

算法设计与分析

第12讲 完全性

汪小林

北京大学 信息科学技术学院

主要内容

- 7.1 P类与NP类
 - 7.1.1 易解的问题与难解的问题
 - 7.1.2 判定问题
 - 7.1.3 NP类
- 7.2 多项式时间变换与NP完全性
 - 7.2.1 多项式时间变换
 - 7.2.2 NP完全性
 - 7.2.3 Cook-Levin定理——第一个NP完全问题
- 7.3 几个NP完全问题

最大可满足性与三元可满足性、顶点覆盖,团与独立集、哈密顿回路
与货郎问题、恰好覆盖、子集和,背包,装箱与双机调度

7.1 P类与NP类

7.1.1 易解的问题与难解的问题

评价算法好坏的重要标准——运行时间

快速排序算法 $O(n\log n)$

Dijkstra算法 $O(n^2)$

最大团问题的回溯法 $O(n2^n)$

用一台每秒10亿次的超大型计算机计算

快速排序算法给10万个数据排序, 运算量约为

$10^5 \times \log_2 10^5 \approx 1.7 \times 10^6$, 仅需 $1.7 \times 10^6 / 10^9 = 1.7 \times 10^{-3}$ 秒.

什么是好算法?

Dijkstra算法求解1万个顶点的图的单源最短路径问题, 运算量约为 $(10^4)^2=10^8$, 约需 $10^8/10^9=0.1$ 秒.

回溯法解100个顶点的图的最大团问题, 运算量为 $100 \times 2^{100} \approx 1.8 \times 10^{32}$, 需要 $1.8 \times 10^{32}/10^9 = 1.8 \times 10^{21}$ 秒
 $= 5.7 \times 10^{15}$ 年, 即5千7百万亿年!

再从另外一个角度来看——1分钟能解多大的问题. 1分钟60秒, 这台计算机可做 6×10^{10} 次运算, 用快速排序算法可给 2×10^9 (即, 20亿)个数据排序, 用Dijkstra算法可解 2.4×10^5 个顶点的图的单源最短路径问题. 而用回溯法一天只能解41个顶点的图的最大团问题.

算法的时间复杂度

函数 f 和 g 是**多项式相关的**: 如果存在多项式 p 和 q 使得, 对任意的 $n \in \mathbb{N}$, $f(n) \leq p(q(n))$ 和 $g(n) \leq q(f(n))$.

例如 $n \log n$ 与 n^2 , $n^2 + 2n + 5$ 与 n^{10} 都是多项式相关的,
 $\log n$ 与 n , n^5 与 2^n 不是多项式相关的.

问题 Π 的实例 I 的**规模**: I 的二进制编码的长度, 记作 $|I|$.

定义7.1 如果存在函数 $f: \mathbb{N} \rightarrow \mathbb{N}$ 使得, 对任意的规模为 n 的实例 I , 算法 A 对 I 的运算在 $f(n)$ 步内停止, 则称算法 A 的**时间复杂度**为 $f(n)$.

多项式时间算法: 以多项式为时间复杂度.

易解的问题: 有多项式时间算法.

难解的问题: 不存在多项式时间算法.

几点说明

1.当采用合理的编码时,输入的规模都是多项式相关的.

“合理的”是指在编码中不故意使用许多冗余的字符.

例如, 设实例 I 是一个无向简单图 $G=<V,E>$,

$$V=\{a,b,c,d\}, E=\{(a,b),(a,d),(b,c),(b,d),(c,d)\}.$$

若用邻接矩阵表示, 则编码

$$e_1=0101/1011/0101/1110/, \text{ 长度为20.}$$

若用关联矩阵表示, 则编码

$$e_2=11000/10110/00101/01011/, \text{ 长度为24.}$$

设 G 有 n 个顶点 m 条边, 则用邻接矩阵时 $|I|=n(n+1)$, 用关联矩阵时 $|I|=n(m+1)$. 两者是多项式相关的.

几点说明

2. 自然数应采用二 ($k \geq 2$)进制编码, 不能采用一进制编码.
 n 的二进制编码有 $\lceil \log_2(n+1) \rceil$ 位, 一进制编码有 n 位, 两者不是多项式相关的.

3. 时间复杂度常表成计算对象的某些自然参数的函数, 如图的顶点数或顶点数与边数的函数.

实例的二进制编码的长度与这些自然参数通常都是多项式相关的.

4. 关于操作指令集的两点说明. 运行时间通常是计算执行的操作指令数, 这就要求执行的指令数与实际的运行时间是多项式相关的.

几点说明

4.1 要求每一条指令的执行时间是固定的常数. 例如, 2个不超过计算机字长的二进制数的四则运算是一条合理的指令. 而2个任意长度的二进制数的四则运算不能作为合理的指令. 当超过计算机字长时, 必须进行分段处理.

4.2 算法的计算步数与采用的操作指令集有关. 这就要求任何两个“合理的”操作指令集, 其中一个指令集中的每一条指令都可以用另一个指令集中的指令模拟, 且模拟所用的指令条数不超过某个固定的常数.

- 可以规定一个基本操作指令集, 如它由位逻辑运算与、或、非组成, 然后认为任何可以用这个基本操作指令集中常数条指令实现的操作都是合理的指令, 由有限种合理的指令构成的操作指令集是合理的操作指令集.

几点说明

- 在上述约定下, 算法是否是多项式时间的与采用的编码和操作指令集无关, 从而一个问题是易解的、还是难解的也与采用的编码和操作指令集无关.

设有编码 σ_1, σ_2 和操作指令集 D_1, D_2 . 实例 I 的这两种编码的长度分别为 $|I|_1$ 和 $|I|_2$. 存在多项式 p 和 q , 使得 $|I|_1 \leq p(|I|_2)$ 和 $|I|_2 \leq q(|I|_1)$. 又存在常数 k_1 和 k_2 , 使得 $D_1(D_2)$ 中的每一条指令都可以用 $D_2(D_1)$ 中的不超过 k_1 (k_2) 条指令模拟.

设算法 A 在采用编码 σ_1 和操作指令集 D_1 时存在多项式 f 使得, 对任意的实例 I , 算法在 $f(|I|_1)$ 步内停机. 不妨设 f 是单调递增的. 若采用操作指令集 D_2 , 算法至多运行 $k_1 f(|I|_1) \leq k_1 f(p(|I|_2))$ 步. 故 A 在采用编码 σ_2 和操作指令集 D_2 时也是多项式时间的. 反之亦然.

易解的问题与难解的问题

易解的问题. 如排序、最小生成树、单源最短路径等

已证明的难解问题. 一类是不可计算的, 即根本不存在求解的算法, 如希尔伯特第十问题——丢番图方程是否有整数解. 另一类是有算法, 但至少需要指数时间, 或指数空间, 甚至更多的时间或更大的空间. 如带幂运算的正则表达式的全体性, 即任给字母表 A 上的带幂运算的正则表达式 R , 问: $\langle R \rangle = A^*$? 这个问题至少需要指数空间.

既没有找到多项式时间算法、又没能证明是难解的问题. 如哈密顿回路问题、货郎问题、背包问题等

7.1.2 判定问题

判定问题: 答案只有两个——是, 否.

