

利用图论模型解决几类特殊的线性规划问题

METHODS BASED ON GRAPH MODELS TO SOLVE SPECIAL KINDS OF LINEAR PROGRAMMING

高 2013 级 13 班李凌霄
Class 13, Grade 2013, Lingxiao Li

2011-08-30

摘要 Abstract

线性规划 (Linear Programming, 简称 LP) 是最优化问题中的重要领域之一, 研究线性约束下线性目标函数的最值问题, 广泛应用于军事作战、经济分析、经营管理和工程技术等方面。本文提出了几类特殊的线性规划问题, 通过与图论模型相联结, 得到了较好的解决。

Linear Programming is a mathematical method for determining a way to achieve the best outcome in a given mathematical model for some list of requirements represented as linear relationships, which has proved useful in modeling diverse types of problems in planning, routing, scheduling, assignment, and design.

This thesis suggests several kinds of LP problems with speciality which can be solved with analogy to classic graph models, such as shortest path, weighted perfect matching, and network flow.

目录 Content

1	三角形不等式的启示 Inspiration of Triangle Inequality	2
1.1	预备知识 Preliminary	2
1.1.1	松弛性质 Relaxation Property	2
1.1.2	几个常用算法简介 Some Known Algorithms	2
1.2	差分约束系统 System of Difference Constraints	2
1.3	[例一]SCOI2011 糖果	3
1.3.1	描述	3
1.3.2	输入格式	3
1.3.3	输出格式	3
1.3.4	数据范围	3
1.3.5	分析	3
1.4	一些思考 Some Thoughts	4
2	网络流的守恒思想 The Conservation of Network Flow	4
2.1	预备知识 Preliminary	4
2.1.1	流的性质 Properties of Flow	4
2.1.2	求最大流的算法 Algorithms for Maximum Flow Problem	4
2.1.3	费用流 Minimum Cost Flow	4
2.2	[例二]CTSC2005 玩具的重量	4
2.2.1	描述	4
2.2.2	输入格式	5
2.2.3	输出格式	5
2.2.4	数据范围	5
2.2.5	分析	5
2.3	[例三]NOI2008 志愿者招募	6
2.3.1	描述	6
2.3.2	输入格式	6

2.3.3	输出格式	6
2.3.4	数据范围	6
2.3.5	分析	6
3	KM 算法与二元规划 Kuhn - Munkres Algorithm and the System of Two Variables	7
3.1	预备知识 Preliminary	7
3.2	[例四]SGU206 Roads	7
3.2.1	描述	7
3.2.2	数据范围	7
3.2.3	分析	7
4	总结	8

§ 1 三角形不等式的启示 Inspiration of Triangle Inequality

§ 1.1 预备知识 Preliminary

图论中最为经典的例子莫过于最短路了。在这里只讨论单源最短路 (SSSP)。设 $f[i]$ 为目前已知的从源点 s 到 i 顶点的最短路径估计, $f^*[i]$ 为实际最短路。则显然 $f[i] \geq f^*[i]$ 。

1.1.1 松弛性质 Relaxation Property

令 $d[u][v] = \begin{cases} \min\{p.weight\} & \text{存在 } u \text{ 到 } v \text{ 的路径 } p \\ \infty & u \text{ 到 } v \text{ 不存在路径} \end{cases}$

可以得到三角形不等式:

$$\forall u, v \in V, f^*[u] \leq f^*[v] + d[v][u] \quad (1)$$

若发现对于 v 节点, 有 $f[v] > f[u] + d[u][v]$, 则可以更新 $f[v]$ 使其更接近目标 $f^*[v]$, 即让 $f[v] = f[u] + d[u][v]$ 。这便是松弛操作 (relaxation)。而几乎所有的最短路算法都依赖于松弛操作。

若对任何节点都无法松弛, 那么有 $f = f^*$ 。反之亦然。证明在此略过。

1.1.2 几个常用算法简介 Some Known Algorithms

算法名称	时间复杂度	实现原理
Bellman-Ford	$O(V E)$	如果最短路存在, 则每个顶点最多经过一次, 因此不超过 $ V - 1$ 条边。由最优性原理, 只需依次考虑经过边数为 $1, 2, \dots, V - 1$ 的最短路。对每条边松弛 $ V - 1$ 次即可。
SPFA	$O(k E)$	此算法为 Bellman-Ford 的改进。并不是每次松弛都是有价值的。只需每次将 f 值变化了的点加入队列, 更新次数降低, 实际运行效果非常好。
Dijkstra	$O(E + V \log V)$	采用标记永久化技术 (亦即贪心), 保证每次加入点时已求出最短路。不适用于含负边权的图。
DP(递推)	$O(V + E)$	按拓扑关系递推。只适用于 DAG (有向无环) 图。

