NP-完全性

算法设计与分析小班课

2022 年 5 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.

NP-完全性一章是课程和过去的算法性质的课程的不同之处的开始, 在 NP-完全性的讨论中我们主要回答的是 "不可能" 的问题 (应该说是几乎不可能). 其中最重要的思想和方法就是**归约**. 某老师曾经说, 归约是人类解决问题和比较问题的自然思想, 它是使得计算机理论在看待问题时不同于数学的一个重要元素.

1 归约方法和基本问题

NP 是计算复杂性的一个类, 严格解说或者展开讨论计算复杂性则是另外课程的内容. 我们以下主要采用应用的态度, 就是重点看如何用归约的方法证明 NP-完全性.

很遗憾的是 (或许也让人感到兴奋), 证明一个问题是 NP-完全的一般地是很难的问题, 归约也没有固定的技巧——如果你能证明/解决一些具有实际意义的问题是 NP-完全或是难的, 通常就有可能发表论文. 下面从一些简单的问题中讲一些归约的直观, 主要是解释课本中的归约是怎么想出来的.

首先有两个细节问题. 第一是一个经典的笑话, 就是归约方向不能弄错. 要证明一个问题是困难的, 我们必须把一个**已知困难的问题**化归到它. 这是因为, 如果新问题能容易地解决, 已知困难的问题就也能容易解决, 产生矛盾. 所以归约方向是把老问题 $\Longrightarrow$ 新问题. 第二是若要证明 NP-完全性, 则不能忘记先验证问题在 NP 中. 我们能想到的每个自然的判定问题并非都 (明显) 属于 NP, 而且也有一些优化问题本身是属于 NP 的, 所以虽然这步简单但仍有意义.

问题 1 (不显然属于 NP 的 NP 问题) 给一个有限长度 0, 1 字符串的有限集合 $S = \{s_1, \ldots, s_k\}$, 我们称串 u 是 S 的**串联**, 如果对某个 $i_1, \ldots, i_t \in [k]$ 有 $u = s_{i_1} \cdots s_{i_t}$. 现在给两个集合 $A = \{a_1, \ldots, a_m\}, B = \{b_1, \ldots, b_n\}$, 问是否存在一个字符串 u, 同时是 A 和 B 的串联.

(1) 某同学说: 给一个字符串和相应的子串划分, 我们可以在多项式时间内检查它是否同时为 A, B 的串联, 因此这是 NP 内的问题. 说明这论证哪里存在问题.
(2) 事实上, 这问题确实在 NP 内, 证明之.

解 (1) 因为我们尚不清楚最小的证据字符串的长度是否达到指数级别.

(2) 我们证明, 如果答案为是, 那么最短的满足条件的字符串的长度不超过输入长度的多项式级别. 事实上, 不妨设 $m \le n$ 并令 L 为 $A \cup B$ 中最长的字符串的长度, 则其长度不超过 $(nL)^2$. 如果不然, 我们考虑 u 中的每个位置, 它在解中对应于某个 a_i 的第 k 个位置, 以及对应于某个 b_i 的第 k' 个位置. 根据抽屉原理, $|u| > (nL)^2$, 故必然存在两个位置 $p < p'$, p 对应于某个 a_i 的第 k 个位置, 以及对应于某个 b_j 的第 k 个位置, 那么现在可以把 $p, p+1, \ldots, p'-1$ 这些位置的字符删去, 仍然得到合法的解, 矛盾. □

回归主题. 总地来说, 归约的大方向是:

- 找出一个结构和新问题相似的已知困难的老问题. 最直接的就是问题的内容类似, 比如: 具有序列/排列性质的问题应考虑 HamiltonCycle, 具有选出 (满足一定要求的) 子集的应考虑 ExactCover/SetCover, IndependentSet, VertexCover, 具有划分/分类性质的应考虑 GraphColoring, 具有数值性质的应考虑 SubsetSum, 等等.
- 在看似不太一样的问题之间建立联系. 除了那些几乎等价的问题之外 (例如点覆盖和独立集, 这些请回忆集合论与图论的知识), 这步的宗旨——

$$\text{图} \longleftrightarrow \text{关系} \longleftrightarrow \text{集合} \quad \text{以及} \quad \text{组合优化问题} \longrightarrow \text{线性规划}.$$

  意思是说, 图上的边都是某种关系/性质, 而后者其实就是集合 $\times$ 集合; 另外许多组合优化问题都可写成 (整数) 线性规划, 由此可以找到一些灵感. 对于非平凡的问题, 一般问题规模需要作多项式级别的扩大.
- 如果什么办法也没有, 就从 3SAT 开始. 这里的 “可满足性约束” 就是其他组合结构中 “方案” 或者 “选择”. 具体地说有两种选择方式, 第一种是每个子句选一个文字成真, 第二种是每个变元选择真或假. 一般先从一种理解出发开始构造, 然后利用另一种理解修补得到最后的构造 (参看下面的小节和例题 34~38).

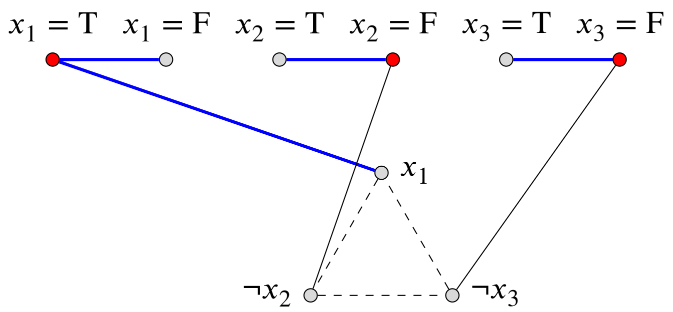
1.1 21 个 NP-完全问题

课本上的问题都是来源于 Karp 的 21 个 NP-完全问题, 这 21 个问题为何以及全部的归约可参照 [Kar72] (某些小班的选读论文是这篇). 前面的例子中, 我们主要解释课本中以 3SAT 起始的归约方法; 后面的例子中, 我们扩展几个课本上没有的问题.

例 2 (3SAT 到顶点覆盖) 从 3SAT 归约可以先从一个子句的例子开始思考. 比方说 $(x_1 \vee \neg x_2 \vee \neg x_3)$.

首先可以发现, 这个归约中应该要仔细调节问题的参数, 因为顶点覆盖中选多了顶点不会影响解的可行性. 先对每个变元造顶点 T, F 并相连 (这两个顶点选一个就能覆盖这条边), 表示变元的选择. 如何来表达子句可满足的呢? 考虑用三个顶点表示一个子句

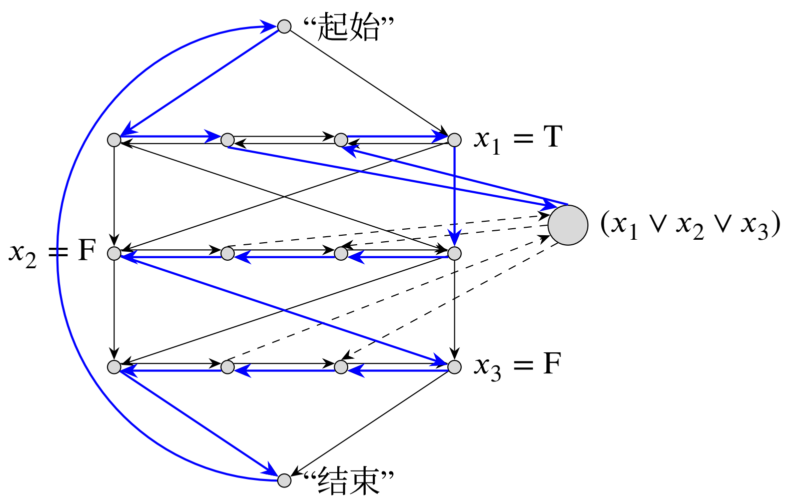
中的文字, 若 x_i 为真, 就把 $x_i = \mathrm{T}$ 和文字 x_i 的顶点连接, 否则与 $\neg x_i$ 连接.