判定问题 $\Pi = \langle D_\Pi, Y_\Pi \rangle$, 其中 D_Π 是实例集合, $Y_\Pi \subseteq D_\Pi$ 是所有答案为 “Yes” 的实例.

哈密顿回路(HC): 任给无向图 G , 问 G 有哈密顿回路吗?

货郎问题(TSP): 任给 n 个城市, 城市 i 与城市 j 之间的正整数距离 $d(i, j)$, $i \neq j$, $1 \leq i, j \leq n$, 以及正整数 D , 问有一条每一个城市恰好经过一次最后回到出发点且长度不超过 D 的巡回路线吗? 即, 存在 $1, 2, \dots, n$ 的排列 σ 使得

$$\sum_{i=1}^{n-1} d(\sigma(i), \sigma(i+1)) + d(\sigma(n), \sigma(1)) \leq D?$$

组合优化问题与判定问题

0-1背包: 任给 n 件物品和一个背包, 物品 i 的重量为 w_i , 价值为 v_i , $1 \leq i \leq n$, 以及背包的重量限制 B 和价值目标 K , 其中 w_i, v_i, B, K 均为正整数, 问能在背包中装入总价值不少于 K 且总重量不超过 B 的物品吗? 即, 存在子集 $T \subseteq \{1, 2, \dots, n\}$ 使得

$$\sum_{i \in T} w_i \leq B \quad \text{且} \quad \sum_{i \in T} v_i \geq K?$$

搜索问题, 组合优化问题与判定问题的对应.

如果搜索问题, 组合优化问题有多项式时间算法, 则对应的判定问题也有多项式时间算法; 通常反之亦真.

组合优化问题与判定问题

组合优化问题 Π^* 由3部分组成:

- (1) 实例集 D_{Π^*} ;
- (2) $\forall I \in D_{\Pi^*}$, 有一个有穷非空集 $S(I)$, 其元素称作 I 的可行解.
- (3) $\forall s \in S(I)$, 有一个正整数 $c(s)$, 称作 s 的值.

如果 $s^* \in S(I)$, 对所有的 $s \in S(I)$, 当 Π^* 是最小(大)化问题时,

$$c(s^*) \leq c(s) \quad (c(s^*) \geq c(s))$$

则称 s^* 是 I 的最优解, $c(s^*)$ 是 I 的最优值, 记作 $\text{OPT}(I)$.

Π^* 对应的判定问题 $\Pi = \langle D_{\Pi}, Y_{\Pi} \rangle$ 定义如下:

$D_{\Pi} = \{(I, K) \mid I \in D_{\Pi^*}, K \in \mathbb{Z}^*\}$, 其中 \mathbb{Z}^* 是非负整数集合.

当 Π^* 是最小化问题时, $Y_{\Pi} = \{(I, K) \mid \text{OPT}(I) \leq K\}$;

当 Π^* 是最大化问题时, $Y_{\Pi} = \{(I, K) \mid \text{OPT}(I) \geq K\}$.

7.1.3 NP类

定义7.2 所有多项式时间可解的判定问题组成的问题类称作**P类**.

定义7.3 设判定问题 $\Pi = \langle D, Y \rangle$, 如果存在两个输入变量的多项式时间算法 A 和多项式 p , 对每一个实例 $I \in D, I \in Y$ 当且仅当存在 $t, |t| \leq p(|I|)$, 且 A 对输入 I 和 t 输出 “Yes”, 则称 Π 是**多项式时间可验证的**, A 是 Π 的**多项式时间验证算法**, 而当 $I \in Y$ 时, 称 t 是 $I \in Y$ 的**证据**.

由所有多项式时间可验证的判定问题组成的问题类称作**NP类**.

例如, HC, TSP, 0-1背包 \in NP

非确定型多项式时间算法

非确定型多项式时间算法:

对给定的实例 I , 首先“猜想”一个 t , $|t| \leq p(|I|)$, 然后检查 t 是否是证明 $I \in Y$ 的证据, 猜想和检查可以在多项式时间内完成, 并且当且仅当 $I \in Y$ 时能够正确地猜想到一个证据 t .

定理7.1 $P \subseteq NP$.

问题: $P=NP$?

7.2 多项式时间变换与NP完全性

7.2.1 多项式时间变换

如何比较两个问题的难度？

定义7.4 设判定问题 $\Pi_1 = \langle D_1, Y_1 \rangle$, $\Pi_2 = \langle D_2, Y_2 \rangle$. 如果函数 $f: D_1 \rightarrow D_2$ 满足条件:

- (1) f 是多项式时间可计算的,
- (2) 对所有的 $I \in D_1$, $I \in Y_1 \Leftrightarrow f(I) \in Y_2$,

则称 f 是 Π_1 到 Π_2 的**多项式时间变换**.

如果存在 Π_1 到 Π_2 的多项式时间变换, 则称 Π_1 **可多项式时间变换到** Π_2 , 记作 $\Pi_1 \leq_p \Pi_2$.

例

例7.1 $\text{HC} \leq_p \text{TSP}$.

证 对HC的每一个实例 I : 无向图 $G=<V,E>$, TSP对应的实例 $f(I)$ 为: 城市集 V , 任意两个不同的城市 u 和 v 之间的距离

$$d(u,v) = \begin{cases} 1, & \text{若}(u,v) \in E, \\ 2, & \text{否则} \end{cases}$$

以及界限 $D=|V|$.

例

最小生成树: 任给连通的无向赋权图 $G=\langle V, E, W \rangle$ 以及正整数 B , 其中权 $W:E \rightarrow \mathbb{Z}^+$, 问有权不超过 B 的生成树吗?

最大生成树: 任给连通的无向赋权图 $G=\langle V, E, W \rangle$ 以及正整数 D , 其中权 $W:E \rightarrow \mathbb{Z}^+$, 问 G 有权不小于 D 的生成树吗?

例7.2 最大生成树 \leq_p 最小生成树.

证 任给最大生成树的实例 I : 连通的无向赋权图 $G=\langle V, E, W \rangle$ 和正整数 D , 最小生成树的对应实例 $f(I)$: 图 $G'=\langle V, E, W' \rangle$ 和正整数 $B=(n-1)M-D$, 其中 $n=|V|$, $M=\max\{W(e) \mid e \in E\}+1$, $W'(e)=M-W(e)$. 如果存在 G 的生成树 T , 使得 $\sum_{e \in T} W(e) \geq D$

$$\sum_{e \in T} W'(e) = (n-1)M - \sum_{e \in T} W(e) \leq (n-1)M - D = B.$$

反之亦然.

\leq_p 的性质

定理7.2 \leq_p 具有传递性. 即, 设 $\Pi_1 \leq_p \Pi_2$, $\Pi_2 \leq_p \Pi_3$, 则 $\Pi_1 \leq_p \Pi_3$.

证 设 $\Pi_i = \langle D_i, Y_i \rangle$, $i=1,2,3$, f 和 g 是 Π_1 到 Π_2 和 Π_2 到 Π_3 的多项式时间变换. 对每一个 $I \in D_1$, 令 $h(I) = g(f(I))$.