从上表可以看出, 对于特殊性较高的图, 往往可以利用性质, 设计出非常高效的算法。

§ 1.2 差分约束系统 System of Difference Constraints

差分约束系统是指许多形如 $x_{a_k} - x_{b_k} \leq c_k$ 的不等式组。如果求出一个特解 $\{x_i\}$, 那么所有 $\{x_i + k\}$ 都是一组解¹。所以关键就在于找到一组特解。

如果给约束条件变变形, 就成了 $x_{a_k} \leq x_{b_k} + c_k$, 这不就是(1)吗?

给每个 x_i 建立一个对应的顶点, 每个约束条件建立一条边, 求一遍 SSSP, 得到的 $f = f^*$ 就是一组解。

¹但无法生成所有解

如果约束条件是 $x_i - x_j \geq c_k$ 呢? 其实只要两端同时乘以 -1, 就转化为了原问题。即便不转化, 对比(1), 我们发现本质是在求最长路, 最长路的对偶问题就是边权为负的最短路!

还有一些特殊情况需要注意。比如对单个未知数的限制。设一附加源点 s 。由于 $f[s] = 0$, 所以对 x_i 的约束也可变形为 $x_i - x_s \leq c_k$ 。从 s 直接连边即可解决。

如果边权全为正, 可以采用效率更高的 Dijkstra 算法。

下面, 看看差分约束系统在信息学中的应用吧。

§ 1.3 [例一]SCOI2011 糖果

1.3.1 描述

幼儿园里有 N 个小朋友, $lxhgww$ 老师现在想要给这些小朋友们分配糖果, 要求每个小朋友都要分到糖果。但是小朋友们也有嫉妒心, 总是会提出一些要求, 比如小明不希望小红分到的糖果比他的多, 于是在分配糖果的时候, $lxhgww$ 需要满足小朋友们的 K 个要求。幼儿园的糖果总是有限的, $lxhgww$ 想知道他至少需要准备多少个糖果, 才能使得每个小朋友都能够分到糖果, 并且满足小朋友们所有的要求。

1.3.2 输入格式

输入的第一行是两个整数 N, K 。

接下来 K 行, 表示这些点需要满足的关系, 每行 3 个数字, X, A, B 。

如果 $X = 1$, 表示第 A 个小朋友分到的糖果必须和第 B 个小朋友分到的糖果一样多;

如果 $X = 2$, 表示第 A 个小朋友分到的糖果必须少于第 B 个小朋友分到的糖果;

如果 $X = 3$, 表示第 A 个小朋友分到的糖果必须不少于第 B 个小朋友分到的糖果;

如果 $X = 4$, 表示第 A 个小朋友分到的糖果必须多于第 B 个小朋友分到的糖果;

如果 $X = 5$, 表示第 A 个小朋友分到的糖果必须不多于第 B 个小朋友分到的糖果;

1.3.3 输出格式

输出一行, 表示 $lxhgww$ 老师至少需要准备的糖果数, 如果不能满足小朋友们的所有要求, 就输出 -1。

1.3.4 数据范围

对于 30% 的数据, 保证 $N < 100$;

对于 100% 的数据, 保证 $N < 100000$;

对于所有的数据, 保证 $K \leq 100000, 1 \leq X \leq 5, 1 \leq A, B \leq N$ 。

1.3.5 分析

可以发现, 本题就是十分明显的差分约束系统, 只不过要最小化 $\sum_{i=1}^n x_i$ 。为每个未知数建立一个对应顶点, 同时为满足隐含的 $x_i \geq 1$, 建一个附加源 s , 向其他每个点连一条长度为 1 的边。求 SSSP。如果出现负环, 则无解, 输出 -1; 否则, $\sum_{i=1}^n f[i]$ 就是答案。

但是, 点数和边数都达到了 100000, 单纯用 SPFA 很难在规定时限 1s 内通过所有测试点。

优化 1: SLF(Small Label First)