这时要让所有的边被覆盖, 子句中所有的文字都应该为真, 还不满足要求. 设想赋值为 T, T, T, 则选择的点用红色给出, 覆盖的边用蓝色给出. 我们希望再选上 $\neg x_2, \neg x_3$, 由此可以发现应该添加一个三角形, 此时这刚好是一个最小点覆盖 (因为上方至少要选一个, 下方至少要选两个). 反过来, 最小点覆盖一定保证有一个文字是真的, 否则整个子句三角形的三个顶点都要选上. 因此构造这样的图之后, 只要询问是否存在大小不超过 $(2 \times$ 子句个数 $+$ 变元个数$)$ 的点覆盖即可. ◇

注 3 其实上面这构造并没有之后的几个例子那么自然. 但是把 3SAT 归约到独立集容易得多, 可以思考一下怎么办. (每个子句一个三角形, 选出一个成真文字即可, x 和 $\neg x$ 相互连边表示不能同时选, 独立集参数等于子句个数.) 总之, 顶点覆盖、独立集和团都是 NP-完全的. ♤

例 4 (3SAT 到 (有向) Hamilton 回路) 直观如下: 注意 Hamilton 回路是问 "有或无" 的问题, 因此可以设想构造出所有变元的全部赋值方案作为回路, 这些赋值都有可能作为回路的选择, 而真/假的选择可以作为 "岔道口", 即回路上的方向. 这一点不难想到. 但是怎样描述子句的可满足性呢? 从一个子句出发, 先看一看 $(x_1 \vee x_2 \vee x_3)$.



我们要强迫 Hamilton 回路在这三条路上至少有一个方向正确, 也就是说上图中三条路至少有一条从左到右. 由此只需要加入一个对应于子句的顶点, 要访问这个顶点就必

须在某条路的方向是正确的——如上图所示, 增加一个想要的方向的经过新顶点的通路即可. 图中蓝色的回路表示的是赋值为 T, F, F 的情况.

这样一来, 对于多个子句的情况, 只需要在单行道中间再添加一对一对的顶点和子句顶点相连来逼迫方向正确就可以了, 这就是归约的想法. ◇

注 5 Hamilton 通路也可以用一样的方法归约, 去掉终点到起点的回环就可以. 另外 Hamilton 回路也可以直接归约到 Hamilton 通路, 只需要任取一个顶点 v 将其分裂为 v_{in} 和 v_{out}. 将无向 Hamilton 回路归约到 Hamilton 回路的办法也是一样的. ♤

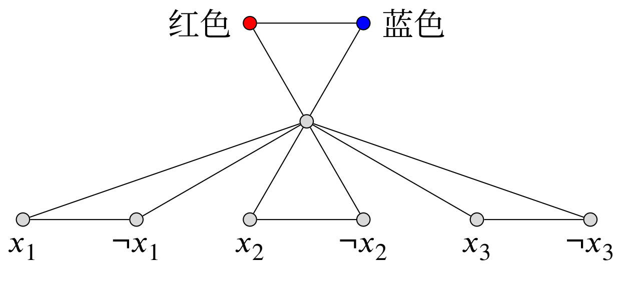
例 6 (3SAT 到恰好覆盖) 此问题想起来比 Hamilton 回路直接得多, 结合 3SAT 的第一种选择的理解很快可以构造.

子句应选择为真的文字. 对于子句 C_i, 我们制造三个集合 (无序对) $\{C_i, z_{ij}\}$ $(1 \leq j \leq 3)$, 表示子句 C_i 的第 j 个文字 z_{ij} 是取真值的. 可满足性一定要选出其中的一个. 但是, 取假的文字对应的 z_{ij} 就未能选进来, 为此我们添加集合 $F_i = \{x_i\} \cup \{x_i$ 无否定的文字$\}$, $T_i = \{x_i\} \cup \{x_i$ 带否定的文字$\}$. 当 x_i 取假时, 一切 F_i 中的文字都要通过 F_i 来补偿, 否则通过 T_i 来补偿. 注意引入 x_i 作为元素是因为这两个只能选一个.

最后, 由于恰好覆盖的要求, 带 C_i 的集合只能选一个. 举例来说, 满足 $(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2)$ 的赋值 T, T, F 中, 应该选出 $\{C_1, z_{11}\}, \{C_1, z_{13}\}, \{C_2, z_{22}\}$, 但 z_{13} (或者 z_{11}) 不能选进来, 而因为 $z_{13} = \neg x_3$ 取真, 所以 T_3, F_3 都无法补偿这个元素, 因此最后还要引入所有的文字 $\{z_{ij}\}$ 集合来填充. 这就完成了归约. ◇

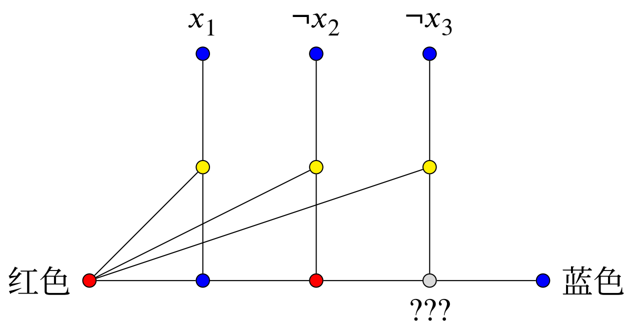
例 7 (图的 3-染色) 3-染色问题: 判定一个给定的无向图是否能够用不多于三种颜色合法染色. (显然 2-染色问题可以多项式时间内求解.)

相邻顶点的颜色不能相同是一个非常好的结构. 因为有数字 3, 所以把 3SAT 归约到它. 对于每个变元 x_i, 我们给出 $x_i, \neg x_i$ 两个顶点并连接一条边, 这样二者的颜色不能相同, 代表了对变元的赋值. 为了迫使所有的变元顶点都只染两种颜色, 所以我们可以构造一些共顶点的三角形:



给定中心点的颜色之后, $x_i, \neg x_i$ 的颜色只能是剩下两种而且不同. 不妨设这两种颜色是红色和蓝色 (中心结点为黄色), 而且红色表示取 x_i 为真. 现在要迫使这些顶点的颜色能够满足子句, 仍然从例子 $(x_1 \vee \neg x_2 \vee \neg x_3)$ 入手, 就是希望 x_1 为红、x_2 为蓝、x_3 为蓝中至少有一个成立. 这样看来, 我们必须再添加一些辅助的结点和边 (注意颜色由最上面两个点决定, 这两个点一定要用到), 使得 x_1 为蓝、$\neg x_2$ 为蓝、$\neg x_3$ 为蓝同时成立不可能.