计算 f 和 g 的时间上界分别为多项式 p 和 q , 不妨设 p 和 q 是单调递增的. 计算 h 的步数不超过 $p(|I|) + q(|f(I)|)$. 输出作为合理的指令, 一步只能输出长度不超过固定值 k 的字符串, 因而 $|f(I)| \leq kp(|I|)$. 于是, $p(|I|) + q(|f(I)|) \leq p(|I|) + q(kp(|I|))$, 得证 h 是多项式时间可计算的.

又, 对每一个 $I \in D_1$,

$$I \in Y_1 \Leftrightarrow f(I) \in Y_2 \Leftrightarrow h(I) = g(f(I)) \in Y_3,$$

得证 h 是 Π_1 到 Π_3 的多项式时间变换.

\leq_p 的性质

定理7.3 设 $\Pi_1 \leq_p \Pi_2$, 则 $\Pi_2 \in \mathbf{P}$ 蕴涵 $\Pi_1 \in \mathbf{P}$.

证 设 $\Pi_1 = \langle D_1, Y_1 \rangle$, $\Pi_2 = \langle D_2, Y_2 \rangle$, f 是 Π_1 到 Π_2 的多项式时间变换, A 是计算 f 的多项式时间算法. 又设 B 是 Π_2 的多项式时间算法. 如下构造 Π_1 的算法 C : 对每一个 $I \in D_1$, 首先应用 A 得到 $f(I)$, 然后对 $f(I)$ 应用 B , C 输出“**Yes**”当且仅当 B 输出“**Yes**”.

推论7.4 设 $\Pi_1 \leq_p \Pi_2$, 则 Π_1 是难解的蕴涵 Π_2 是难解的.

由例7.2及最小生成树 $\in \mathbf{P}$, 得知最大生成树 $\in \mathbf{P}$.

由例7.1, 如果TSP $\in \mathbf{P}$, 则HC $\in \mathbf{P}$. 反过来, 如果HC是难解的, 则TSP也是难解的.

7.2.2 NP完全性

定义7.5 如果对所有的 $\Pi' \in \text{NP}$, $\Pi' \leq_p \Pi$, 则称 Π 是**NP难的**.
如果 Π 是NP难的且 $\Pi \in \text{NP}$, 则称 Π 是**NP完全的**.
NP完全问题是NP中最难的问题.

定理7.5 如果存在NP难的问题 $\Pi \in \text{P}$, 则 $\text{P} = \text{NP}$.

推论7.6 假设 $\text{P} \neq \text{NP}$, 那么, 如果 Π 是NP难的, 则 $\Pi \notin \text{P}$.

定理7.7 如果存在NP难的问题 Π' 使得 $\Pi' \leq_p \Pi$, 则 Π 是NP难的.

推论7.8 如果 $\Pi \in \text{NP}$ 并且存在NP完全问题 Π' 使得 $\Pi' \leq_p \Pi$, 则 Π 是NP完全的.

证明NP完全性的“捷径”

为了证明 Π 是NP完全的, 只需要做两件事:

(1) 证明 $\Pi \in \text{NP}$;

(2) 找到一个已知的NP完全问题 Π' , 并证明 $\Pi' \leq_p \Pi$.

7.2.3 Cook-Levin定理

合式公式是由变元, 逻辑运算符以及圆括号按照一定的规则组成的表达式. 变元和它的否定称作**文字**. 有限个文字的析取称作**简单析取式**. 有限个简单析取式的合取称作**合取范式**. 给定每一个变元的真假值称作一个赋值. 如果赋值 t 使得合式公式 F 为真, 则称 t 是 F 的**成真赋值**. 如果 F 存在成真赋值, 则称 F 是**可满足的**.

例如 $F_1 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge \neg x_2$ 是一个合取范式.

令 $t(x_1)=1, t(x_2)=0, t(x_3)=1$ 是 F_1 的成真赋值, F_1 是可满足的.

$F_2 = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge x_2 \wedge \neg x_3$ 不是可满足的.

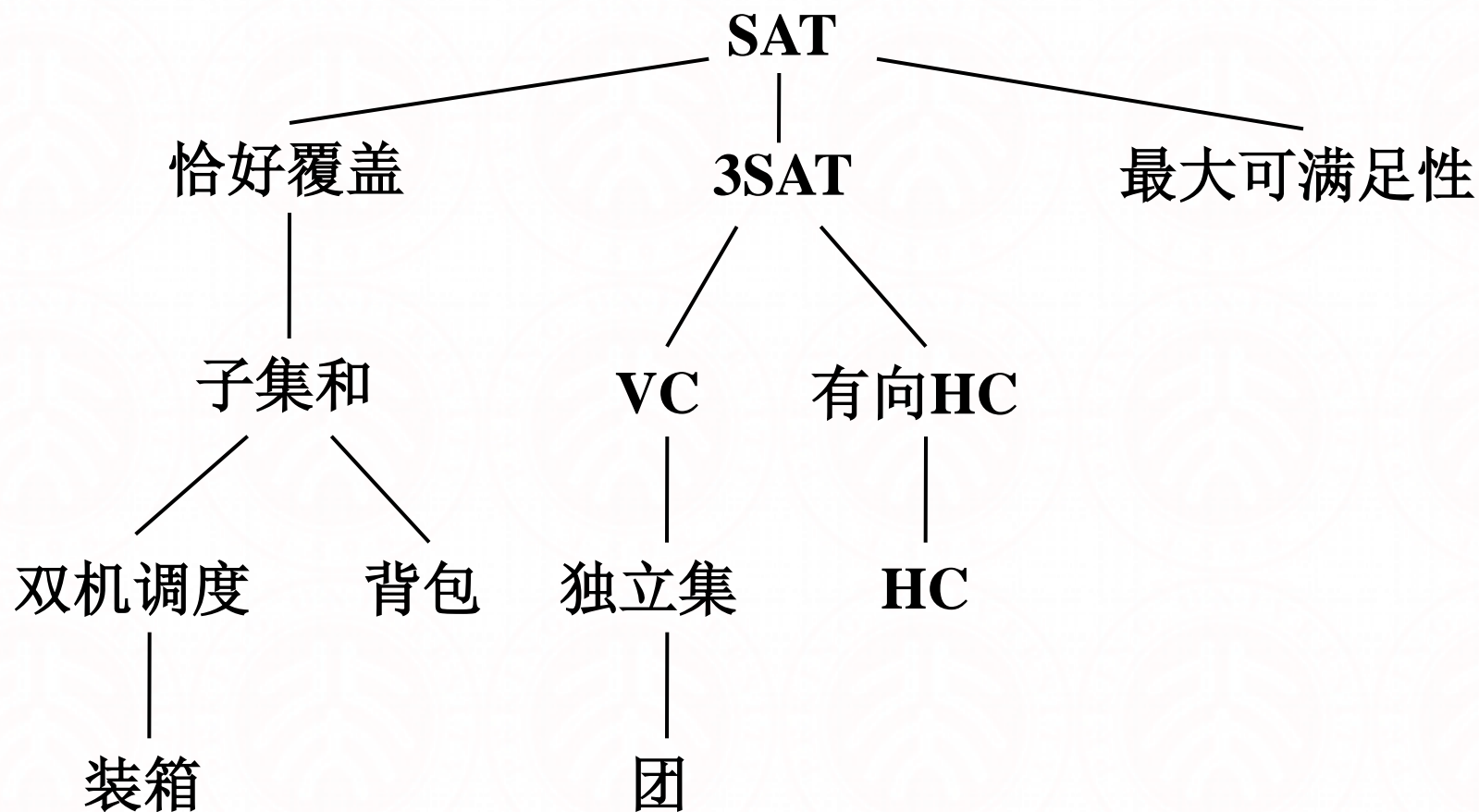
Cook-Levin定理

可满足性(SAT): 任给一个合取范式 F , 问 F 是可满足的吗?

