007, 剪刀石头布)*

在一些一对一游戏的比赛（如下棋、乒乓球和羽毛球的单打）中，我们经常会遇到A胜过B，B胜过C而C又胜过A的有趣情况，不妨形象的称之为剪刀石头布情况。有的时候，无聊的人们会津津乐道于统计有多少这样的剪刀石头布情况发生，即有多少对无序三元组(A, B, C)，满足其中的一个人在比赛中赢了另一个人，另一个人赢了第三个人而第三个人又胜过了第一个人。注意这里无序的意思是说三元组中元素的顺序并不重要，将(A, B, C)、(A, C, B)、(B, A, C)、(B, C, A)、(C, A, B)和(C, B, A)视为相同的情况。

有 n 个人参加一场这样的游戏的比赛，赛程规定任意两个人之间都要进行一场比赛：这样总共有 $\frac{n(n-1)}{2}$ 场比赛。比赛已经进行了一部分，我们想知道在极端情况下，比赛结束后最多会发生多少剪刀石头布情况。即给出已经发生的比赛结果，而你可以任意安排剩下的比赛的结果，以得到尽量多的剪刀石头布情况。

$1 \le n \le 100$

直接求(A, B, C)构成剪刀石头布情况的数量不好求，可以考虑(A, B, C)不构成剪刀石头布的情况。

如果(A, B, C)不构成剪刀石头布的情况，那么A、B、C中，一个人赢了2场，一个人赢了1场，一个人赢了0场。

若第 i 个人赢了 w_i 场，那么 $\sum_i \frac{w_i(w_i-1)}{2}$ 就是不构成剪刀石头布情况的(A, B, C)的数量。

最大化剪刀石头布情况数量，就是最小化不满足的数量，就是最小化 $\sum_i \frac{w_i(w_i-1)}{2}$。

可以建出网络流模型：

建立源点 S 和汇点 T。

对第 i 个人建立点 P_i，P_i 向 T 连一条容量为 n 的边。

对未进行的 i、j 之间的比赛建立点 $C_{i,j}$，S 向 $C_{i,j}$ 连一条容量为1的边，$C_{i,j}$ 向 P_i、P_j 各连一条容量为1的边。

这样求出的最大流就是一种合法方案。P_i 与 T 相连的边的流量，就是第 i 个人在这些未进行的比赛中赢的次数。

为了最小化 $\sum_i \frac{w_i(w_i-1)}{2}$，我们让这个值以费用的形式在网络流图中体现出来。

费用要加在 P_i 与 T 相连的边上。但每当这条边的流量加1时，费用不是不变的，而是一次比一次大。

可以把这条边拆成很多条，每条边的容量是1，费用是流量加1时的费用。

例如，若流量是 x 时，这条边的总费用是 $\frac{x(x-1)}{2}$，就把这条边拆成许多条容量为1的边，费用分别是 $0, 1, 2, ...$。因为求的是最小费用最大流，所以会先增广费用小的边。这样，流量是 x 时，费用就是 $\frac{x(x-1)}{2}$ 了。

这道题目，用到了补集转化的思想。同时，如果边的费用不固定，可以通过拆边来解决。

4.2 小结

费用流比最大流能够解决更多类型的问题，也更加灵活多变。在很多时候，需要先将原问题进行转化，再通过费用流解决问题。

5 流量平衡思想

有一些问题难以通过直观的方法直接进行网络流建模，但我们可以得到问题中变量之间的一些关系，一般可以写成若干个等式。

在网络流图中，除源点、汇点外，其他顶点都满足流量平衡。流量平衡也可以写成等式的形式。

有些问题中的等式，可以构造出网络流图，将等式以网络流图中流量平衡的形式表示出来。

下面的两个例子，就用到了这种思想。

5.1 建模举例

例 10. *(NOI2008, 志愿者招募)*

中奥成功后，布布经过不懈努力，终于成为奥组委下属公司人力资源部门的主管。布布刚上任就遇到了一个难题：为即将启动的奥运新项目招募一批短期志愿者。经过估算，这个项目需要 n 天才能完成，其中第 i 天至少需要 a_i 个人。布布通过了解得知，一共有 m 类志愿者可以招募。其中第 i 类可以从第 s_i 天工作到第 t_i 天，招募费用是每人 c_i 元。新官上任三把火，为了出色地完成自己的工作，布布希望用尽量少的费用招募足够的志愿者，但这并不是他的特长！于是布布找到了你，希望你帮他设计一种最优的招募方案。