通过一些试验我们可以发现如下图的结构是可以让这时出现矛盾的. 反过来, 简单枚举有一个、两个和三个为红的情况可发现都可 3-染色.



这样我们只需要对每个子句中希望为真的文字, 把上面的子图 “粘贴” 上去就可以了 (其中红色、蓝色和代表文字的顶点要覆盖上去). 这就完成了归约. $\diamond$

例 8 (最大割) 最大割: 给一张边权为整数的带权图, 确定是否有权值至少为 k 的割.

在网络流部分我们已经知道图中求割是一个划分问题, 因此先考虑如下问题. 划分: 给定集合 $S = \{a_1, \ldots, a_n\} \subseteq \mathbb{Z}_{\geq 1}$, 问是否存在 $S' \subseteq S$ 使得 $\sum_{x\in S'} x = \sum_{x\notin S'} x$? 我们把子集和问题归约到划分. 设子集和问题的输入是 $T = \{b_1, \ldots, b_m\}$ 和正整数 K. 子集和想要求一些元素 $b_{i_1} + \cdots + b_{i_k} = K$.

模仿课本中双机调度的技巧, 需要添加一些元素. 在这个基础上要找一个相等关系. 剩下集合的元素的和是 $\sum_i b_i - K$. 为了补齐这个差, 我们可以考虑添加一个元素 $\sum_i b_i - 2K$, 但这个数不能保证是正整数. 所以我们可以改为添加一个 $\sum_i b_i$ 和一个 $2K$. 这就完成了归约. 即 SubsetSum(S, K) $\iff$ Partition$(S \cup \{\mathrm{sum}\, S, 2K\})$. 事实上, 若子集和有解 T, 则 $T \cup \{\mathrm{sum}\, S\}$ 就是划分问题的解; 若划分有解 S', 则其中不能同时有 $\mathrm{sum}\, S$ 和 $2K$ (已经超过一半), 只需去掉其中的 $\mathrm{sum}\, S$ 或 $2K$ 就得到子集和的解.

最后把划分归约到最大割. 对于 $S = \{a_1, \ldots, a_n\} \subseteq \mathbb{Z}_{\geq 1}$, 我们构造完全图 K_n, 边 $\{i, j\}$ 的权是 $a_i a_j$, 最大割的参数为 $\frac{1}{4}\left(\sum_{i=1}^n a_i\right)^2$ (注意这一定是整数, 否则可以直接回答无可行的划分). 割 $(X, \overline{X})$ 的权是

$$\sum_{i\in X, j\in\overline{X}} a_i a_j = \sum_{i\in X} a_i \sum_{j\in\overline{X}} a_j \leq \frac{1}{4}\left(\sum_{i=1}^n a_i\right)^2,$$

等号成立当且仅当 $\sum_{i\in X} a_i = \sum_{j\in\overline{X}} a_j$, 这就完成了归约. $\diamond$

2 NP-完全性的补充例题

进行证明的基础是掌握各种基本问题及其结构, 寻找的简单技巧前面一节开头已经讲过. 我们目前已知的有 (用蓝色标出的 (在本课程中) 非常常用): (三元) 可满足性、顶

点覆盖 + 团 + 独立集 + 支配集[1]、Hamilton 回路 + 通路、旅行商问题、恰好覆盖、子集覆盖、子集和、装箱、双机调度、0-1 背包、划分、3-染色、最大割、整数线性规划.

下面我们给出归约主体, 只对不太明显的给出等价性证明. 要注意归约方向的问题, 如果思考方向反了, 不仅可能做错, 还有可能百思不得其解——待证明的问题经常比基本问题难很多, 只需要构造基本问题的一些很直观、很简单的特例就可以了 (不要想着用已知的问题解决待证明的问题). 应考时对于证明 NP-完全的题目, 不要忘记验证在 NP 中.

问题 9 SubsetPacking: 某商店有 n 位顾客和 m 种商品. 店主确定了之前他们购买商品的记录, 希望找出 k 个顾客的集合, 它们两两没有买过相同的东西. 证明该问题是 NP-完全的.

证明 选出一些结构 + 结构之间不能有关系 $\Longrightarrow$ 独立集.

归约: 给定图 $G(V,E)$, 我们构造 $|V|$ 个顾客, 商品为 $\{1,2,\dots,|E|\}$. 若顶点 v 和 e 邻接, 则让顾客 v 购买商品 e. □

注 10 类似问题: 某系统中有 m 种资源和 n 个进程, 每个资源只能由一个进程独占使用. 现在每个进程都请求这 m 种资源的若干种, 希望确定是否能合理分配资源, 使得 k 个进程都得到了请求的全部资源. 证明该问题是 NP-完全的. ♤

问题 11 组合拍卖: 临近毕业, 某同学有一个物品的集合 $S=\{1,2,\dots,n\}$ 要出售 (物品不可分割, 同一件物品不能同时卖给两个及以上的同学), 有 m 个同学有意购买这些物品中的一个或多个, 第 i 个同学希望得到物品集合 $S_i \subseteq S$, 同时出价 b_i. (注意: 有意购买这些物品的同学的需求要么被满足, 要么被完全拒绝, 不能部分满足, 即是同学 i 或者支付 b_i 得到 S_i, 或者支付 0, 得到 $\varnothing$.) 问: 该同学能否合理分配这些物品, 使得最后回收的钱数至少为 B? 证明该问题是 NP-完全的.

证明 选出一些结构 + 结构之间不能有关系 + 最大化 $\Longrightarrow$ 独立集.

归约: 给定图 $G(V,E)$, 我们构造 $|V|$ 个同学和 $|E|$ 个物品, 同学 v 想要和它邻接的全部边, 出价为 1. □

问题 12 最大公共子图: 给定两张图 $G_1(V_1,E_1), G_2(V_2,E_2)$ 和非负整数 b, 问: 是否存在两个结点集合 $V_1' \subseteq V_1, V_2' \subseteq V_2$, 使得分别去掉这些顶点之后, 两个图分别留下至少 b 个结点, 而且剩余部分同构? 证明该问题是 NP-完全的.

证明 去掉一些顶点 + 留下一些顶点 $\Longrightarrow$ 独立集.

归约: 给定图 $G(V,E)$, 取 $G_1=G=(V,E), G_2=(V,\varnothing)$. 若新问题存在大小为 b 的公共子图, 则在相应解中 G_1 所剩的顶点之间两两无边相连, 这就是大小为 b 的独立集. 若存在 b 大小的独立集, 则删掉该集合中所有其他顶点和相应的边后, 得到的图没有边, 所以和 G_2 的相应子图同构. 这就完成了归约. □

[1]顶点覆盖可归约到支配集, 请大家自己证明 (课本习题 9.19).