定理7.9 (Cook-Levin定理) SAT是NP完全的.

定理7.10 $P=NP$ 的充分必要条件是存在NP完全问题 $\Pi \in P$.

7.3 几个NP完全问题



7.3.1 最大可满足性与三元可满足性

最大可满足性(MAX-SAT): 任给关于变元 x_1, x_2, \dots, x_n 的简单析取式 C_1, C_2, \dots, C_m 及正整数 K , 问存在关于变元 x_1, x_2, \dots, x_n 的赋值使得 C_1, C_2, \dots, C_m 中至少有 K 个为真吗?

设判定问题 $\Pi = \langle D, Y \rangle$, $\Pi' = \langle D', Y' \rangle$, 如果 $D' \subseteq D$, $Y' = D' \cap Y$, 则 Π' 是 Π 的特殊情况, 称作 Π 的**子问题**.

例如 “给定一个平面图 G , 问 G 是哈密顿图吗?” 是HC的子问题.
SAT是MAX-SAT的子问题: 取 $K=m$.

MAX-SAT

限制法: 如果已知 Π 的某个子问题 Π' 是NP难的, 则 Π 也是NP难的——一般情况不会比特殊情况容易. 容易把 Π' 多项式时间变换到 Π : 只需把 Π' 的实例 I 看作 Π 特殊情况的实例, 即可得到 Π 对应的实例.

定理7.11 MAX-SAT是NP完全的.

证 MAX-SAT的非确定性多项式时间算法: 猜想一个赋值, 检查是否有 K 个简单析取式满足.

要证 $\text{SAT} \leq_p \text{MAX-SAT}$. 任给SAT的实例 I : 关于变元 x_1, x_2, \dots, x_n 的合取范式 $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, 其中 C_1, C_2, \dots, C_m 是简单析取式, 对应的MAX-SAT的实例 $f(I)$: 简单析取式 C_1, C_2, \dots, C_m 和正整数 $K = m$.

3SAT

3元合取范式: 每一个简单析取式恰好有3个文字的合取范式.

三元可满足性(3SAT): 任给一个3元合取范式 F , 问 F 是可满足的吗?

定理7.12 3SAT是NP完全的.

证 显然 $3SAT \in NP$.

要证 $SAT \leq_p 3SAT$. 任给一个合取范式 F , 要构造对应的 3元合取范式 $F' = f(F)$, 使得 F 是可满足的当且仅当 F' 是可满足的.

设 $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, 对应的 $F' = F_1' \wedge F_2' \wedge \dots \wedge F_m'$, F_j' 是对应 C_j 的合取范式, 并且

C_j 是可满足的当且仅当 F_j' 是可满足的.

证明

(1) $C_j=z_1$. 引入两个新变元 y_{j1}, y_{j2} , 令

$$F_j' = (z_1 \vee y_{j1} \vee y_{j2}) \wedge (z_1 \vee \neg y_{j1} \vee y_{j2}) \wedge (z_1 \vee y_{j1} \vee \neg y_{j2}) \wedge (z_1 \vee \neg y_{j1} \vee \neg y_{j2}).$$

(2) $C_j=z_1 \vee z_2$. 引入一个新变元 y_j , 令

$$F_j' = (z_1 \vee z_2 \vee y_j) \wedge (z_1 \vee z_2 \vee \neg y_j).$$

(3) $C_j=z_1 \vee z_2 \vee z_3$. 令 $F_j' = C_j$.

(4) $C_j=z_1 \vee z_2 \vee \dots \vee z_k, k \geq 4$. 引入 $k-3$ 个新变元 $y_{j1}, y_{j2}, \dots, y_{j(k-3)}$, 令

$$F_j' = (z_1 \vee z_2 \vee y_{j1}) \wedge (\neg y_{j1} \vee z_3 \vee y_{j2}) \wedge (\neg y_{j2} \vee z_4 \vee y_{j3}) \\ \wedge \dots \wedge (\neg y_{j(k-4)} \vee z_{k-2} \vee y_{j(k-3)}) \wedge (\neg y_{j(k-3)} \vee z_{k-1} \vee z_k).$$

设赋值 t 满足 C_j , 则存在 i 使得 $t(z_i)=1$. 当 $i=1$ 或 2 时, 令 $t(y_{js})=0$

($1 \leq s \leq k-3$); 当 $i=k-1$ 或 k 时, 令 $t(y_{js})=1$ ($1 \leq s \leq k-3$); 当 $3 \leq i \leq k-2$ 时, 令 $t(y_{js})=1$ ($1 \leq s \leq i-2$), $t(y_{js})=0$ ($i-1 \leq s \leq k-3$). 则有, $t(F_j')=1$.

证明

反之, 设 $t(F_j')=1$. 若 $t(y_{j1})=0$, 则 $t(z_1 \vee z_2)=1$; 若 $t(y_{j(k-3)})=1$, 则 $t(z_{k-1} \vee z_k)=1$; 否则必有 $s(1 \leq s \leq k-4)$ 使得 $t(y_{js})=1$ 且 $t(y_{j(s+1)})=0$, 从而 $t(z_{s+2})=1$. 总之, 都有 $t(C_j)=1$.

F_j' 中简单析取式的个数不超过 C_j 中文字个数的4倍, 每个简单析取式有3个文字, 因此可以在 $|F|$ 的多项式时间内构造出 F' .

局部替换法 要证 $\Pi_1 \leq_p \Pi_2$. 当 Π_2 是 Π_1 的子问题或两者的结构相似时, 往往可以把 Π_1 的实例的每一个子结构替换成对应的 Π_2 实例的子结构.

7.3.2 顶点覆盖、团与独立集

设无向图 $G=\langle V, E \rangle$, $V' \subseteq V$. V' 是 G 的一个

顶点覆盖: G 的每一条边都至少有一个顶点在 V' 中.

团: 对任意的 $u, v \in V'$ 且 $u \neq v$, 都有 $(u, v) \in E$.

独立集: 对任意的 $u, v \in V'$, 都有 $(u, v) \notin E$.

引理7.13 对任意的无向图 $G=\langle V, E \rangle$ 和子集 $V' \subseteq V$, 下述命题是等价的:

- (1) V' 是 G 的顶点覆盖,
- (2) $V-V'$ 是 G 的独立集,
- (3) $V-V'$ 是补图 $G^c=\langle V, E^c \rangle$ 的团.

顶点覆盖、团与独立集

顶点覆盖(VC): 任给一个无向图 $G=\langle V,E \rangle$ 和非负整数 $K \leq |V|$, 问 G 有顶点数不超过 K 的顶点覆盖吗?

团: 任给一个无向图 $G=\langle V,E \rangle$ 和非负整数 $J \leq |V|$, 问 G 有顶点数不小于 J 的团吗?

