计算复杂性与图算法

主讲人：李泽鹏

2019年9月12日

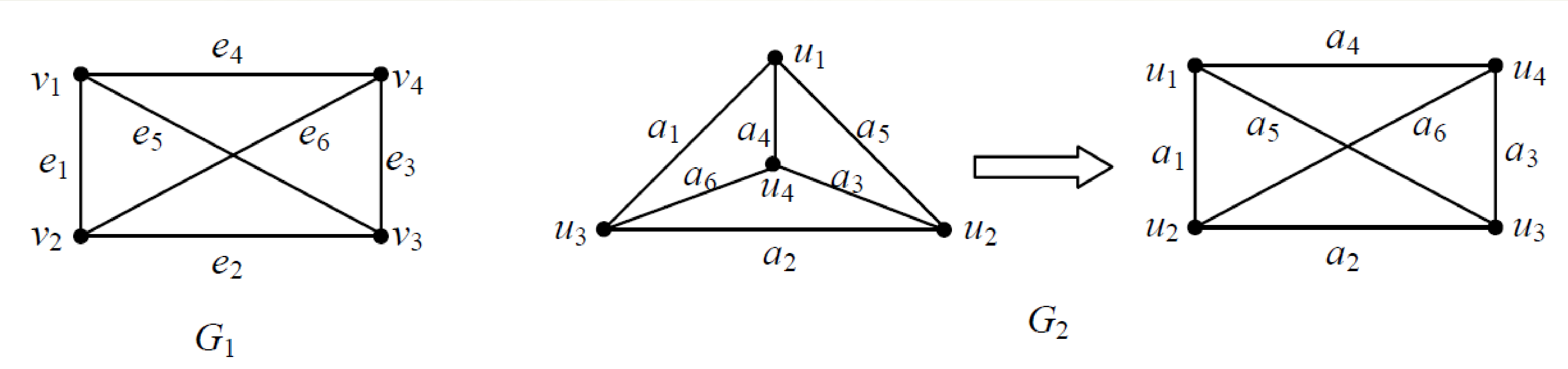
第 1 章 计算复杂性

1. 若干优化问题

(a) 旅行商问题（TSP问题）

TSP问题： 有 n 个城镇。一个旅行商从某一城镇出发，欲经过其余n-1个城镇一次且仅一次，最后回到出发点。他应该如何设计旅行路线，才能使所走路程最短？

(b) 图同构问题



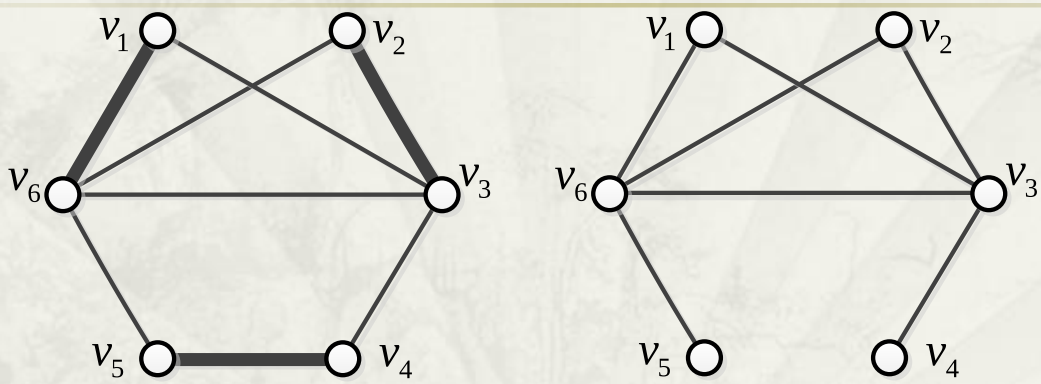
易见G_1和G_2的顶点及边之间都一一对应，且连接关系完全相同，只是顶点和边的名称不同。这样的两个图称为是同构的(isomorphic)。

图同构问题：对两个顶点数都为n的图，如何判断它们是否同构？

(c) 完美匹配问题

匹配：图中互不相邻的边子集。

完美匹配：包含整个图所有顶点的匹配。



完美匹配问题：给定一个图G，判断G是否含有完美匹配？

2. 计算复杂性

算法：是指有限条指令的集合，这个指令集确定了解决某个问题的运算或操作的序列。

算法A解问题Q是指：把问题Q的任何实例作为算法A的输入，A能够在有限步停机，并输出该实例的正确解。

算法的复杂度是对**算法效率**的度量。
包含时间复杂度和空间复杂度。

多项式时间算法：对规模为n的输入，它的运行时间至多为$O(n^k)$，其中k为常数。

P (Polynomial) **问题（P-类）：**
指的是在多项式时间内可解的问题，即可以在$O(n^k)$内求解的问题，其中k为某个常数，n为问题的输入规模。

例如排序问题、查找问题等。

NP(Nondeterministic Polynomial)**问题(NP-类)：**
指的是在多项式时间内“可验证”的问题，即给定问题的一组解，在问题输入规模的多项式时间内可以验证该解是否是正确的。

P-类中的任何问题都属于NP-类，因为任意P问题在多项式时间内有解，那么给定一组解，在多项式时间内肯定是可验证该解的正确性的。

P=NP?