问题 13 某学校有 n 种体育活动项目, 现在要招聘体育老师. m 位应聘者各有能够教学的体育活动的集合 (例如 A 能教棒球、足球, B 能教足球、篮球, 等等). 学校希望确定是否能选出 k 个应聘者, 使得之后 n 种体育活动项目都有人教. 证明该问题是 NP-完全的.

证明 选出一些结构 + 结构囊括某些内容 $\Longrightarrow$ 顶点覆盖, "能教" $\approx$ 图中关系.

归约: 给定图 $G(V,E)$, 我们构造 $|V|$ 个应聘者, 体育活动为 $\{1,2,\ldots,|E|\}$. 若顶点 v 和 e 邻接, 则应聘者 v 可以教体育活动 e. □

问题 14 HittingSet: 给定 $A=\{a_1,\ldots,a_n\}$ 和子集族 $B_1,\ldots,B_m \subseteq A$. 问: 是否存在一个 $H \subseteq A$, 满足 $|H| \le k$ 而且对每个 $1 \le i \le m$ 都有 $H \cap B_i \ne \varnothing$? 证明该问题是 NP-完全的.

证明 选出 A 中一些元素 + 元素囊括 $(B_i)_{i=1}^m \Longrightarrow$ 顶点覆盖, "边" $\approx$ 集合.

归约: 给定图 $G(V,E)$, 构造 $A=V$, 子集族的全体就是边的全体 (注意边就是两个元素的集合). □

问题 15 称一个 CNF 是 monotone 的, 如果其中的文字不包含任何否定. 显然任何 monotone CNF 都是可满足的, 因为可以把所有的变元都取为真.

MonotoneSATwithFewTrueVariables: 给出一个 monotone CNF 和正整数 k, 问是否存在一个成真赋值, 使得不超过 k 个变元取值为真. 证明该问题是 NP-完全的.

证明 选出一些变元成真 + 每个子句都要有被选择的变元 $\Longrightarrow$ 顶点覆盖 $\Longrightarrow$ 逻辑表达式: 每个边都要选一个顶点.

归约: 给定图 $G(V,E)$, 对于每条边 $\{v_i,v_j\} \in E$, 构造子句 $(v_i \vee v_j)$. □

注 16 类似问题: 给一个有向图和其中的有向路径 $P_1,\ldots,P_m$, 问是否能够选择不超过 k 个图中顶点, 使得 $P_1,\ldots,P_m$ 中各有至少一个顶点被选中. (直接限制: 无向图任意定向, 路径为全部的边集.) ♤

问题 17 某人希望在 m 个地点 (全体记为 L) 中的某一些监测 n 个频率 $1,2,\ldots,n$ 的通信信号. 不过存在 b 个干扰信号, 每个干扰信号 i 由 $(F_i \subseteq [n], L_i \subseteq L)$ 表示, 意指干扰信号 i 会在一切 L_i 中地点干扰全部的 F_i 中的通信信号. 问: 是否可以选出一些地点 $L' \subseteq L$, 使得每个信号至少能在一个 L' 中地点不被干扰地监测到, 而且 $|L'| \le k$? 证明该问题是 NP-完全的.

证明 选出一些地点 + 地点囊括信号 $\Longrightarrow$ 顶点覆盖, 边 $\approx$ 能否监测到.

归约: 给定图 $G(V,E)$, 我们构造 $|V|$ 个地点, 通信信号为 E. 对于每条边 $\{v_i,v_j\}$, 添加一个干扰信号, 使得它在除了 v_i,v_j 两个地点之外, 干扰全部的边 $\{v_i,v_j\}$ 代表的信号. □

问题 18 某星球上的国际组织有 n 位成员国 $c_1,\ldots,c_n$, 它们经常开会投票来提出决议. 投票包括赞成、反对和弃权三种. 只有当赞成票严格地多于反对票时, 提案才通过. 现在给出了过去 t 次表决的情况, 包括每位成员每次的投票情况和最后的投票结果. 称一个

集合 $M' \subseteq \{c_1, \dots, c_n\}$ 是**关键大国**, 如果只考虑这些国家的投票, 每次的投票结果都不会改变. 问: 是否存在一个国家数不超过 k 的关键大国的集合? 证明该问题是 NP-完全的.

证明 选出一些国家 + 囊括所有表决结果 $\Longrightarrow$ 顶点覆盖, 边 $\approx$ 提案.

归约: 给定图 $G(V, E)$, 我们构造 $|V|$ 个国家, 过去的提案为 E. 对于每条边 $\{v_i, v_j\}$, 让 v_i, v_j 在这个提案中都投赞成票, 其他国家都投弃权. □

问题 19 最近流行的重开大学游戏可以视为一个有向图, 玩家有一个初始点 s 和一个结束点 t, 以及重大事件结点集 $T_1, \dots, T_k$. 这些结点是重大事件发生的顶点, 例如 T_1 中的顶点代表算法设计与分析考试的五种分数, T_2 中是保研夏令营七种结果, 等等. 某人沉迷于 “重开大学游戏”, 他/她希望找到一条 s, t 路径, 使得在路上能够体验所有可能的重大事件 (只需要经过每个 T_i 的顶点中的一个就可以). 证明判定这是否可能是 NP-完全的.

证明 序列 $\overset{?}{\Longrightarrow}$ Hamilton 回路/通路? No. 选择一些元素 + 囊括入某个 T_i $\Longrightarrow$ 顶点覆盖 $\Longrightarrow$ HittingSet (例题 14).

归约: 对于 HittingSet 问题实例 $A = \{a_1, \dots, a_n\}, B_1, \dots, B_m$ 和大小限制 k, 我们构造有 $k+2$ 层的重开大学游戏. 第一层是 s, 最后一层是 t, 中间每一层都有 n 个顶点 $(v_{ij})_{j=1}^n$, 表示对 Hitting set 的元素的选择. 这些层构成 “全连接神经网络”. 而对于每个集合 S_ℓ, 我们令重大事件 $T_\ell = \{v_{ij} : a_j \in S_\ell, 1 \le i \le k\}$. □

问题 20 给定定义在 $[0, t]$ 上的 n 个分段线性的连续函数 $f_1, \dots, f_n$ 和实数 B, 问: 是否可以选择 k 个函数 $f_{i_1}, \dots, f_{i_k}$, 使得

$$\int_0^t \max\{f_{i_1}(x), \dots, f_{i_k}(x)\}\, \mathrm{d}x \ge B?$$

证明该问题是 NP-完全的.

证明 首先这个问题在 NP 中不是特别显然. 证明如下: 因为线性函数的积分可以显式 $O(1)$ 计算出来, 而一个解函数 $\max\{f_{i_1}(x), \dots, f_{i_k}(x)\}$ 的分段点不超过 k^2q^2 个, 其中 q 是所有备选函数里分段数最多的, 因此这确实是 NP 中的问题.

选择一些结构 + 取 max $\approx$ 至少某个结构能满足要求 $\Longrightarrow$ 顶点覆盖.