独立集: 任给一个无向图 $G=\langle V,E \rangle$ 和非负整数 $J \leq |V|$, 问 G 有顶点数不小于 J 的独立集吗?

根据引理7.13, 很容易把这3个问题中的一个问题多项式时间变换到另一个问题.

顶点覆盖

定理7.14 顶点覆盖是NP完全的.

证: VC的非确定型多项式时间算法: 任意猜想一个子集 $V' \subseteq V$, $|V'| \leq K$, 检查 V' 是否是一个顶点覆盖.

要证 $3SAT \leq_p VC$. 任给变元 x_1, x_2, \dots, x_n 的3元合取范式

$F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, 其中 $C_j = z_{j1} \vee z_{j2} \vee z_{j3}$, z_{jk} 是某个 x_i 或 $\neg x_i$.

如下构造VC的实例 $f(F): G = \langle V, E \rangle$ 和 $K = n + 2m$, 其中 $V = V_1 \cup V_2$,

$$E = E_1 \cup E_2 \cup E_3,$$

$$V_1 = \{ x_i, x_i^f \mid 1 \leq i \leq n \}$$

$$E_1 = \{ (x_i, x_i^f) \mid 1 \leq i \leq n \}$$

证明

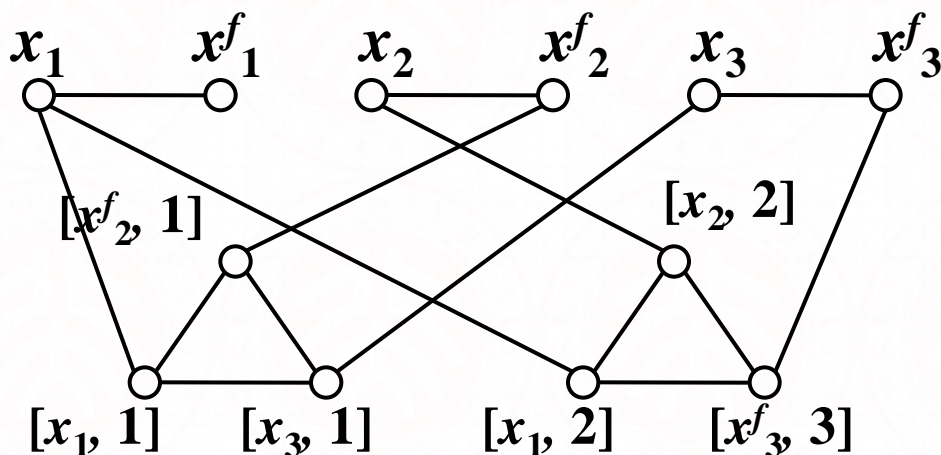
$$V_2 = \{[z'_{jk}, j] \mid k=1,2,3, 1 \leq j \leq m\},$$

$$E_2 = \{([z'_{j1}, j], [z'_{j2}, j]), ([z'_{j2}, j], [z'_{j3}, j]), ([z'_{j3}, j], [z'_{j1}, j]) \mid 1 \leq j \leq m\}.$$

$$E_3 = \{([z'_{jk}, j], z'_{jk}) \mid k=1,2,3, 1 \leq j \leq m\}$$

这里设 $C_j = z_{j1} \vee z_{j2} \vee z_{j3}$, 当 $z_{jk} = x_i$ 时, $z'_{jk} = x_i$; 当 $z_{jk} = \neg x_i$ 时, $z'_{jk} = x_i^f$.

例如, 对应 $F = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$ 的 $f(F): K=7$, 图 G 如下



证明

要证 F 是可满足的 $\Leftrightarrow G$ 恰好有 K 个顶点的顶点覆盖.

任何顶点覆盖 V' 在 x_i 和 x_i^f 中至少取一个, 在 $[z'_{j1}, j]$ 、 $[z'_{j2}, j]$ 和 $[z'_{j3}, j]$ 中至少取2个, 故 V' 至少有 $n+2m$ 个顶点. 而 $K=n+2m$, 故任何顶点数不超过 K 的顶点覆盖 V' 恰好包含 K 个顶点, 且在 x_i 和 x_i^f 中取一个, 这恰好对应对 x_i 的赋值, 取 x_i 对应 $t(x_i)=1$, 取 x_i^f 对应 $t(x_i)=0$; 每个三角形的顶点 $[z'_{j1}, j]$ 、 $[z'_{j2}, j]$ 和 $[z'_{j3}, j]$ 中取2个. 设 t 是 F 的成真赋值, 对每一个 $i(1 \leq i \leq n)$, 若 $t(x_i)=1$, 则取 x_i ; 若 $t(x_i)=0$, 则取 x_i^f . 这 n 个顶点覆盖 E_1 . 对每一个 $j(1 \leq j \leq m)$, 由于 $t(C_j)=1$, C_j 至少有一个文字 z_{jk} 的值为1. 于是, 从对应的三角形的顶点 $[z'_{jk}, j]$ 引出的边 $([z'_{jk}, j], z'_{jk})$ 已被覆盖. 取该三角形的另外2个顶点, 这就覆盖了这个三角形的3条边和引出的另外2条边. 这样取到的 $n+2m$ 个顶点是 G 的一个顶点覆盖.

证明

反之, 设 $V' \subseteq V$ 是 G 的一个顶点覆盖且 $|V'| \leq K = n + 2m$. 根据前面的分析, 每一对 x_i 和 x_i^f 中恰好有一个属于 V' , 每一个三角形恰好有 2 个顶点属于 V' . 对每一个 $i (1 \leq i \leq n)$, 若 $x_i \in V'$, 则令 $t(x_i) = 1$; 若 $x_i^f \in V'$, 则令 $t(x_i) = 0$. 对每一个 $j (1 \leq j \leq m)$, 设 $[z'_{jk}, j] \notin V'$, 为了覆盖边 $([z'_{jk}, j], z'_{jk})$, 必有 $z'_{jk} \in V'$. 由于 $t(z_{jk}) = 1$, 从而 $t(C_j) = 1$. 因此, t 是 F 的成真赋值, 得证 F 是可满足的.

G 有 $2n + 3m$ 个顶点和 $n + 6m$ 条边, 显然能在多项式时间内构造 G 和 K ,

定理 7.15 独立集和团是 NP 完全的.

