算法设计与分析

NP-完全

林海
ruitingzhou@whu.edu.cn

本章内容提要

- 易解问题与难解问题
- P类问题和NP类问题
- NP完全问题
- co-NP类问题
- NPI类问题
- P、NP、co-NP、NPI类之间的区别与联系
- NP完全问题的计算机处理技术简介

算法效率

复杂度为**O(n)**的算法是高效率?

复杂度为 **O(n log n)**?

$O(n^2)$?

$O(n^{10})$?

$O(n^{\log n})$?

$O(2^n)$?

O(n!) ?

多项式时间算法
$O(n^c)$ for some constant c

非多项式时间算法

引言

易解问题与难解问题

- 如果对一个问题Π存在一个算法，时间复杂性为$O(n^k)$，其中n是问题规模，k是一个非负整数，则称问题Π存在多项式时间算法。这类算法在可以接受的时间内实现问题求解， e.g., 排序、串匹配、矩阵相乘。
- 现实世界里还存在很多问题至今没有找到多项式时间算法，计算时间是指数和超指数函数（如2^n和n!），随着问题规模的增长而快速增长。
- 通常将前者看作是易解问题，后者看作难解问题。

易解问题与难解问题的主要区别

在学术界已达成这样的共识：把多项式时间复杂性作为易解问题与难解问题的分界线，主要原因有：

1) 多项式函数与指数函数的增长率有本质差别

问题规模 n	多项式函数					指数函数	
	logn	n	nlogn	n^2	n^3	2^n	n!
1	0	1	0.0	1	1	2	1
10	3.3	10	33.2	100	1000	1024	3628800
20	4.3	20	86.4	400	8000	1048376	2.4E18
50	5.6	50	282.2	2500	125000	1.0E15	3.0E64
100	6.6	100	664.4	10000	1000000	1.3E30	9.3E157

易解问题与难解问题的主要区别

2) 计算机性能的提高对易解问题与难解问题算法的影响

假设求解同一个问题有5个算法A1~A5，时间复杂度T(n)如下表，假定计算机C2的速度是计算机C1的10倍。下表给出了在相同时间内不同算法能处理的问题规模情况：

T(n)	n	n′	变化	n′/n
10n	1000	10000	n′=10n	10
20n	500	5000	n′=10n	10
5nlogn	250	1842	$\sqrt{10}$ n<n′<10n	7.37
$2n^2$	70	223	$\sqrt{10}$ n	3.16
2^n	13	16	n′=n+log10≈n+3	≈1

易解问题与难解问题的主要区别

3) 多项式时间复杂性忽略了系数，不影响易解问题与难解问题的划分

问题规模n	多项式函数			指数函数	
	n^8	10^8n	n^{1000}	1.1^n	$2^{0.01n}$
5	390625	5×10^8	5^{1000}	1.611	1.035
10	10^8	10^9	10^{1000}	2.594	1.072
100	10^{16}	10^{10}	10^{2000}	13780.6	2
1000	10^{24}	10^{11}	10^{3000}	2.47×10^{41}	1024

观察结论：n≤100时，（不自然的）多项式函数值大于指数函数值，但n充分大时，指数函数仍然超过多项式函数。

判定问题和最优化问题

判定问题：element uniqueness

- 输入：一个整数序列S
- 问题：在S中存在两个相等的元素吗？

最优问题：element count

- 输入：一个整数序列S
- 问题：一个在S中频度最高的元素

判定问题和最优化问题

判定问题：coloring

- 输入：一个无向图G=(V,E),一个正整数k>=1
- 问题：G可以k着色吗？

最优问题：chromatic number

- 输入：一个无向图G=(V,E),
- 问题：G的色数（最少颜色数）

P类问题和NP类问题

这个划分标准是基于对所谓判定问题的求解方式。

先看看什么是判定问题。事实上，实际应用中的大部分问题问题可以很容易转化为相应的判定问题，如：

- 排序问题 ⇒ 给定一个实数数组，是否可以按非降序排列?
- 图着色问题：给定无向连通图G=(V,E)，求最小色数k，使得任意相邻顶点具有不同的着色 ⇒

  给定无向连通图G=(V,E)和正整数k，是否可以用k种颜色.....？

P类判定问题例子

- 排序问题：给出一个n个整数的表，它们是否按非降序排列？
- 最短路径问题：给出一个边上有正权的有向图G = (V，E)，两个特异的顶点是s,t in V,和一个正整数k, 在s到t间是否存在一条路径，它的长度最多是k。