归约: 我们将顶点覆盖的实例 $G = (V, E)$ 归约到该问题. 其中 $V = [n], E = \{e_0, \dots, e_{m-1}\}$. 对每个顶点 i, 构造函数 $f_i : [0, 2m-1] \longrightarrow [0, 1]$ 如下. 对每个条边 e_j $(0 \le j \le m-1)$, 如果 v_i 和 e_j 邻接, 就定义 $[2j, 2j+1]$ 上的函数值是 1, 否则是 0. 另取 $f_i(2j+1.5) = 0.5$, 其他各位置的函数值通过直接连线来获得. 因此 f_i 的图线类似于图 1 左侧.

这里主要的部分是红色的代表选择覆盖边的操作, 而蓝色的部分是为了保证连续加入的. 可以看出, 如果选入的函数成为顶点覆盖, 那么最后的函数 $\max\{f_{i_1}(x), \dots, f_{i_k}(x)\}$ 应该形如图 1 右侧, 其下的面积为 $m + \frac{3}{4}(m-1)$, 所以这就是 B, 这就完成了归约. □

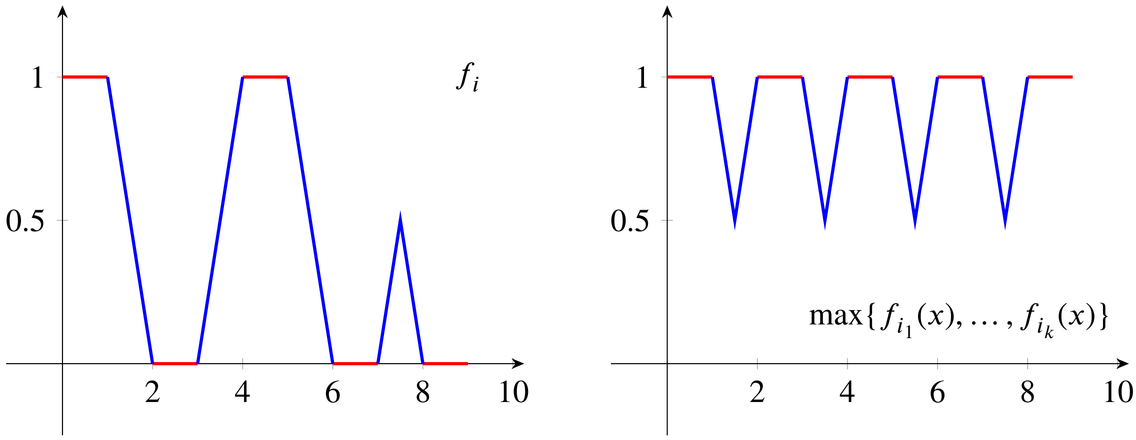


图 1: 假设 v_i 和边 e_0, e_2 相连, 和 e_1, e_3, e_4 不相连

问题 21 FeedbackSet: 给定图 $G(V, E)$, 问: 是否存在一个 $X \subseteq V$ 使得 $|X| \leq k$ 而且 $G \setminus X$ 是无圈的? (这样的集合称为 feedback set, 一个容易想到的应用是操作系统的死锁检测和消除.) 证明该问题是 NP-完全的. 提示: X 满足条件等价于每个 G 中的圈都包含至少一个 X 中顶点.

证明 每个圈中都要选一个顶点 $\Longrightarrow$ "覆盖" 每个圈 $\Longrightarrow$ 顶点覆盖. 注意顶点覆盖一定是一个 feedback set (因为实际上删掉它之后图中就没有边了), 因此工作是迫使 feedback set 必须是一个顶点覆盖.

归约: 给定 $G(V, E)$, 构造 FeedbackSet 的实例为, 在每个 $e \in E$ 上生长一个三角形——即对于 $\{u, v\} \in E$, 添加一个 w 并连接 uw, vw, 设新图是 G'. 对于 G' 中的一个大小为 k 的 feedback set, 它必须在每条边生长的三角形中至少选择一个. 对于每条边 $\{u, v\}$ 对应的三角形 uvw, 如果这个 feedback set 选择了 w, 我们就把它替换成 v. 因为每个三角形都对应一条边, 所以替换后得到的一定是 G 的顶点覆盖而且大小不超过 k. 反过来是简单的, 对于每个 G 的顶点覆盖, 它原封不动地就是 G' 的 feedback set, 因为对于 G 中原来的边删掉它们会导致这些边全部丢失, 而同时会破掉每个 uvw 圈. □

注 22 类似问题: DirectedFeedbackSet, 即把图和圈都改成有向的, 这时归约不用加顶点, 在每条边上都补反向边就可以. ♤

问题 23 考虑如下有向图 $D(V, E)$ 中的传染病模型. 每个人 $v \in V$ 都有一个 "免疫力水平" $\theta(v) \in [0, 1]$. 若边指向 v 的 v 的邻居 $\mathcal{N}^-(v)$ 里有不少于 $\theta(v)$ 比例的顶点都得了传染病, 则 v 也会得病. 我们把 $\theta(v) = 0$ 的顶点视为传染源, 初始时立刻感染. 规定对于 $d^-(v) = 0$ 的顶点, 除非 $\theta(v) = 0$, 否则他/她永远不会感染. 显然, 我们可以在 $O(|V|)$ 时间内模拟计算出最后得病的人数.

对人群 $S \subseteq V$, 规定把 S 中所有的人的免疫力水平都设为零, 这样得到的最后得病人数称为 S 的**传染力** $f(S)$. 问: 是否存在一个人群 S 使得 $|S| \leq k$ 而且 $f(S) \geq b$? 证明该问题是 NP-完全的.

证明 感染 $\approx$ 邻居中要选择一些顶点 $\Longrightarrow$ 顶点覆盖.

归约: 给定图 $G(V,E)$, 新图中的顶点为 $V \cup E$. 对每条边 $e = \{v_i, v_j\}$ 构造一个小团体 $(v_i, e), (v_j, e)$, 所有 e 的免疫力水平都是 0.5 (传染 $\approx$ 覆盖 $\approx$ 选一半顶点). 那么 G 中一个大小为 k 顶点覆盖等价于新图中能传染 $k + |E|$ 个人的 S. $\Longrightarrow$ 是平凡的. 若以集合 S 开始能感染 $k + |E|$ 个人, 我们可以把 S 中那些选择 E 代表的顶点都换成一个和它相邻的 V 中顶点, 这样初始集合大小不变, 仍能保证最后感染 $k + |E|$ 个人, 修改后的集合只在 V 中取, 根据构造, 它要感染 $k + |E|$ 个人就必须是大小为 k 的顶点覆盖. □

问题 24 设 (X, Y) 是二部图. 称边集 $E' \subseteq E$ 是 (a,b)-骨架, 如果最多 a 个 X 中顶点和 E' 中边邻接, 而且最少 b 个 Y 中顶点和 E' 中边邻接. 给定二部图和 a, b, 证明判定是否存在 (a,b)-骨架是 NP-完全的.

证明 选择一些结构 + 覆盖一些结构 $\overset{?}{\Longrightarrow}$ 顶点覆盖? No, 不易回到二部图上, 应该选择有两类结构的覆盖问题 $\Longrightarrow$ 子集覆盖.