构件设计法

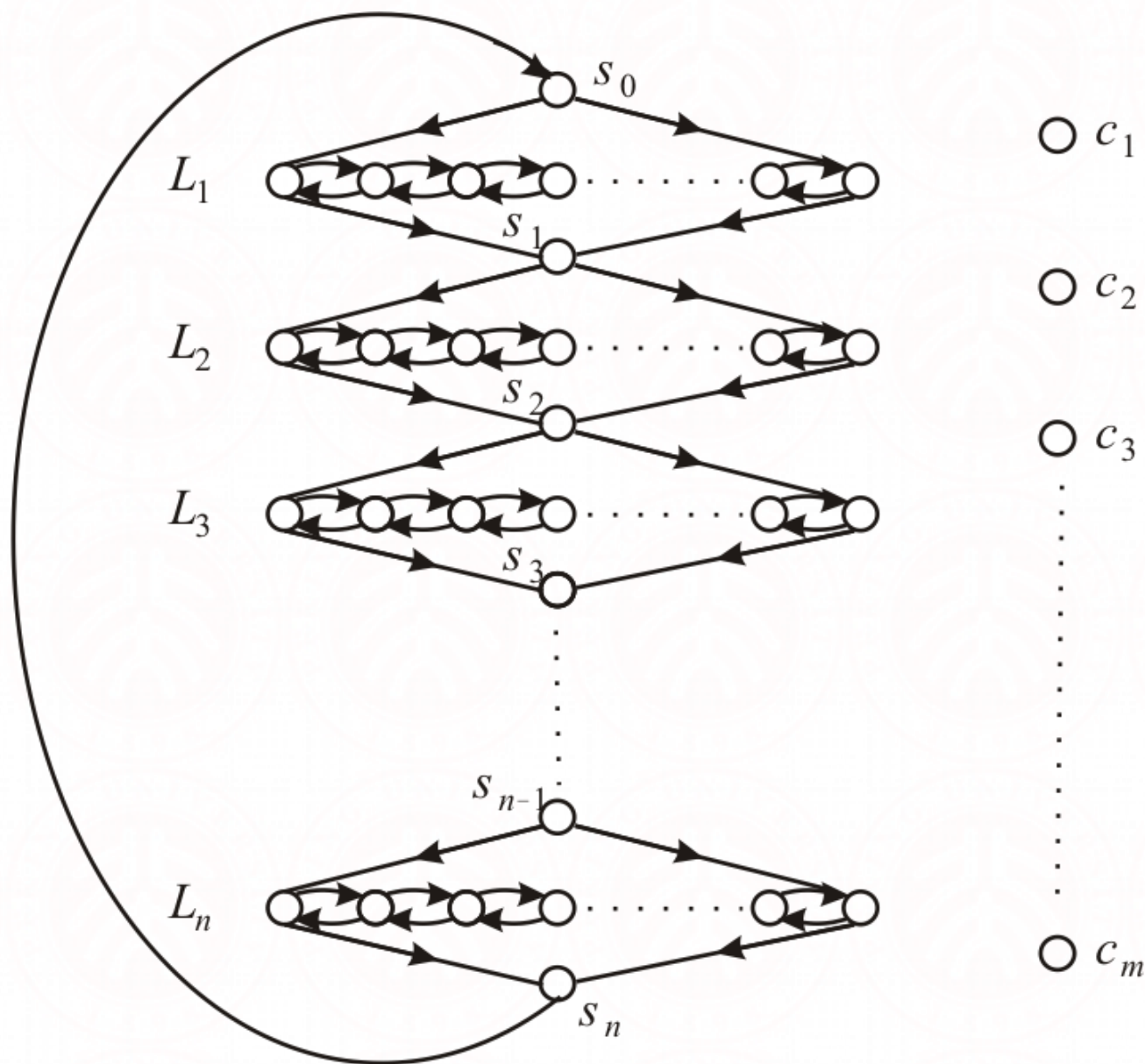
构件设计法 定理7.14证明中设计了2种“构件”——变元构件和简单析取式构件. 变元构件是一对顶点 x_i, x_i^f 及连接它们的边; 简单析取式构件是三角形. 用这些构件及构件之间的连接构成 G , 每个构件各有其功能, 通过这种方式到达用VC的实例表达3SAT的实例的目的.

7.3.3 哈密顿回路与货郎问题

有向哈密顿回路: 任给有向图 D , 问: D 中有哈密顿回路吗?

定理7.16 有向HC是NP完全的.

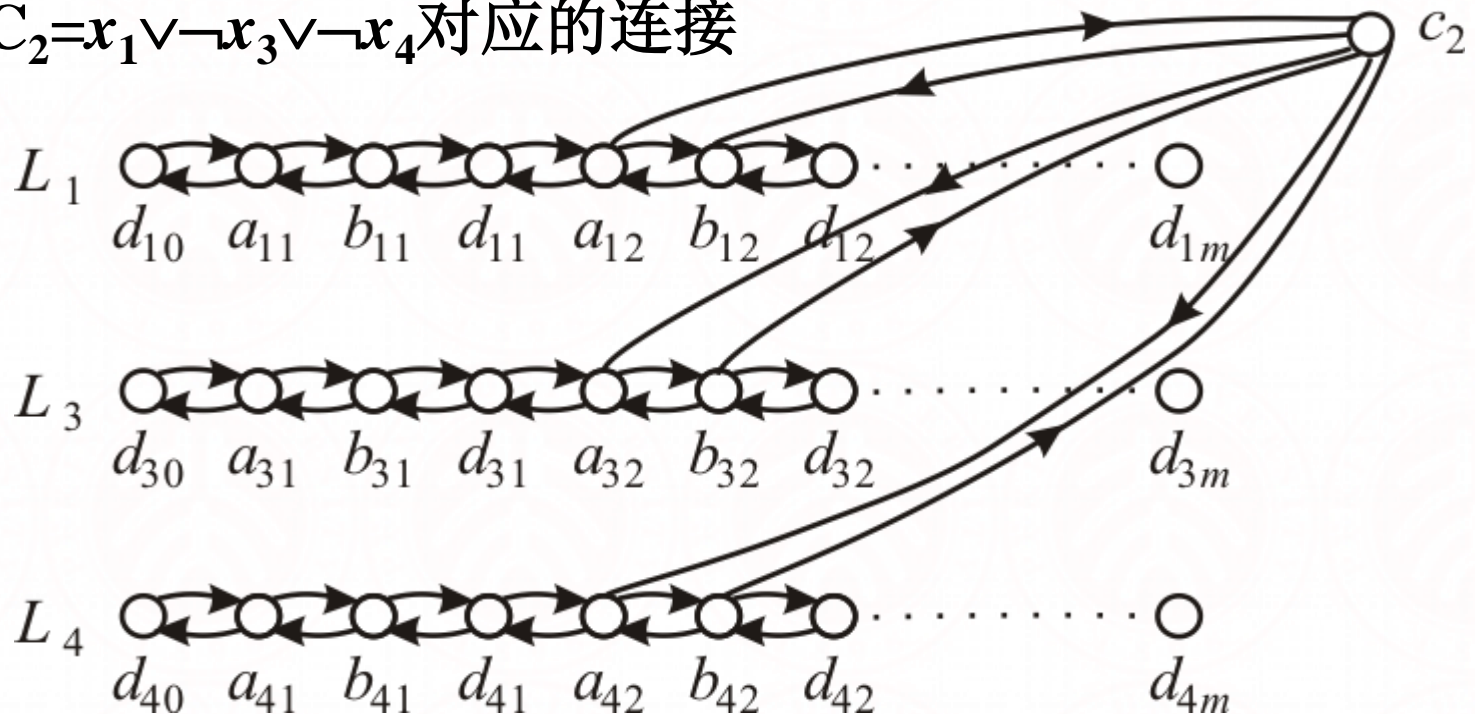
证 要证 $3SAT \leq_p$ 有向HC. 任给变元 x_1, x_2, \dots, x_n 的3元合取范式 $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, 其中 $C_j = z_{j1} \vee z_{j2} \vee z_{j3}$, 每个 z_{jk} 是某个 x_i 或 $\neg x_i$. 采用构件设计法构造有向图 D . 表示变元 x_i 的构件是一条由一串水平的顶点组成的链 L_i , 相邻的两个顶点之间有一对方向相反的有向边. 只有两种可能的方式通过 L_i 上的所有顶点——从左到右或者从右到左通过 L_i 上的所有顶点, 这恰好对应 x_i 的值为1或者为0. 表示简单析取式 C_j 的构件是一个顶点 c_j . 添加 s_0, s_1, \dots, x_n , 并通过它们把 L_1, L_2, \dots, L_n 连接起来.