该问题是理论计算机科学领域中非常重要的问题，至今没有解决。它被“克雷数学研究所”(Clay Mathematics Institute)列为千禧年大奖难题之首。

3. NP-完全问题

优化问题： 是指求解满足给定限制条件的最优解的问题。（如上述TSP问题的形式）

判定问题： 是指答案只有两个——“是”或“否”，或者“1”和“0”——的问题。（如上述图同构问题和完美匹配问题的形式）

形式上，判定问题Q可定义为有序对<D_Q,Y_Q>,其中D_Q是实例集合，由Q的所有可能的实例组成；$Y_Q \subseteq D_Q$由所有答案为“是”的实例组成。

多项式归约： 设判定问题Q_1=<D_1,Y_1>，Q_2=<D_2, Y_2>. 如果函数 $f: D_1 \to D_2$满足条件:

（1）f是多项式时间可计算的，即存在计算f的多项式时间算法;

（2）对所有的$I \in D_1$, $I \in Y_1$当且仅当$f(I) \in Y_2$.

则称f是Q_1到Q_2的多项式时间归约，记为

$$Q_1 \leqslant Q_2 \text{。}$$

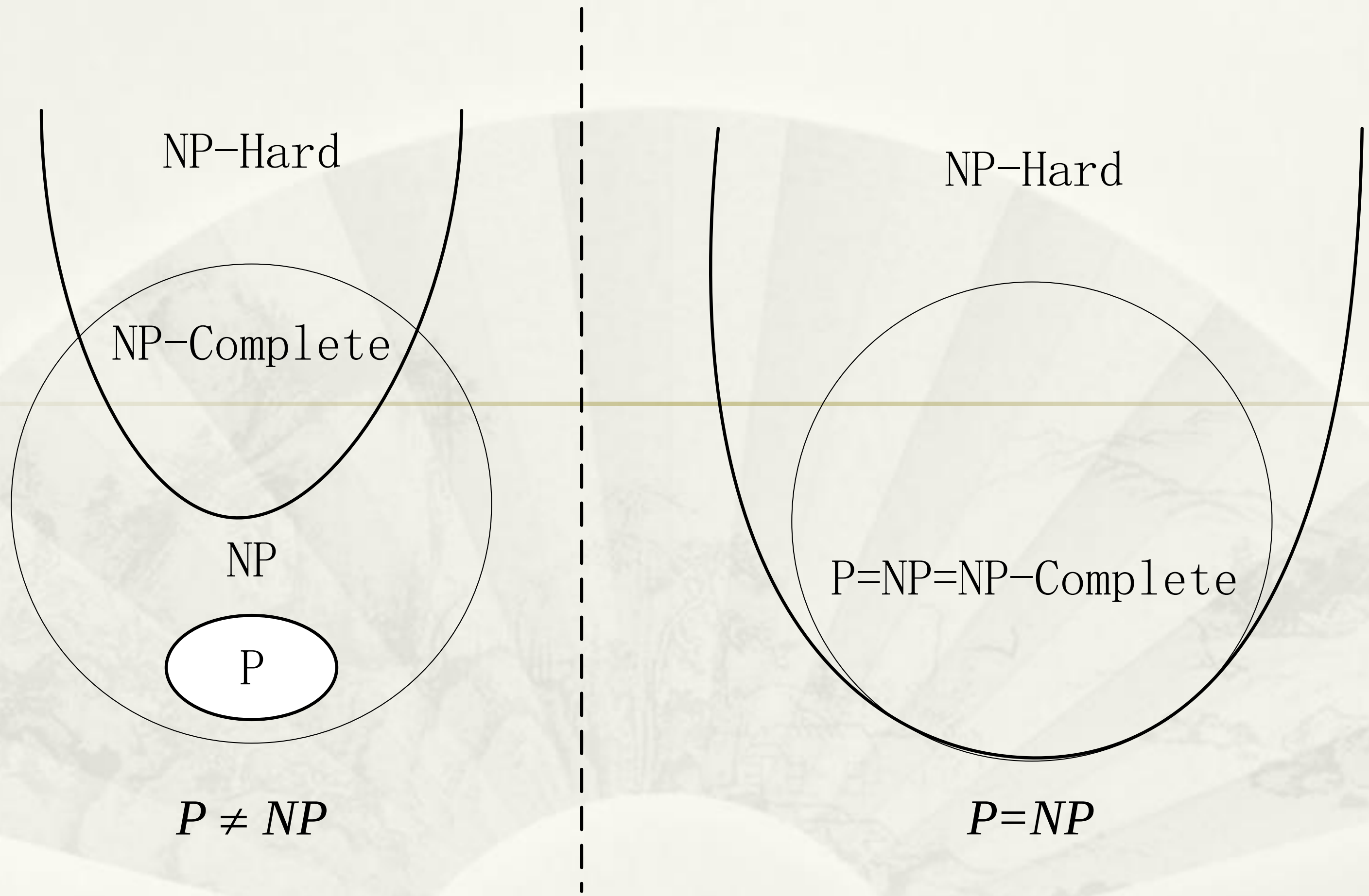
多项式时间归约具有**传递性**：

设$Q_1 \leqslant Q_2$，$Q_2 \leqslant Q_3$，则$Q_1 \leqslant Q_3$.

NP-难与NP-完全：

如果对所有的$Q' \in \mathbf{NP}$，都有$Q' \leqslant Q$，则称Q是NP-难的(NP-hard)。

如果Q是NP-难的且$Q \in \mathbf{NP}$，则称Q是NP-完全的(NP-complete) 。


NP-Hard
NP-Complete
NP
P
P ≠ NP
NP-Hard
P=NP=NP-Complete
P=NP

定理： 设$Q_1 \leqslant Q_2$， 那么

（1）若$Q_2 \in$ **P**， 则$Q_1 \in$ **P** 。

（2）若Q_1是NP-难的， 则Q_2也是NP-难的。

（3）若Q_2 ∈**NP**并且Q_1是NP-完全的， 则Q_2也是NP-完全的。

4. Cook-Levin定理——第一个NP-完全问题

20世纪70年代初，S.A.Cook和L.A.Levin分别独立地证明了第一个NP完全问题，这是命题逻辑中的一个基本问题.

在命题逻辑中，变元的取值为0或1，其中0表示“假”，1表示“真”. 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.

例如:

$$F_1 = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \to (x_1 \vee \overline{x_2})$$