归约: 给定 $S_1, \dots, S_m \subseteq U$, 其中 $|U| = n$, 我们构造 $X = \{S_1, \dots, S_m\}$, Y 为 U 中所有元素. 从 S_i 顶点到其中的所有元素顶点连边. 取 $a = k, b = n$, 注意 $b = n$ 表示选出的边必须构成集合覆盖, 这就完成了归约. □

问题 25 给定有向图 $D(V,E)$ 和权函数 $w\colon E \longrightarrow \mathbb{R}$, 问: 图中是否存在一个权和为 0 的有向圈? 证明该问题是 NP-完全的.

证明 求和且相等 $\Longrightarrow$ 子集和.

归约: 对于子集和问题 $\{b_1, \dots, b_n\}$ 和 K, 构造 $n+1$ 个顶点 $0, 1, \dots, n$. 对于每个 $i < j$ 都连接一条权为 b_j 的边, n 到 0 有一条权 $-K$ 的边. 注意这张图上的所有圈都包含 $(n, 0)$ 这条边就完成了归约. □

问题 26 以物易物问题: 一群人 $p_1, \dots, p_n$ 有一些物品 $\mathscr{A} = \{a_1, \dots, a_m\}$. 每件物品恰好为一个人所有, 假设 p_i 拥有的物品集合是 A_i. 每个人 p_i 都有一个价值函数 $v_i\colon \mathscr{A} \longrightarrow \mathbb{Z}_{\geq 1}$. 我们称两个人 p_i, p_j 是**有机会进行交易**的, 如果存在 $A_i' \subseteq A_i, A_j' \subseteq A_j$ 使得

$$\sum_{a \in A_i'} v_j(a) > \sum_{a \in A_j'} v_j(a) \quad \text{而且} \quad \sum_{a \in A_j'} v_i(a) > \sum_{a \in A_i'} v_i(a),$$

即互相**严格地**更想要对方手中的**一部分**东西. 证明判定是否存在两个有机会交易的人是 NP-完全的. (这说明探测严格的 Pareto 优化是一般的困难的事情.)

证明 求和 $\Longrightarrow$ 子集和 $\Longrightarrow$ 两个人应该就够了, 比大小 $\Longrightarrow$ 添加一个物品作为求和结果的代表 + 夹逼.

归约: 对于子集和问题 $\{b_1, \dots, b_n\}$ 和 K, 让一个人拥有 $b_1, \dots, b_n$, 另一个人拥有代表求和结果的 b_{n+1}. 两人对 $b_1, \dots, b_n$ 的价值判断都是相应的子集元素的值. 对于前一个人, $v_1(b_{n+1}) = K + 1$, 而后一个人的价值 $v_2(b_{n+1}) = K - 1$. 这个设置是为了不等式夹逼

子集和. 注意有机会交易, 等价于

$$\sum_{i_1,\ldots,i_k} b_{i_k} < K+1 \quad \text{而且} \quad \sum_{i_1,\ldots,i_k} b_{i_k} > K-1,$$

这正对应了子集和的结果. □

问题 27 DoubleShortestPath: 给一张带权图 $G(V,E)$, 每条边 e 都有一个长度 $\ell_e \geq 0$ 和经过这条边的风险 $r_e \geq 0$. 给定 $s,t \in E$, 问: 是否存在一条 s,t 路径使得路径的长度和不超过 L, 而且风险和不超过 R? 证明该问题是 NP-完全的.

证明 求和 $\Longrightarrow$ 子集和, 路径 $\approx$ 选择. 注意子集和的等号等价于两个计数器的 $\leq$ (选和不选的都有上界).

归约: 对于子集和问题 $\{b_1,\ldots,b_n\}$ 和 K, 构造一个有 $n+1$ 个结点的路径选择的问题, 路径的选择就是对元素的选择. 最左侧的点是 s, 最右侧的点是 t, 中间的每对路径中, 一条 $(\ell,r)=(b_i,0)$, 表示选择了 b_i, 另一条则为 $(0,b_i)$, 表示不选择 b_i (例如下图).

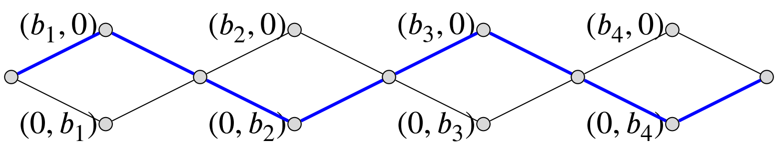


图 2: 蓝色路径表示子集 $\{b_1,b_3\}$

取 $L=K, R=\sum_i b_i - K$, 则存在想要的路径等价于有某个 $S' \subseteq \{b_1,\ldots,b_n\}$ 使得

$$\sum_{x\in S'} x \leq K \quad \text{而且} \quad \sum_{x\notin S'} x \leq \sum_i b_i - K,$$

这正对应了子集和的结果. □

问题 28 装箱问题的限制与推广: 有 n 个装有危险品的集装箱需要装到 m 辆卡车中, 每辆卡车最多承载 k 个箱子. 请判定以下问题是否是 NP-完全的.

(1) 若有一些集装箱不能放到特定的卡车中, 问: 是否能将这些集装箱一次运走?

(2) 若集装箱可以放到任何卡车中, 但是有一些不能放在同一辆卡车中中, 问: 是否能将这些集装箱一次运走?

解 (1) 是很简单的网络流问题 (匹配 + 限制), 并非 NP-完全.

(2) 冲突 $\Longrightarrow$ 3SAT 或者 3-染色. 不难看出 3-染色更容易, 因为这甚至不用归约, 只需要限制一下就可以了 (3 辆容量无限的卡车). □

问题 29 聚类问题: 设 $p_1,\ldots,p_n$ 为一族点, 其上有 (伪) 距离函数 $d(\cdot,\cdot)$, 满足非负性和对称性, 但不一定满足三角不等式. 现在希望把它们聚为 k 类, 使得每一类中的点之间的最小距离都不超过 B. 证明判定这是否可能是 NP-完全的.

证明 聚类 $\Longrightarrow$ (3-) 染色 $\Longrightarrow$ 有边相连 $\approx$ 不要聚在一起.

归约: 给定图 $G(V,E)$, 我们规定顶点为待聚类的点, 距离函数为

$$d(v_i,v_j)=\begin{cases}0 & (i=j),\\ 1 & (\{v_i,v_j\}\notin E),\\ 114514 & (\{v_i,v_j\}\in E).\end{cases}$$

并设定 $B=1,k=3$. 这样只要有满足条件的聚类, 那些有边相连的都不会聚为一类, 由此完成归约. □

问题 30 (某年期末考试题) 给定正整数 $x_1,\dots,x_n$ 和 k,B, 问: 是否可能把这些整数划分为集合 $S_1,\dots,S_k$, 使得

$$\sum_{i=1}^{k}\left(\sum_{x\in S_i}x\right)^2\le B?$$

证明该问题是 NP-完全的.

证明 划分 $\Longrightarrow$ Partition 问题 (例题 8 的媒介问题).