证明

关键是两种构件之间的连接: 链 L_i 有 $3m+1$ 的顶点, 依次为 $d_{i0}, a_{i1}, b_{i1}, d_{i1}, a_{i2}, b_{i2}, d_{i2}, \dots, a_{im}, b_{im}, d_{im}$. 对每一个 $C_j = z_{j1} \vee z_{j2} \vee z_{j3}$, 如果 $z_{jk} = x_i$, 则添加 $\langle a_{ij}, c_j \rangle$ 和 $\langle c_j, b_{ij} \rangle$; 如果 $z_{jk} = \neg x_i$, 则添加 $\langle c_j, a_{ij} \rangle$ 和 $\langle b_{ij}, c_j \rangle$.

例如 $C_2 = x_1 \vee \neg x_3 \vee \neg x_4$ 对应的连接



证明

设 F 是可满足的, t 是 F 的成真赋值. 要根据 t 构造一条从 s_0 到 s_n , 最后回到 s_0 的哈密顿回路, 先暂时不考虑所有的 c_j . 依次对 $i=1,2,\dots,n$ 进行, 若 $t(x_i)=1$, 则从 s_{i-1} 到 d_{i0} , 从左到右经过 L_i 的所有顶点到达 d_{im} , 再到 s_i ; 若 $t(x_i)=0$, 则从 s_{i-1} 到 d_{im} , 从右到左经过 L_i 的所有顶点到达 d_{i0} , 再到 s_i . 最后, 从 s_n 回到 s_0 . 现在要将所有 c_j 插入这条回路. 设 $C_j=z_{j1} \vee z_{j2} \vee z_{j3}$, 由于 $t(C_j)=1$, 必有 $k(1 \leq k \leq 3)$ 使得 $t(z_{jk})=1$. 若 $z_{jk}=x_i$, 则通路从左到右经过 L_i , 且有有向边 $\langle a_{ij}, c_j \rangle$ 和 $\langle c_j, b_{ij} \rangle$. 于是, 可以把 c_j 插在 a_{ij} 与 b_{ij} 之间; 若 $z_{jk}=\neg x_i$, 则通路从右到左经过 L_i , 且有有向边 $\langle b_{ij}, c_j \rangle$ 和 $\langle c_j, a_{ij} \rangle$. 于是, 可以把 c_j 插在 b_{ij} 与 a_{ij} 之间. 这就得到 D 中的一条哈密顿回路.

证明

反之, 设 D 有一条哈密顿回路 P , P 必须从 s_n 到 s_0 . 不妨设 P 从 s_0 开始到 s_n , 最后回到 s_0 结束. 我们称上面构造的那种哈密顿回路是正常的, 即正常的回路从左到右或者从右到左通过每一条 L_i , 每一个 c_j 插在某个 a_{ij} 和 b_{ij} 或者 b_{ij} 和 a_{ij} 之间. 如果 P 是正常的, 容易根据 P 规定 F 的一个成真赋值 t : 若 P 从左到右通过 L_i , 则令 $t(x_i)=1$; 若 P 从右到左通过 L_i , 则令 $t(x_i)=0$. 根据 c_j 插入 L_i 的方式, 不难证明必有 $t(C_j)=1$.

要证 P 一定是正常的. 假设不然, 破坏正常性的惟一可能是 P 从某条链 L_s 上的顶点 u 到 c_j 后没有回到同一条链中的顶点, 而是到另一条链 $L_t(s \neq t)$ 中的顶点. 若 $u=a_{sj}$, 由于 b_{sj} 只与 a_{sj} 、 c_j 及 d_{sj}

证明

相邻, P 已经过 a_{sj} 和 c_j , b_{sj} 只剩下一个相邻的顶点, 故 P 不可能通过 b_{sj} . 若 $u=b_{sj}$, 由于 a_{sj} 只与 b_{sj} 、 c_j 及 $d_{s(j-1)}$ 相邻, P 已经过 b_{sj} 和 c_j , a_{sj} 也只剩下一个相邻的顶点, 故 P 不可能通过 a_{sj} . 都与 P 是哈密顿回路矛盾, 所以 P 一定是正常的.

构造 D 可以在多项式时间内完成.

HC与TSP

定理7.17 HC是NP完全的.

证 要证有向HC \leq_p HC. 任给一个有向图 $D=<V,E>$, 要构造无向图 $G=<V',E'>$ 使 D 有哈密顿回路当且仅当 G 有哈密顿回路. 把 D 的每一个顶点 v 替换成3个顶点 v^{in} , v^{mid} 和 v^{out} , 用边连接 v^{in} 和 v^{mid} , v^{mid} 和 v^{out} . D 的每条有向边 $<u,v>$ 在 G 中换成 $(u^{\text{out}}, v^{\text{in}})$.

即
$$V' = \{ v^{\text{in}}, v^{\text{mid}}, v^{\text{out}} \mid v \in V \},$$

$$E' = \{ (u^{\text{out}}, v^{\text{in}}) \mid <u,v> \in E \} \cup \{ (v^{\text{in}}, v^{\text{mid}}), (v^{\text{mid}}, v^{\text{out}}) \mid v \in V \}.$$

定理17.18 TSP是NP完全的.

7.3.4 恰好覆盖

恰好覆盖: 给定有穷集 $A=\{a_1, a_2, \dots, a_n\}$ 和 A 的子集的集合 $W=\{S_1, S_2, \dots, S_m\}$, 问: 存在子集 $U \subseteq W$ 使得 U 中的子集都是不相交的且它们的并集等于 A ? 称 W 这样的子集 U 是 A 的恰好覆盖.

例如, 设 $A=\{1,2,3,4,5\}$, $S_1=\{1,2\}$, $S_2=\{1,3,4\}$, $S_3=\{2,4\}$, $S_4=\{2,5\}$, 则 $\{S_2, S_4\}$ 是 A 的恰好覆盖. 若把 S_4 改为 $S_4=\{3,5\}$, 则不存在 A 的恰好覆盖.

定理7.19 恰好覆盖是NP完全的.