P类判定问题例子

- 2着色问题：给出一个无向图G,它是否是2可着色的？即它的顶点是否可仅用两种颜色着色，使两个邻接顶点不会分配相同的颜色？注意，当且仅当G是二分图，即当且仅当它不包含奇数长的回路时，它是2可着色的。
- 可满足问题：给出一个合取范式（CNF)形式的布尔表达式f，这里每个子句恰好由两个文字组成，问f是可满足的吗？

确定性算法与P类问题

对判定问题求解，可以采用确定性算法

- 定义10.1(确定性算法): 设A是求解问题Π的一个算法，如果在算法的整个执行过程中，每一步只有一个确定的选择，则称算法A是确定性算法。
  - 特点：对同一输入实例，运行算法A，所得结果是一样的。
- 定义10.2(P类问题): 如果对于某个判定问题Π，存在一个非负整数k，对于输入规模为n的实例，能够以$O(n^k)$的时间运行一个确定性算法，得到yes或no的答案，则称该判定问题Π是一个P(Polynomial)类问题。
  - 事实上，所有易解问题都是P类问题。

非确定性算法与NP类问题

- 定义(非确定性算法):设A是求解问题Π的一个算法，如果算法A以如下猜测+验证的方式工作，称算法A为非确定性(nondeterminism)算法：
- 猜测阶段：对问题的输入实例产生一个任意字串y，在算法的每次运行，y可能不同，因此猜测是以非确定的形式工作。这个工作一般可以在线性时间内完成。
- 验证阶段：在这个阶段，用一个确定性算法验证两件事：首先验证猜测的y是否是合适的形式，若不是，则算法停下并回答no；若是合适形式，则继续检查它是否是问题x的解，如果确实是x的解，则停下并回答yes，否则停下并回答no。要求验证阶段在多项式时间内完成。

注意对非确定性算法输出yes/no的理解：

- 若输出no，并不意味着不存在一个满足要求的解，因为猜测可能不正确；若输出yes，则意味着对于该判定问题的某一输入实例，至少存在一个满足要求的解。

NP类问题

- 定义(NP类问题): 如果对于判定问题Π，存在一个非负整数k，对于输入规模为n的实例，能够以$O(n^k)$的时间运行一个非确定性算法，得到yes/no的答案，则该判断问题Π是一个NP(nondeterministic polynomial)类问题。

※注意：NP类问题是对于判定问题定义的，事实上，可以在多项式时间内应用非确定性算法解决的所有问题都属于NP类问题。

例子

- 问题COLORING，NP？
- 当图G用编码表示后，一个算法A可以很容易地构建并运作如下。
  - 首先通过对顶点集合产生一个任意的颜色指派以“猜测”一个解；
  - 接着，A验证这个指派是否是有效的指派，如果它是一个有效的指派，那么A停下并且回答 yes, 否则 它停下并回答no。
  - 请注意，根据不确定性算法的定义，仅当对问题的实例回答是 yes时，A回答yes。其次是关于需要的运行时间，A在猜测和验证两个阶段总共花费不多 于多项式时间。

P与NP之间的不同

- P是一个判定问题类，这些问题可以用一个确定性算法在多项式时间内判定或解出；
- NP是一个判定问题类，这些问题可以用一个确定性算法在多项式时间内检查或验证它们的解。

关于P与NP关系的初步思考
--从字面含义

1) 若问题Π属于P类，则存在一个多项式时间的确定性算法，对它进行判定或求解；显然，也可以构造一个多项式时间的非确定性算法，来验证解的正确性，因此，问题也属NP类。因此，显然有