$1 \le n \le 1000, 1 \le m \le 10000$

假设共3天，第 i 天招募 p_i 人。

共有三类志愿者，分别是：

从第1天到第3天，费用为 c_1 ，招募了 b_1 人。

从第2天到第3天，费用为 c_2 ，招募了 b_2 人。

从第1天到第2天，费用为 c_3 ，招募了 b_3 人。

可以列出不等式组：

$$\begin{aligned} p_1 = b_1 + b_3 &\ge a_1 \\ p_2 = b_1 + b_2 + b_3 &\ge a_2 \\ p_3 = b_1 + b_2 &\ge a_3 \end{aligned}$$

设第 i 天招募的志愿者人数超出最少人数 d_i 人，其中 $d_i \ge 0$ 。可以得到等式：

$$\begin{aligned} p_1 = b_1 + b_3 &= a_1 + d_1 \\ p_2 = b_1 + b_2 + b_3 &= a_2 + d_2 \\ p_3 = b_1 + b_2 &= a_3 + d_3 \end{aligned}$$

相邻两个等式相减，得到新的等式组：

$$
\begin{aligned}
p_1 &= b_1 + b_3 = a_1 + d_1 \\
p_2 - p_1 &= b_2 = a_2 - a_1 + d_2 - d_1 \\
p_3 - p_2 &= -b_3 = a_3 - a_2 + d_3 - d_2 \\
-p_3 &= -b_1 - b_2 = -a_3 - d_3
\end{aligned}
$$

整理一下，得到：

$$
\begin{aligned}
b_1 + b_3 - a_1 - d_1 &= 0 \\
b_2 - a_2 + a_1 - d_2 + d_1 &= 0 \\
-b_3 - a_3 + a_2 - d_3 + d_2 &= 0 \\
-b_1 - b_2 + a_3 + d_3 &= 0
\end{aligned}
$$

网络流图中，除了源点、汇点，其他顶点都满足流量平衡。若流出的流量记为正，流入的流量记为负，那么流入流出流量的代数和为0。可以对上面的每一个等式各建立一个点，等式表示的就是这个点的流量平衡。

对于每一个变量 b_i、 d_i，都恰好在两个等式中出现了，而且在一个等式中为正，一个等式中为负。网络流图中，一条连接 x、 y 的边，在 x、 y 点的流量平衡等式中各出现一次，一次为正，一次为负。所以，每一个变量 b_i、 d_i，都可以作为网络流图中的一条边。

对于常量 a_i，也在两个等式中出现，一次为正，一次为负。为正时，表示流出，可以从这个点向汇点连边。为负时，表示流入，可以从源点向这个点连边。

需要最小化 $\sum b_i \cdot c_i$，可以将这个值以费用的形式表示出来。

可以这样来建图：

建立源点 S 和汇点 T。

建立 $n+1$ 个点，点编号为 $1 \sim n+1$，表示 $n+1$ 个等式。

第 $i\,(2 \le i \le n+1)$ 个点向第 $i-1$ 个点连一条容量为正无穷，费用为0的边。

第 i 类志愿者可以从第 s_i 天工作到第 t_i 天，费用是每人 c_i 元。那么就从第 s_i 个点向第 t_i+1 个点连一条容量为正无穷，费用为 c_i 的边。