$$F_2 = (x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge \overline{x_2}$$

$$F_3 = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge x_2 \wedge \overline{x_3}$$

F_1是合式公式，但不是合取范式；F_2和F_3是合取范式，当然也是合式公式.

设F是关于变元x_1，x_2，…，x_n的合式公式．给定每一个变元的真假值称作关于变元x_1，x_2，…，x_n的赋值.

如果赋值$t: \{x_1, x_2, \ldots, x_n\} \to \{0,1\}$使得 $t(F)=1$，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，令$t(x_1) = 1, t(x_2) = 0, t(x_3) = 1$．$t$是$F_1$和$F_2$的成真赋值，从而$F_1$和$F_2$是可满足的．$t$不是$F_3$的成真赋值，事实上$F_3$不是可满足的.

可满足性问题（SAT）: 任给一个合取范式F，问F是可满足的吗？

Cook－ Levin定理：SAT是NP-完全的.

5. 几个NP-完全问题

3SAT问题

3SAT问题就是一般SAT问题的特殊情况，3SAT问题的实例是指每个简单析取式（从句）中恰含3个文字的合取范式。

例如：

$$F = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3})$$

定理. 3SAT问题是NP-完全的。

证明： 首先，由于SAT问题属于NP类，因此3SAT问题也属于NP类。

下面证明SAT $\leqslant$3SAT. 任给一个合取范式F，要在多项式时间构造一个新的合取范式F'，使其满足：

1) F'的每个从句包含3个文字；

2) F是可满足的当且仅当F'是可满足的。

如果F的某个从句只包含1个文字，设为x，则将该从句替换为4个包含3个文字的从句，其中y, z是两个新变量。即

$$x = (x \lor y \lor z) \land (x \lor \bar{y} \lor z) \land (x \lor y \lor \bar{z}) \land (x \lor \bar{y} \lor \bar{z})$$

如果F的某个从句只包含2个文字，设为$(x_1 \lor x_2)$，则将该从句替换为2个包含3个文字的从句，其中w是新变量。即

$$(x_1 \lor x_2) = (x_1 \lor x_2 \lor w) \land (x_1 \lor x_2 \lor \overline{w})$$