$$P \subseteq NP$$

2) 若问题Π属于NP类，则存在一个多项式时间的非确定性算法，来猜测并验证它的解；但不一定能构造一个多项式时间的确定性算法，来对它进行求解或判定。

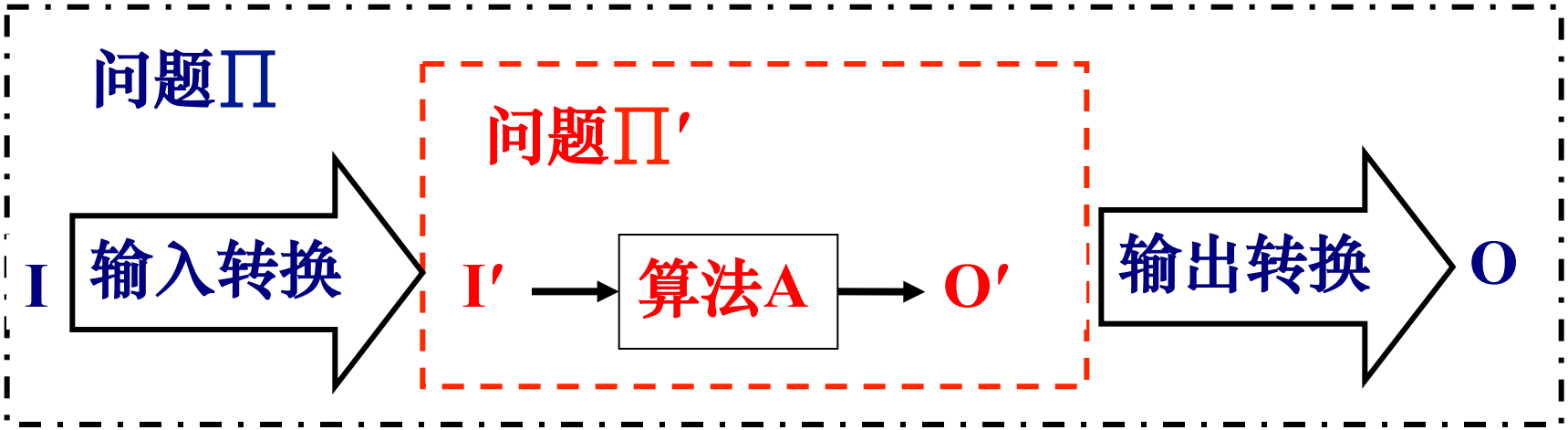
- 因此，人们猜测$P \neq NP$，但是否成立，至今未得到证明。

10.4 NP完全问题

- NP完全问题是NP类问题的子类，一个具有特殊性质与特殊意义的子类。

问题变换（归约）

- NP类问题在最坏情况下的时间复杂性一般都是快速增长的指数函数。我们希望能够在NP类问题内部找到一种方法，比较两个问题的复杂性。
- 比较两个问题的计算复杂性的方法是问题变换。



多项式时间变换（归约）

- 定义(多项式时间变换):若在O(τ(n))时间内完成上述输入/输出转换，则称问题Π以τ(n)时间变换到问题Π′，记为

$$\Pi \propto_{\tau(n)} \Pi'$$

- 其中，n为问题规模；若在多项式时间内完成上述转换，则称问题Π以多项式时间变换到问题Π′，记为

$$\Pi \propto_{poly} \Pi'$$

举例：多项式时间变换

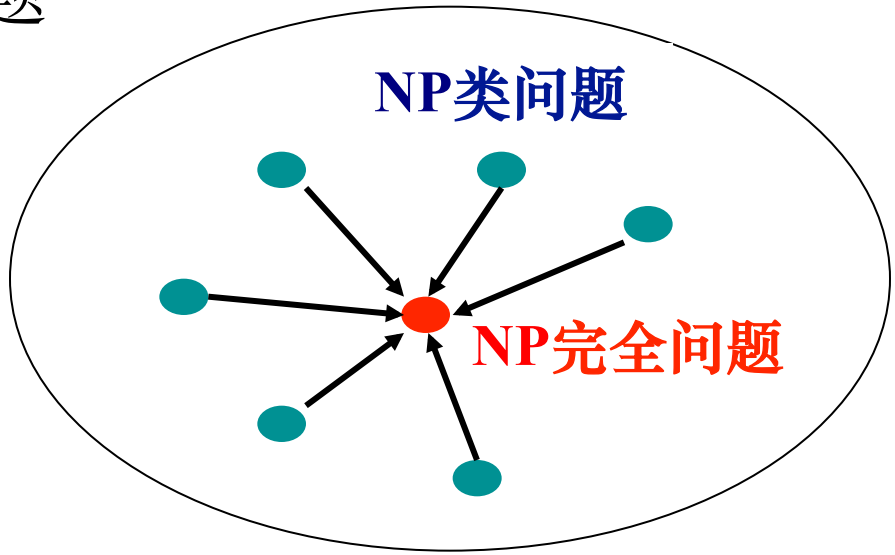
- 如：求解一元一次方程（问题A）可归为求解一元二次方程（问题B）
  - 一元二次方程的二次项系数为0
  - 用求解一元二次方程的解法来解一元一次方程
- 如：哈密顿回路可以归约为TSP问题（旅行商问题）
  - TSP是一个完全图