归约: 给定 $x_1,\dots,x_n$, 设 $S=\sum_i x_i$, 划分是要找到满足 $\frac{1}{2}S$ 可能的解, 故可取 $B=2\cdot(S/2)^2=\frac{1}{2}S^2,k=2$. 实际上, 注意到任何的划分得到的子集和的平方和都不低于 $\frac{1}{2}S^2$ 就完成了归约. □

问题 31 (某年期末考试题) 强独立集: 称图 $G(V,E)$ 的一个顶点子集 I 是**强独立集**, 如果对任何 $u,v\in I$ 都有 $d(u,v)>2$, 也就是 u,v 之间不存在长度为 1 或者 2 的路径. 证明判定图中是否存在大小至少为 k 的强独立集的问题是 NP-完全的.

证明 结构完全相似 $\Longrightarrow$ 独立集, 把原独立集的顶点之间的距离增大 $\Longrightarrow$ 拆边增点 $\Longrightarrow$ 阻止新点被选择.

归约: 给定 $G(V,E)$, 我们在每个 e 上细分一个顶点 u_e, 并把全部的 u_e 两两相连, 得到的新图称为 G'. 如果 G 中有大小为 k 的独立集, 那它当然是 G' 中的强独立集. 若 G' 中有强独立集, 这个集合不能选任何 u_e 顶点 (因为这些顶点都和其他顶点距离不超过 2), 所以强独立集中的点都来源于 G, 又根据构造这些点在 G 中都不可能相邻, 这就完成了归约. □

问题 32 以下是一个和染色体分析有关的问题. 设有以 $\Sigma\,(|\Sigma|=q)$ 为字母表的 2ℓ 长度的字符串集合 $\mathscr{S}=\{s_1,\dots,s_n\}$ 和 $\mathscr{T}=\{t_1,\dots,t_m\}$. 问: 是否可能适当排列得到一个串 $s_{i_1}s_{i_2}\cdots s_{i_n}$, 满足对于每个 $1\le j\le n-1$, 都存在某个 $t_k\in\mathscr{T}$ 使得 $s_{i_j}[\ell:]s_{i_{j+1}}[:\ell]=t_k$? 即排列之后, 可以把某个 t_k 同 s_{i_j} 的后 ℓ 个字符和 $s_{i_{j+1}}$ 的前 ℓ 个字符完全对齐. 证明该问题是 NP-完全的.

证明 排列 $\Longrightarrow$ Hamilton 通路, 字符 $\approx$ 路径指示牌.

归约: 给定有向图 $D(V,E)$, 我们取 $\ell = 1$, $\mathcal{S}$ 为所有顶点 $\{v_1v_1', \dots, v_nv_n'\}$, 代表从 v_i 进入, 从 v_i' 离开. $\mathcal{T}$ 则为 $\{vw' : (v,w) \in E\}$. 这样, 对齐就是指可以通过一条边走到另一个顶点. □

问题 33 进货问题: 在接下来的 k 天中, 某食堂打算每天推出一道新菜 $S_i\,(1 \le i \le k)$. 每道菜可视为一个集合 $S_i \subseteq \{I_1, \dots, I_n\}$, 表示做这个菜需要原料 $I_1, \dots, I_n$ 中的一些, 具体地说, S_i 需要 a_{ij} 克的原料 I_j. 原料 I_j 可以从市场上买到, 单价是每克 c_j 元, 但是购买时必须买 s_j 克的整数倍 (例如没有人会只买一个鸡蛋), 它的保质期是 t_j 天.

可以看出, 因为每次都要买充分多的量, 食堂可以通过合理安排这些菜的推出顺序和购买原料的方案, 使得尽量不浪费食材. 问: 是否存在一个安排方案, 使得可以完成任务而且花费钱数不超过 x? 证明该问题是 NP-完全的.

证明 排列 $\Longrightarrow$ Hamilton 通路, 这里边就是原料的需求.

归约: 给定有向图 $D(V,E)$, 新菜的集合为 V, 原料的集合为 E. 凡是和 v 邻接的边都为 v 需要的原料. 每道菜都需要 1 克原料, 但是原料只能按 2 克的批次, 每次 2 元来购买, 保质期为 2 天. 这样, 通过边 e 邻接的两个顶点只需要购买原料 e 一次. 所以, 一个 Hamilton 通路可节约 $|V| - 1$ 元钱, 保质期保证了无法通过其他任何方法来节省开支, 故设定 $x = 2|E| - (|V| - 1)$, 这就完成了归约. □

问题 34 某操作系统的狂热粉丝同学想要为自己选择一个合适的操作系统 (比如光是 Linux 就有无数发行版), 但是升级系统带来许多连带的兼容性问题, 比如在 `Linux` 下邮件客户端不再能是 `Outlook`, 而有些游戏在 `MacOS` 下无法运行, 等等. 把这个选择问题抽象如下: 在集合 $A_1, \dots, A_n$ 中各选出一个元素, 要求这些元素不能在冲突对 P 中. 例如 $A_1 = \{$`Windows`, `MacOS`, `Linux`$\}$, $A_2 = \{$`Microsoft Office`, `Libre Office`$\}$, $P = \{($`Linux`, `Microsoft Office`$)\}$. 证明判定这是否可能是 NP-完全的.

证明 这和注 3 里给出的独立集问题是相同的, 直接从 3SAT 归约也可以. □

问题 35 在微积分中, 我们知道在一个闭集上的连续函数总有最大值和最小值. 这些最值可能在边界取到, 也可能在内点取到. 因为边界的存在, 我们直观地会怀疑如下问题是困难的: 设 $f(x_1, \dots, x_n)$ 是 $[0,1]^n \longrightarrow \mathbb{R}$ 的整系数 n 元多项式, 给定整数 B, 问: 是否存在使得 $f(\cdot) \le B$ 的点? 证明该问题确实是 NP-完全的.

证明 结构不甚清楚 $\Longrightarrow$ 3SAT. 当然如果熟悉 Bool 函数的话这题就更明显了.

归约: 对每个变元 x_i 我们都引入一个相应的变量. 每个子句对应一个乘积, 若文字为 x_i, 则加入 $1 - x_i$, 否则加入 x_i, 然后把全部的乘项相加. 比如 $(x_1 \vee \neg x_2 \vee x_3)$ 对应 $(1 - x_1)x_2(1 - x_3)$. 我们令 $B = 0$. 由于每一项都是非负的, 因此存在 ≤ 0 的点等价于每项都是零, 这就对应着每个子句至少一个文字为真的成真赋值. □

问题 36 EvasivePath: 给一张有向图 $D(V,E)$ 和顶点的子集 $Z_1, \dots, Z_k \subseteq V$ (这些集合可以有交集), 问给定的顶点 s, t 之间是否存在一条路径 P, 使得对任意的 $1 \le i \le k$, 路径中最多经过属于 Z_i 的顶点一次 (即 $|P \cap Z_i| \le 1$). 证明该问题是 NP-完全的.

证明 结构不甚清楚 $\Longrightarrow$ 3SAT, 路径选择 $\approx$ 真值赋值, 经过一次 $\approx$ 变元非真即假.