设队首元素为 h , 队列中要加入节点 u , 在 $f[u] \leq f[h]$ 时加到队首而不是队尾, 否则和普通的 SPFA 一样加到队尾。

优化 2: LLL(Large Label Last)

设队列 Q 中的队首元素 h , f 的平均值 $\bar{f} = \frac{\sum_{i \in Q} f[i]}{|Q|}$, 每次出队时, 设出队元素为 u , 若 $f[u] > \bar{f}$, 把 u 移到队列末尾, 如此反复, 直到找到一个 u 使 $f[u] \leq \bar{f}$, 将其出队。

实践证明, 加上以上两个优化就可以 AC。但渐进时间复杂度并未得到改善。我们尝试寻找特殊, 来寻求突破。

我们发现, 本题的二元约束条件的常数要么为 0, 要么为 -1。如果没有环, 就可以直接 DFS 或者 BFS 来在线性时间内求最短路。那么, 能不能把环强行去掉? 考察会出现环的情况, 环上的边权只可能是 0、-1! 如果一个强连通分量 (SCC) 中的边权不全为 0, 只能是无解。否则, 强连通分量中所有点完全没有区别, 我们可以收缩所有 SCC, 得到一个 DAG 图, 这样就为 DP 创造了条件。

线性时间就能非常完美地解决问题。还需注意的是,要用上 64 位整型,tarjan 求 SCC 可能要手写栈。

§ 1.4 一些思考 Some Thoughts

如果约束条件未知数的系数不再是 0、1,而是任意非负实数呢?即求解不等式组 $\{p_k x_{a_k} - q_k x_{b_k} \leq c_k \mid p_k, q_k \geq 0\}$ 。

先做恒等变换,等价于求解 $\{x_{a_k} \leq \frac{q_k}{p_k} x_{b_k} + \frac{c_k}{p_k}\}$,如果用 SPFA 动态更新,直到无法获得更优解,就和最短路没什么两样了。只是复杂度不好估计。

如果两元关系不再是作差,而是之和,又如何在满足约束条件的情况下最小化变元之和?如果不再是两元,而是更多,又怎么办?最高次不再是 1,单纯形法²也失效,图能辅助解决吗?这就是接下来所要探索的内容。

§ 2 网络流的守恒思想 The Conservation of Network Flow

§ 2.1 预备知识 Preliminary

流网络 (flow network) 是一个有向图 $G = (V, E)$,其中每条 $(u, v) \in E$ 有一个非负容量上界 $c(u, v) \geq 0$ 。规定:若 $(u, v) \notin E$, $c(u, v) = 0$ 。在有下界的网络流中还存在容量下界 $b(u, v)$,与容量上界类似。网络中有两个特殊点:源 s 与汇 t 。

2.1.1 流的性质 Properties of Flow

流网络 G 的流 (**flow**) 是一个函数 $f: V \times V \rightarrow \mathbf{R}$,且满足下列三个性质:

1. 容量限制 $\forall u, v \in V, f(u, v) \leq c(u, v)$ 。在有上下界的网络流中,还满足 $\forall u, v \in V, b(u, v) \leq f(u, v) \leq c(u, v)$
2. 反对称性 $\forall u, v \in V, f(u, v) = -f(v, u)$ 。
3. 流守恒性 $\forall u \in V - \{s, t\}, \sum_{v \in V} f(u, v) = 0$ 。

称 $f(u, v)$ 为从 u 到 v 的流。

流 f 定义为 $f = \sum_{u \in V} f(s, u)$ 。最大流 f_{\max} 定义为 $\max\{f\}$ 。

2.1.2 求最大流的算法 Algorithms for Maximum Flow Problem

求无下界最大流常用连续最短增广路算法 (Dinic's Algorithm),复杂度为 $O(|V|^2|E|)$,实际效果良好,特别地,在二分图中复杂度更低,为 $O(\sqrt{|V|}|E|)$ 。详见 [3]。

若存在下界,往往采用参数搜索,即二分答案后,转化为无源汇的网络流,再结合最大流算法求解。具体参见 [4]。

2.1.3 费用流 Minimum Cost Flow

若给每条边加上一个权值 $w(u, v)$,定义流的费用为 $g = \sum_{u, v \in V} f(u, v) \times w(u, v)$,其中 $f(u, v) > 0$ 。