下面考虑F中包含k个文字的从句$(x_1 \vee x_2 \vee \dots \vee x_k)$，其中$k \geq 4$。此时，添加$k-3$个新变量$y_1, y_2, \dots, y_{k-3}$，并构造如下的$k-2$个包含3个文字的从句：

$$(x_1 \vee x_2 \vee y_1)，(\overline{y_1} \vee x_3 \vee y_2)，(\overline{y_2} \vee x_4 \vee y_3)，\dots,$$
$$(\overline{y_{k-4}} \vee x_{k-2} \vee y_{k-3})，(\overline{y_{k-3}} \vee x_{k-1} \vee x_k)$$

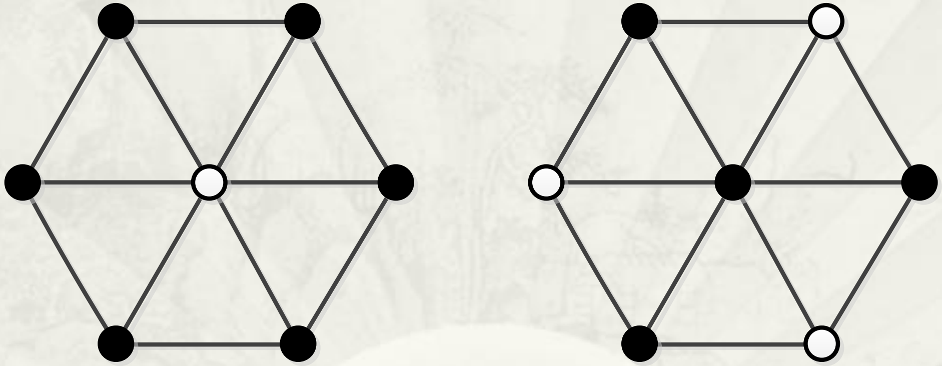
易验证，这$k-2$个从句的合取运算与$(x_1 \vee x_2 \vee \dots \vee x_k)$是等价的。因此，$F'$是可满足的当且仅当$F$是可满足的。

另外，此构造过程是多项式时间的，因此，SAT $\preccurlyeq$ 3SAT.

顶点覆盖：

设无向图$G = (V, E), V' \subseteq V$,如果$G$的每条边都至少有一个顶点在$V'$中，则称$V'$是$G$的一个顶点覆盖。

例如：

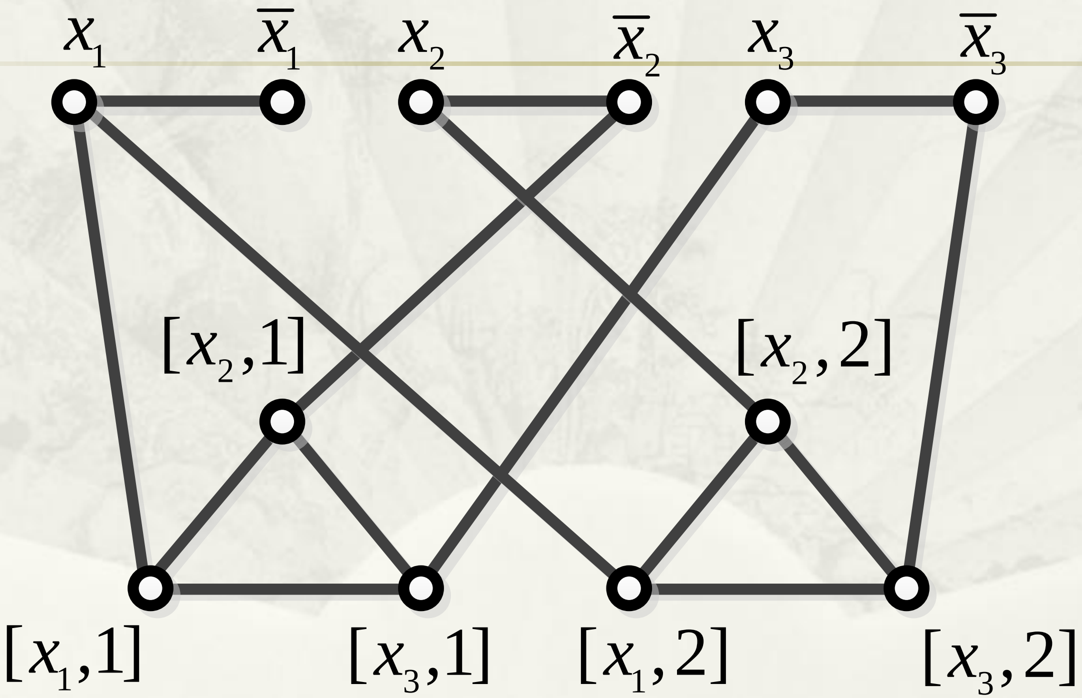


顶点覆盖问题（VC）

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

定理. VC是NP-完全的。

用3SAT进行归约。例如：$F=(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3})$



谢谢！