证明

证 要证可满足性 \leq_p 恰好覆盖. 任给变元 x_1, x_2, \dots, x_n 的合取范式 $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, 其中 $C_j = z_{j1} \vee z_{j2} \vee \dots \vee z_{js_j}$

$$A = \{x_1, x_2, \dots, x_n, C_1, C_2, \dots, C_m\} \cup \{p_{jt} \mid 1 \leq t \leq s_j, 1 \leq j \leq m\},$$

其中 p_{jt} 代表 C_j 中的文字 z_{jt} . W 包含下述子集:

$$T_i^T = \{x_i, p_{jt} \mid z_{jt} = \neg x_i, 1 \leq t \leq s_j, 1 \leq j \leq m\}, \quad 1 \leq i \leq n,$$

$$T_i^F = \{x_i, p_{jt} \mid z_{jt} = x_i, 1 \leq t \leq s_j, 1 \leq j \leq m\}, \quad 1 \leq i \leq n,$$

$$C_{jt} = \{C_j, p_{jt}\}, \quad 1 \leq t \leq s_j, 1 \leq j \leq m.$$

$$\{p_{jt}\}, \quad 1 \leq t \leq s_j, 1 \leq j \leq m.$$

要证 F 是可满足的当且仅当 W 含有 A 的恰好覆盖. 设 $U \subseteq W$ 是 A 的恰好覆盖, 对每一个 i , 若 $T_i^T \in U$, 则令 $t(x_i) = 1$; 若 $T_i^F \in U$, 则令 $t(x_i) = 0$. 对每一个 j , 必有一个 $C_{jt} = \{C_j, p_{jt}\} \in U$. $z_{jt} = x_i$ 或 $\neg x_i$. 若 $T_i^T \in U$, 则 $p_{jt} \notin T_i^T$, 从而 $z_{jt} = x_i$. 有 $t(x_i) = 1$, 故 t 满足 C_j .

证明

若 $T_i^F \in U$, 则 $p_{jt} \notin T_i^F$, 从而 $z_{jt} = \neg x_i$. 有 $t(x_i) = 0$, 故 t 也满足 C_j . 得证 t 是 F 的成真赋值.

反之, 设 t 是 F 的成真赋值. 对每一个 i , 若 $t(x_i) = 1$, 则 U 包含 T_i^T ; 若 $t(x_i) = 0$, 则 U 包含 T_i^F . 对每一个 j , 由于 t 满足 C_j , C_j 必有一个文字 z_{jt} 使得 $t(z_{jt}) = 1$, 从而 U 中现有的子集不包含 p_{jt} . 于是, 可以把 C_{jt} 加入 U . 至此, U 覆盖了所有的 x_i 和 C_j , 以及部分 p_{jt} . 最后, 把那些尚未被覆盖的 p_{jt} 构成的单元子集 $\{p_{jt}\}$ 加入 U , 即可得到 A 的恰好覆盖.

由于 F 中的文字数不超过 mn , 故 $|A| \leq n + m + mn$, W 中的子集数不超过 $2n + 2mn$, 每个子集的大小不超过 $n + 1$. 而且构造很简单, 显然可以在多项式时间内完成.

7.3.5 子集和,背包,装箱与双机调度

子集和: 给定正整数集合 $X=\{x_1, x_2, \dots, x_n\}$ 及正整数 N , 问存在 X 的子集 T , 使得 T 中的元素之和等于 N 吗?

装箱: 给定 n 件物品, 物品 j 的重量为正整数 w_j , $1 \leq j \leq n$, 以及箱子数 K . 规定每只箱子装入物品的总重量不超过正整数 B , 问能用 K 只箱子装入所有的物品吗?

双机调度: 有2台机器和 n 项作业 J_1, J_2, \dots, J_n , 这2台机器完全相同, 每一项作业可以在任一台机器上进行, 没有先后顺序, 作业 J_i 的处理时间为 t_i , $1 \leq i \leq n$, 截止时间为 D , 所有 t_i 和 D 都是正整数, 问能把 n 项作业分配给这2台机器, 在截止时间 D 内完成所有的作业吗?

子集和

子集和是0-1背包的子问题——限制0-1背包的实例中所有 $w_i=v_i$ 且 $B=K$.

双机调度可以看作当箱子数 $K=2$ 时装箱的特殊情况——把物品看作作业, 物品的重量是作业的处理时间, 截止时间是每只箱子允许的最大重量.

定理7.20 子集和是NP完全的.

证 要证恰好覆盖 \leq_p 子集和. 给定有穷集 $A=\{a_1, a_2, \dots, a_n\}$ 和 A 的子集的集合 $W=\{S_1, S_2, \dots, S_m\}$, 对应的子集和实例包括非负整数的集合 $X=\{x_1, x_2, \dots, x_n\}$ 及非负整数 N , 每个 x_j 和 N 都可表成 kn 位的二进制数, 这 kn 位分成 n 段, 每段 k 位, $k=\lceil \log_2(m+1) \rceil$.

证明

N 的每一段的第一位(最右的一位)为1, 其余的为0. x_j 对应于子集 S_j . 当 $a_i \in S_j$ 时, 从左到右 x_j 的第 i 段的第一位为1, 其余的为0.

例如, $A=\{a_1, a_2, a_3, a_4\}$, $S_1=\{a_1, a_2\}$, $S_2=\{a_1, a_3, a_4\}$, $S_3=\{a_2\}$,

$N=01010101$, $x_1=01010000$, $x_2=01000101$, $x_3=00010000$.

要证 W 中有 A 的恰好覆盖当且仅当存在子集 $T \subseteq X$ 使得 T 中元素之和等于 N . 设 $U \subseteq W$ 是 A 的恰好覆盖, 令 $T=\{x_j \mid S_j \in U\}$. 由于 A 中的每一个元素在 U 的所有 S_j 中恰好出现一次, 故对于二进制数的每一段, 在 T 的所有 x_j 中恰好有一个的这一位为00...01, 从而 T 中所有元素之和等于 N .

证明

反过来, 设 X 的子集 T 中元素之和等于 N , 令 $U=\{S_j \mid x_j \in T\}$. T 中至多有 m 个数, 每一段有 $k=\lceil \log_2(m+1) \rceil$ 位, 最大值为 $2^k-1 \geq m$, 故 T 中的数相加时不会出现段之间的进位. 从而, 对于每一段, 在 T 的所有 x_j 中恰好有一个的这一段为 $00\dots 01$, 这意味着每一个 a_i 在 U 的所有 S_j 中恰好出现一次, 即 U 是 A 的恰好覆盖. 构造 X 和 N 显然可以在多项式时间内完成,

0-1背包与伪多项式时间算法

定理7.21 0-1背包是NP完全的.

注意: 0-1背包问题优化形式的动态规划算法, 其时间复杂度为 $O(nB)$, 其中 n 是物品的个数, B 是重量限制. 这不是多项式时间算法, 而是指数时间算法.

伪多项式时间算法: 算法的时间复杂度以 $|I|$ 和 $\max(I)$ 的某个二元多项式 $p(|I|, \max(I))$ 为上界, 其中 $\max(I)$ 是实例 I 中数的最大绝对值.

双机调度与装箱

定理7.22 双机调度是NP完全的.

证要证子集和 \leq_p 双机调度. 任给一个子集和实例: 非负整数集合 $X=\{x_1, x_2, \dots, x_n\}$ 及非负整数 N , 对应的双机调度实例有 $n+2$ 项作业 J_1, J_2, \dots, J_{n+2} , 处理时间为 $x_1, x_2, \dots, x_n, a, b$, 截止时间为 D . 要求存在 X 的子集 T 使得 当且仅当 $N+a = M-N+b = D$.

于是, $a = M-2N+b$. 取

$$b = M + 2N, \quad a = 2M, \quad D = 2M + N,$$

其中 $M = x_1 + x_2 + \dots + x_n$.

定理7.23 装箱是NP完全的.

小结

证明问题是NP难的

- (1) 要选好一个已知的NP完全问题.
- (2) 常用的证明方法:限制法, 局部替换法和构件设计法.