多项式时间变换的性质

- 传递性：设Π、Π’和Π’’是3个判定问题，若$\Pi \propto_{\tau(n)} \Pi'$，且$\Pi' \propto_{\tau(n)} \Pi''$，则$\Pi \propto_{\tau(n)} \Pi''$。

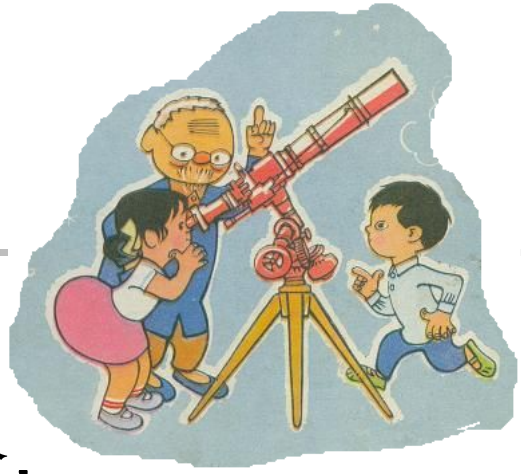
NP完全问题

NP完全问题是一类具备如下特殊性质的NP类问题



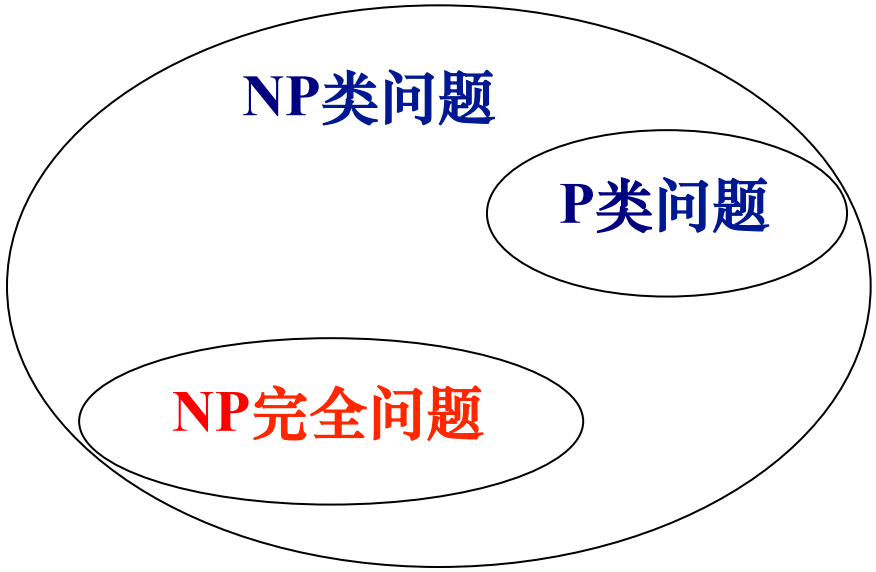
- Π（该问题本身）就是一个NP类问题
- 每一个NP类问题都可以通过多项式时间化为Π

对“NP完全问题”



- NP完全问题是NP类问题中最难的一类问题，至今已经发现了几千个，但一个也没有找到多项式时间算法。
- 如果某一个NP完全问题能在多项式时间内解决，则每一个NP完全问题都能在多项式时间内解决。
- 这些问题也许存在多项式时间算法，因为计算机科学是相对新生的科学，肯定还有新的算法设计技术有待发现；
- 这些问题也许不存在多项式时间算法，但目前缺乏足够的依据来证明这一点。

P类问题、NP类问题、NP完全问题的关系



关于P与NP关系的再思考
--从深层意义

至今没有人能证明是否P ≠ NP。

- 若P=NP，说明NP类中所有问题（包括NP完全问题）都具有多项式时间算法；

- 若P ≠ NP，说明NP类中除P类之外的所有问题（包括NP完全问题）都不存在多项式时间算法。

无论哪种答案，都将为算法设计提供重要的指导和依据。

目前人们普遍猜测：P ≠ NP

最基本的NP完全问题

- 1971年，美国的Cook证明了Cook定理：布尔表达式的可满足性(SAT)问题是NP完全的。

可满足性问题即合取范式的可满足性问题，来源于许多实际的逻辑推理的应用。合取范式形如$A_1 \wedge A_2 \wedge \ldots \wedge A_n$，其中子句$A_i$ $(1 \le i \le n)$形如：$a_1 \vee a_2 \vee \ldots \vee a_k$, 其中$a_j$称为文字，为布尔变量。SAT问题是指：是否存在一组对所有布尔变量的赋值，使得整个合取范式为真? 例如

$$f = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_3 \vee x_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_3 \vee x_4)$$