类似定义最小费用流的费用 g_{\min} ,最大费用流的费用 g_{\max} 。

求最小费用流简单快速的办法就是最短路增广,这样可以保证每次增广的都是最小费用的路径。由于含负权,采用 SPFA。复杂度的界比较高,但实际上远小于复杂度上界。

§ 2.2 [例二]CTSC2005 玩具的重量

2.2.1 描述

冰冰有三个玩具:皮卡丘、维妮孙悟空和芭比娃娃。她并不知道这些玩具的具体重量(采用 NOI 单位),但是知道每个玩具重量的大概范围,如下表:

²一种解决线性规划的通法

玩具名称	最小可能重量	最大可能重量
皮卡丘	1	3
维妮孙悟空	2	4
芭比娃娃	3	5

这些范围太粗略,冰冰希望能把它们缩小一些。正好佳佳有一个电子天平,不仅可以告诉你左右两边是否一样重,还可以告诉你左边比右边重(或轻)多少。天平很大,左右两边都可以放任意多件玩具。冰冰向佳佳借电子天平,希望能算出每个玩具的精确重量。佳佳为了考验冰冰,只允许她把任意一个玩具往天平的左侧和右侧最多各放一次。例如,如果她曾经把皮卡丘放在天平的左侧,则她不能再次把它放在天平的左侧。冰冰同意了。她一共称量了两次,称量结果如下:

第一次:左边放皮卡丘,右边放维妮孙悟空,返回右边比左边重 1;

第二次:左边放维妮孙悟空,右边放芭比娃娃,返回左边比右边重 1;

可以确定三个玩具的重量一定是 3, 4, 3, 也就是说,通过称量结果所得到的更新后的重量范围是:

玩具名称	最小可能重量	最大可能重量
皮卡丘	3	3
维妮孙悟空	4	4
芭比娃娃	3	3

根据称量结果所得到的精确范围冰冰以后还会买很多很多玩具,她不想每次都自己计算每个玩具的重量。她需要写一个程序计算每个玩具最精确的重量下限和上限,你能帮她吗?

2.2.2 输入格式

输入文件第一行包含两个整数 n 和 m , 即玩具的个数和称量的次数。第二行包含 $2n$ 个数, 第 $2i-1$ 个数和第 $2i$ 个数分别表示第 i 个玩具的重量初始下限和初始上限。以下 m 行, 每行前三个数 L, R, D 表示左边的玩具数、右边的玩具数和左右两边的重量差 ($L, R \geq 0$), 接下来的 L 个数为天平左边的玩具编号, 再接下来的 R 个数为天平右边的玩具编号。输入保证每个玩具在天平的每一边最多出现一次。

2.2.3 输出格式

输出包含 $2n$ 个整数, 第 $2i-1$ 个数和第 $2i$ 个数分别表示第 i 个玩具的重量下限和上限, 即最小可能的整数重量和最大可能的整数重量。如果无解(可能是天平坏了), 只输出一个数 -1 。

2.2.4 数据范围

对于 50% 的数据, 满足 $3 \leq n \leq 10, 1 \leq m \leq 5$;

对于 100% 的数据, 满足 $3 \leq n \leq 2000, 1 \leq m \leq 100$, 重量上限不超过 20000。

2.2.5 分析

“每个玩具在天平的每一边最多出现一次”这个条件十分诱人, 而每次称量结果为一等式, 恰好能对应上流守恒性。我们将每次称量作为一个点, 由于最多出现一次, 玩具可作为边。

- 如果左右两边各出现一次, 则在对应称量的两点间连一条边。
- 如果只在左边出现一次, 则从源向该点连一条边。
- 如果只在右边出现一次, 则从该点向汇连一条边。
- 如果没有出现, 则从源向汇连一条边。

边的容量上下限均取玩具重量的上下界。那么对应的流量就是玩具的重量。每一个可行流就是一组合法解。

那么, 如何确定每个物品的解区间? 如果我们强制是一条边的容量上界等于容量下界, 就等于强制压流。而在已知一组解的情况下, 解区间被划成了两块单调的区间。设 0 表示非法, 1 表示合法, 那么必然是

0000...000011111...11111...111100000...000

左右两边分别二分判可行性即可。

根据等式联想到流量平衡,由上下界网络流强制限流的特点来判可行性,实在是巧妙。即便是等式,也能通过添加辅助变量,来变成等式。“左右两边最多各出现一次”也是不可或缺的条件,通过简单的变形,往往也能转化成这个条件,比如说下面这道题。