对于第 i 个点，若 $a_{i-1} - a_i$ 为正数，就从这个点向 T 连一条容量为 $a_{i-1} - a_i$，费用为0的边。否则就从 S 向这个点连一条容量为 $a_i - a_{i-1}$，费用为0的边。在这里，我们认为 a_0 和 a_{n+1} 的值为0。

由于 a_i 是常量，所以 S 连出的边、连入 T 的边都必须满流。原图的最大流满足这个条件。

为了最小化费用，所以需要求最小费用最大流。

例 11. *(World Finals 2011, Chips Challenge)*

在一个 $n \times n$ 网格里放部件。其中一些格子已经放了部件，一些格子不能放部件，其他格子可以放也可以不放。要求第 x 行的总部件数等于第 x 列的总部件数。为了保证散热，任意行/列的部件不能超过总部件数的 $\frac{A}{B}$。求最多能放多少部件。

$1 \le n \le 40$

用 $a_{i,j}$ 表示第 i 行第 j 列的格子是否放部件，1表示放，0表示不放。用 t_i 表示第 i 行的部件总数，同时也是第 i 列的部件总数。可以列出等式：

$$t_i = a_{i,1} + a_{i,2} + ... + a_{i,n} \quad (1 \le i \le n)$$
$$t_i = a_{1,i} + a_{2,i} + ... + a_{n,i} \quad (1 \le i \le n)$$

整理一下，得到：

$$a_{i,1} + a_{i,2} + ... + a_{i,n} - t_i = 0 \quad (1 \le i \le n)$$
$$t_i - a_{1,i} - a_{2,i} - ... - a_{n,i} = 0 \quad (1 \le i \le n)$$

还有一个限制，是任意行/列的部件不能超过总部件数的 $\frac{A}{B}$。枚举总部件数 tot，就能确定出每行、每列最多放的部件数 $maxt$ 了。求出这种情况下最多放的总部件数 ans。若 $ans \ge tot$，说明方案合法。

通过枚举，确定 $maxt$ 后，可以这样建图：

建立点 A_i 表示第 i 行部件数量的计算式$a_{i,1} + a_{i,2} + ... + a_{i,n} - t_i = 0$。

建立点 B_i 表示第 i 列部件数量的计算式$t_i - a_{1,i} - a_{2,i} - ... - a_{n,i} = 0$。

点 B_i 向点 A_i 连接流量上界为 $maxt$，下界为0，费用为1的边。

若 $a_{i,j}$ 必须为1，就从点 A_i 向点 B_j 连接流量上下界均为1，费用为0的边。

若 $a_{i,j}$ 可以为1，也可以为0，就从点 A_i 向点 B_j 连接流量上界为1，下界为0，费用为0的边。

这个网络流图的一个可行流就是一种合法方案。最大费用可行流就是最优方案。

对所有的 $maxt$ 都求出最优方案，取所有合法方案中的最优解作为最终方案。

5.2 小结

将问题中变量之间的关系以等式的形式表示出来，然后与网络流图中描述流量平衡的等式进行联系。可以将原问题对应到网络流图上。

多数网络流模型都可以用流量平衡思想来理解。所以可以用流量平衡的思想解决大部分网络流问题。但流量平衡思想的缺点是不够直观，有时会把问题变复杂。不过在一些直观上难以理解或难以想到的问题上，流量平衡的思想能够发挥很大的作用。

6 总结

网络流的建模角度非常多，如最大流角度、最小割角度、费用流角度、平衡流角度等。

网络流建模时会用到的技巧也有很多，如拆点、拆边、建分层图等。

网络流建模时，有时需要分析问题，将原问题转化为比较容易进行建模的问题。

有时，网络流建模需要和其他算法结合才能解决问题，如二分、补集转化等。

总之，网络流建模的内容十分丰富。

希望有同学能够从本文中获得灵感，发现新的、更加巧妙的网络流建模方法。

7 参考文献

[1] 刘汝佳、陈锋，《算法竞赛入门经典——训练指南》，清华大学出版社

[2] 胡伯涛，《最小割模型在信息学竞赛中的应用》