归约: 我们构造有 $n+k+2$ 层的有向网络, 这里 n 是变元个数, k 是子句个数. 第一层是 s, 最后一层是 t. 中间的前 n 层每层两个顶点, 表示 $x_i, \neg x_i$; 中间的后 k 层每层三个顶点, 其中顶点代表每个子句中的文字. 这些中间层构成 “全连接神经网络”. 对于每个文字和出现的每对 $x_i, \neg x_i$ 顶点, 我们都将这两个顶点作为一个集合加到 $Z_{(\cdot)}$ 中 (从而表示至多一个为真). 注意这样的集合最多也不超过 $\binom{nk}{2}$ 个, 仍然是多项式级别的. 这就完成了归约. □

问题 37 倒买倒卖问题: 在一张有向图 $D(V,E)$ 中, 投机商人带着 z 单位的钱从源点 s 出发, 通过路径走向 t. 市场上有 n 种商品 $1,2,\dots,n$. 在每个顶点 $p \in V$ (包括 s,t), 对物品 $j\,(1 \le j \le n)$ 都有以进价 $x(j,p)$ 提供的总量为 $s(j,p)$ 的货源, 同时有以销售价格 $y(j,p)$ 购买它的需求量 $d(j,p)$. 在每个结点商人都可以自由地买一些货物, 或者卖一些货物, 或者同时买卖. 问: 是否存在一条路径和买卖方案使得到达 t 结束后, 商人至少有 B 的钱? 证明该问题是 NP-完全的.

证明 思路和上一题完全一样.

归约: 对有 n 个变元 $(x_i)_{1\le i\le n}$ 和 k 个子句的的 3SAT 问题, 我们构造 $n+k+2$ 层的有向网络. 第一层是 s, 最后一层是 t. 这两个结点无供应也无需求. 在中间的前 n 层每层两个顶点, 一个顶点供应货物 x_i, 一个顶点供应货物 $\neg x_i$, 货量分别是它们对应的文字在所有子句中出现的次数, 价格为零. 在中间的后 k 层每层三个顶点, 分别表示每个子句中出现的文字, 每个顶点分别对相应的 “文字货物” 有 1 个单位的需求, 价格为 1. 这些中间层构成 “全连接神经网络”. 设定 $z=0, B=k$.

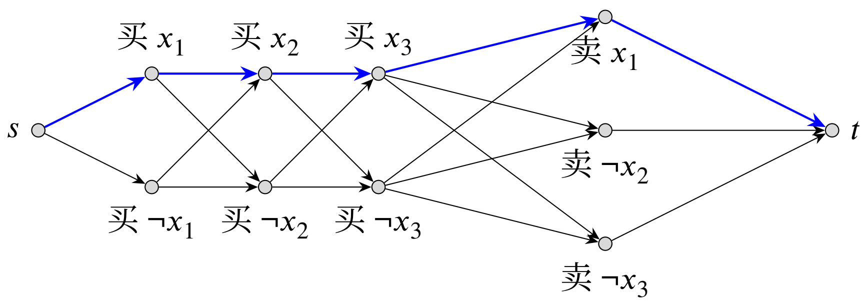


图 3: $(x_1 \vee \neg x_2 \vee \neg x_3)$, 蓝色路径示出了赋值 T, T, T

不难看出商人只能买 $x_i, \neg x_i$ 中的一个, 设定 $B=k$ 就是要求每个子句中至少有一个文字被 “卖出” (成真) 了, 这就完成了归约. □

问题 38 VertexDisjointPaths: 给定有向图 $D(V,E)$ 中的不同点对 $(s_1,t_1),\dots,(s_k,t_k)$, 问: 是否存在连接各点对的有向路径 $P_1,\dots,P_k$, 使得这些路径两两之间没有公共顶点? 证明该问题是 NP-完全的.

证明 做法和 3SAT 到 HamiltonCycle 的归约一样.

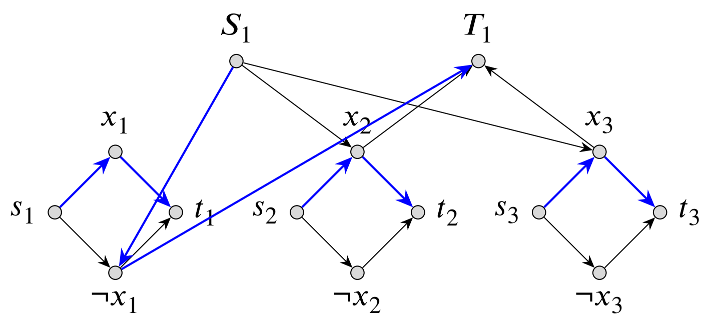


图 4: $(x_1 \vee \neg x_2 \vee \neg x_3)$ 和对应赋值 T, T, T 的顶点不交路径 (蓝色)

归约: 给定有 n 个变元 $(x_i)_{1\le i\le n}$ 和 k 个子句的的 3SAT 问题, 我们分别构造两条 (s_i, t_i) 路径表示变元的选择. 为了迫使子句都能被满足, 我们再为每个子句构造一组 (S_i, T_i), 并让它和每一个该子句不想选择的文字的路径重叠. 比如 $(x_1 \vee \neg x_2 \vee \neg x_3)$ 对应的子句应该和 $\neg x_1, x_2$ 和 x_3 的路径重叠 (参看图 4).

这样一来, 对于多个子句的情况, 只需要中间添加足够多的顶点让 (S_i, T_i) 插入, 从而逼迫问题的解至少选择一个文字为真就可以了, 这就是归约的想法.

严格地说, 对每个 x_i, 都构造两条路径长度为 k 的路径, 一条是 $s_iC_1\cdots C_kt_i$, 代表 $x_i = \mathrm{T}$, 另一条反之, 为 $s_i'C_1'\cdots C_k't_i'$. 对子句 C_j $(1\le j\le k)$, 构造 (S_j, T_j); 对每个在 C_j 中出现的文字, 若它是 x_i, 就添加路径 $S_iC_j'T_i$, 否则添加 $S_iC_jT_i$. 这就完成了归约. □

3 其他下界问题/方法

到目前为止, 人们对于解决一个问题的算法所需要的最低复杂度的处理方法是很少的. 几乎可以说用决策树得到的下界是所有传统算法问题中唯一一个非平凡的、具体的下界. 现今最广泛被使用的办法还是基于一些复杂性假设的归约, 比如如果假设 P $\neq$ NP 成立, 那么所有 NP-完全的问题都不可能有多项式时间的算法. 还有其他的很有用的复杂性假设. 其他证明下界的方法可以是基于信息的 (如通讯复杂度). 课本第 8 章中提到的证明思想最重要的是**构造对抗输入**, 在诸如机器学习 (在线算法) 的领域, 下界的证明通常是用这种办法.

参考文献

以上例子、例题和课本内容合起来就几乎覆盖了 Karp 的全部 21 个问题——除了 Steiner 树和三维匹配. 对于前者, 可以做 [KT06, 第 8 章, 第 38 题] (提示: 用顶点覆盖); 对于后者, 可以阅读 [KT06, §8.6 节] 以及做相应的习题 7~9、14、16、40~42.

对复杂性问题的严格处理感兴趣的同学可以参考 [AB09].

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. ISBN: 978-0-52-142426-4.

[Kar72] R. M. Karp. “Reducibility among combinatorial problems”. 刊于: *Complexity of Computer Computations*. Springer, 1972, pp. 85–103.

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

编写: WC
E-mail: wchang@pku.edu.cn