§ 2.3 [例三]NOI2008 志愿者招募

2.3.1 描述

申奥成功后,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。

2.3.2 输入格式

第一行包含两个整数 N, M , 表示完成项目的天数和可以招募的志愿者的种类。

接下来的一行中包含 N 个非负整数,表示每天至少需要的志愿者人数。

接下来的 M 行中每行包含三个整数 S_i, T_i, C_i , 含义如上文所述。为了方便起见,我们可以认为每类志愿者的数量都是无限多的。

2.3.3 输出格式

仅包含一个整数,表示你所设计的最优方案的总费用。

2.3.4 数据范围

30% 的数据中, $1 \leq N, M \leq 10, 1 \leq A_i \leq 10$;

100% 的数据中, $1 \leq N \leq 1000, 1 \leq M \leq 10000$, 题目中其他所涉及的数据均不超过 $2^{31} - 1$ 。

2.3.5 分析

设 X_i 为第 i 类志愿者招募的人数。则应满足

$$\forall j \in \mathbf{Z} \cap [1, N], \sum_{S_i \leq j \leq T_i} X_i \geq A_j \quad (2)$$

添加辅助变量 Y_j , 使(2)变为等式

$$\sum_{S_i \leq j \leq T_i} X_i - Y_j = A_j$$

等价于

$$\sum_{S_i \leq j \leq T_i} X_i - Y_j - A_j = 0$$

似乎又能利用流量守恒建图,但如何满足“左右各一次”,即在一个等式中符号为正,在另一个等式中符号为负?

题目有一个重要的性质没有用到:每类志愿者只管**连续的一段时间**。这能否为解题提供服务?

等式整体作差! 如果作差的话,许多变量就会两两相消,只有在连续的一段的两端会留下两个变量!

再看看目标函数

$$\text{最小化} \sum_{1 \leq i \leq M} X_i \times C_i$$

这不就是最小费用流吗? 只要在 X_i 对应的边上加权,其余边权值为 0,用最小费用最大流即可解决。

本题还存在另一种直接建图的方法,但不如不等式来的清晰,在此就不作介绍。

§3 KM 算法与二元规划 Kuhn - Munkres Algorithm and the System of Two Variables

§3.1 预备知识 Preliminary

KM 算法用来解决最大权匹配问题: 在一个二分图内, 左顶点集合为 X , 右顶点集合为 Y , 现对于每条边 (i, j) 有权 $w(i, j)$, 求一种匹配使得

$$\sum_{(i, j) \text{ 在匹配中}} w(i, j)$$

最大。

KM 算法的核心思想是: 给每一个顶点一个标号 (顶标) 将原问题转化为求完备匹配。设 X_i 的顶标为 A_i , Y_i 的顶标为 B_i , 那么在算法执行过程中的任意时刻, $A_i + B_j \geq w(i, j)$ 始终成立。

KM 的正确性基于以下定理: 若由二分图中所有满足 $A_i + B_j = w(i, j)$ 的边 (i, j) 构成的子图 (称做相等子图) 有完备匹配, 那么这个完备匹配就是二分图的最大权匹配。

KM 算法的复杂度为 $O(|V|^3)$, 详见 [5]。

在第一节中提出了一个问题: 形如 $x_{a_k} + x_{b_k} \geq c_k$ 的不等式组如何求解? 仔细想想可以发现, 求特解没有意义, 因为每个解都相对独立, 任意加上一个非负实数都不会破坏不等式。

那如何最小化 $\sum x_i$? 且看下面这个题。

§3.2 [例四]SGU206 Roads

3.2.1 描述

一个遥远的王国拥有 m 条道路连接着 n 个城市 m 条道路中有 $n-1$ 条石头路, 其余有 $m-n+1$ 条泥土路, 任意两个城市之间有且仅有一条完全用石头路连接起来的道路。每条道路都有一个唯一确定的编号, 其中石头路编号为 $1, \dots, n-1$, 泥土路编号为 n, \dots, m 。

每条道路都需要一定的维护费, 其中第 i 条道路每年需要 C_i 的费用来维护。最近该国国王准备只维护部分道路以节省费用。但是他还是希望人们可以在任意两个城市间互达。

