一些可以转化为网络流形式的线性规划问题

如果您发现本文有任何错误或者有歧义的地方请联系我, 我会核实后进行修正, 我很在意我写出的文本的质量

> 这是一篇笔记文, 记录一些 OI/ICPC/CCPC 中的网络流题目,
> 主要是帮助自己熟悉一下知识和技术:
>
> - 最大流/最小割/最小费用最大流/特殊问题 (二分图上的最大独立集, 最小点覆盖, 最大匹配) 等图论相关的最优化问题的线性规划形式的特点
> - 哪些形式的线性规划可以转化为上述形式
> - 上述过程中的常见模型与操作技巧

关于线性规划

一些图论最优化问题的线性规划形式

例题:BZOJ1458 士兵占领.

> BZOJ 已经不复存在了, 但是你可以在 OI-archive 上面找到这个题目.
>
> 给一个 $n \times m$ 的矩阵 $0/1$ 矩阵, 要求一些 k 个元素 (x_i, y_i) 是 0,
> 约束: $(\forall i, \sum_j a_{i,j} \geq r_i) \wedge (\forall i, \sum_j a_{j,i} \geq c_i)$.
> 最小化 $\sum_i \sum_j a_{i,j}$

传统做法

考虑一个极端情况, 放满 $nm - k$ 个 1, 如果某一行或者某一列上面的 1 数量仍然不够, 那么就是无解, 否则有解.

考虑在此基础上, 去掉尽量多的 1 仍然满足约束.
令 R_i/C_i 为第 i 行/列上, 可以为 1 的元素数量减去 r_i/c_i 即最大可改进数量.
此时如果想要对 (x, y) 进行 $0 \to 1$ 那么需要 $R_x > 0 \wedge C_y > 0$,
进行改进操作之后, 令 ${R_x}' \leftarrow R_x - 1, {C_y}' \leftarrow C_y - 1$, 就可以继续进行上述改进操作.

我们发现这非常类似二分图多重匹配.

建立 $n + m$ 个点, 分别表示每一行/列,
以及 $S \to row(i)$ 的边, 容量为 R_i ; $col(i) \to T$ 的边, 容量为 C_i,
对于所有可以 $0 \to 1$ 的 (x, y) 连一条 $row(x) \to col(y)$ 的边, 容量为 1.
那么 $S \to T$ 的最大流, 就是这个二分图 (当然要去掉 S, T, 以及相关的边) 的最大多重匹配, 就是最大的可改进元素数量.

更少使用组合意义的推导

> 可以发现上面的推导, 用到了不少组合意义与直觉, 以及模型.
> 这些东西在复杂问题上是不怎么可靠的, 所以下面我们将会给出一个少使用组合意义, 少使用直观理解的推导.

$$\begin{cases} \forall (x, y), 0 \leq a_{x,y} \leq 1 \\ \\ \forall 1 \leq i \leq k, a_{x_i, y_i} = 0 \\ \\ \forall 1 \leq i \leq n, \sum_{j=1}^{m} a_{i,j} \geq r_i \\ \forall 1 \leq i \leq m, \sum_{j=1}^{n} a_{j,i} \geq c_i \end{cases} \qquad \min(\sum_{i=1}^{n} \sum_{j=1}^{m} a_{i,j})$$