国王需要你提交维护每条道路的费用, 以便他能让他的大臣来挑选出需要维护的道路, 使得维护这些道路的费用是最少的。尽管国王不知道走石头路和走泥土路的区别, 但是对于人民来说石头路似乎是更好的选择。为了人民的利益, 你希望维护的道路是石头路。这样你不得不在提交给国王的报告中伪造维护费用。你需要给道路 i 伪造一个费用 D_i , 使得这些石头路能够被挑选。为了不让国王发现, 你需要使得真实值与伪造值的差值和, 即 $f = \sum_{i=1}^m |D_i - C_i|$ 尽量小。

国王的大臣当然不是白痴, 全部由石头路组成的方案如果是一种花费最小的方案, 那么他会选择这种方案。

求出真实值与伪造值的差值和的最小值, 以及得到该最小值的方案, 即每条边的修改后的边权 D_i 。

3.2.2 数据范围

$$2 \leq n \leq 60, n-1 \leq m \leq 400.$$

3.2.3 分析

显然为了得到最小值, 应降低石头路的费用而提升泥土路的费用。设第 i 条道路费用调整了 δ_i , 由最小生成树的性质可知, 每加入一条非树边都会出现一个环, 设加入的泥土路为 r , 而为了使石头路依旧是树边, 必然有

$$C_r + \delta_r \geq \max_{k \text{ 在环中}} \{C_k - \delta_k\}$$

如果将上式的 \max 去掉, 就变成了对每一条在环上的石头路的限制:

$$\bigcup_{k \text{ 在环中}} \{C_r + \delta_r \geq C_k - \delta_k\}$$

等价于

$$\bigcup_{k \text{ 在环中}} \{\delta_r + \delta_k \geq C_k - C_r\}$$

式子右为一常数, 目标是最小化 $\sum_{i=1}^m \delta_i$ 。这就是刚刚提出的问题。显然这可以用单纯形解, 但未免也太大才小用了, 较高的时间和编程复杂度也使人望而生畏。

让我们在来找找特殊, 看看模型是否抽象得足够准确。不难发现, 每个不等式的两个变量都是一条树边, 一条非树边, 这很自然的让人联想到二分图。再关注下不等式。难道不觉得眼熟吗? 求二分图最佳匹配的 KM 算法!

如果把每个 δ 对应成顶点, 那么每个不等式就是一条边, 边权即为 $C_k - C_r$ 。而算法执行结束后每个顶点的可行顶标, 就是 δ 的值, 跟 KM 算法完全一致!

在接触本题前, 很多人包括我在内, 一直以为 KM 算法没什么用, 速度又没费用流好, 完全可以用网络流代替, 就没有去学。很多算法最核心的还是在思想上, 广博地学习, 能潜移默化地影响我们的思维方式, 为我们提供类比的素材, 看似没有用的东西可能会带来巨大帮助。

思考: 如果约束条件为 $x_{a_k} + x_{b_k} \leq c_k$, 并且满足二分图性质, 如何最大化 $\sum x_i$?
其实很简单, 只需改成求最小权最佳匹配。

思考: 如果不再满足二分图性质?

一般图的最佳匹配? 好像不是这么回事, KM 算法根本不适用了。可能只有老老实实用线性规划的一般方法了。

§4 总结

一些看似与图论无关的问题, 在合理的数学抽象之后, 通过类比, 成功地转化为了可解决的图论模型。整个过程看似天马行空, 实际上只不过是几个简单问题的组合。在这其中, 起着串联作用的就是类比思想。这如同一位大牛所说: “养成主动类比的习惯是非常有效的, 尤其在山穷水尽之时, 其往往能够让我们柳暗花明! 当然, 类比思维建立在非常扎实的基础算法功底之上。”

参考文献 References

- [1] 《Introduction to Algorithm》
- [2] 《新编实用算法的分析与程序设计》
- [3] 王欣上《浅谈基于分层思想的网络流算法》
- [4] 周源《一种简易的方法求解流量有上下界的网络中网络流问题》
- [5] wikipedia http://en.wikipedia.org/wiki/KM_algorithm
- [6] 王俊《浅析二分图匹配在信息学竞赛中的应用》
- [7] 冯威《数与图的完美结合 ---- 浅析差分约束系统》
- [8] philhu《记录表》
- [9] Byvoid《NOI2008 employee 解题报告》
- [10] SGU206 解题报告 <http://crfish.blogbus.com/logs/62846336.html>