当x_1和x_3都为真、其余文字任意赋值时，f值为真。

其他NP完全问题

- 可满足性(SAT)问题是第一个被证明的NP完全问题。1972年，Karp证明了十几个问题都是NP完全的。这些NP完全问题的证明思想和技巧，以及利用他们证明的几千个NP完全问题，极大地丰富了NP完全理论。

- 已证明的NP完全问题：

  SAT问题 、最大团问题 、图着色问题 、哈密顿回路问题 、旅行商问题 、背包问题 、最长路径问题、 扫雷游戏...

如何证明一个问题是NP完全的？

- 根据NP完全问题的定义（满足的两个性质），显然地，证明需要分两个步骤进行：
  - 证明问题Π是NP类问题；即可以在多项式时间内以确定性算法验证一个任意生成的串，以确定它是否为问题的一个解。
  - 证明NP类问题中的每一个问题都能在多项式时间内变换为问题Π。由于多项式问题变换具有传递性，所以，只需证明一个已知的NP完全问题能够在多项式时间内变换为问题Π.

NP 完全证明

- 引例：旅行商问题
- 最大团问题

例子-旅行商问题

- 两个问题：
- 哈密顿回路问题：给出一个无向图G= (V，E)，它有哈密顿回路吗？即一条恰好访问每个顶点一次的回路。
- 旅行商问题：给出一个n个城市集合，且给出城市间的距离。对于一个整数k，是否存在最长为k的游程？这里，一条游程是一个回路，它恰好访问每个城市一次。

证明NP完全

- 哈密顿回路问题NP完全，证明旅行商问题也是NP完全的。
- 第一步：说明旅行商问题是NP的。
  - 一个不确定性算法可以从猜测一个城市序列开始，然后验证这个序列是一个游程。如果是的， 就继续看游程的长度是否超过给出的界k。

证明NP完全

- 第二步：证明哈密顿回路问题可以在多项式时间归约到旅行商问题，即

  哈密顿回路问题$\propto_{poly}$旅行商问题

- 即设G是哈密顿回路的任意实例， 我们构建一个含权图G‘ 和一个界限k, 使得当且仅当G′有一个总长不超过k的游程时，G有一条哈密顿回路

证明NP完全

- 设G=(V,E), G′=(V,E′)是顶点V集合的完全图，即

$$E' = \{(u,v) \mid u,v \in V\}$$

接着，我们给 E' 中的每条边指派一个长度如下

$$l(e) = \begin{cases} 1 & \text{若 } e \in E \\ n & \text{若 } e \notin E \end{cases}$$

其中 $n = |V|$。最后，我们指派 $k = n$。从构建中很容易看到，当且仅当 G' 有一个长度恰好为 n 的游程时，G 有一条哈密顿回路。要强调的是，指派 $k = n$ 属于归约的一部分。

团问题

无向图 $G=(V, E)$ 中的**团**(clique)是一个顶点子集 $V' \subseteq V$，其中每一对顶点之间都由 E 中的一条边来连接。换句话说，一个团是 G 的一个完全子图。团的**规模**是指它所包含的顶点数。**团问题**是关于寻找图中规模最大的团的最优化问题。作为判定问题，我们仅仅要考虑的是：在图中是否存在一个给定规模为 k 的团？其形式定义为：

$$\text{CLIQUE}=\{\langle G, k\rangle : G \text{ 是一个包含规模为 } k \text{ 的团的图}\}$$

团问题

团问题是 NP 完全的

- 这是一个NP问题

为了证明 CLIQUE∈NP，对一个给定的图 $G=(V, E)$，用团中顶点集 $V' \subseteq V$ 作为 G 的一个证书。对于任意一对顶点 u，$v \in V'$，通过检查边 (u, v) 是否属于 E，就可以在多项式时间内确定 V' 是否是团。

团问题

团问题是 NP 完全的

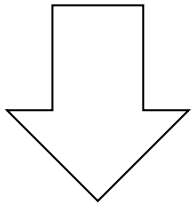
- 3合取范式可以归约为团问题

$$3\text{-CNF-SAT} \leqslant_P \text{CLIQUE}$$

团问题

- 3合取范式归约为团问题

设 $\phi=C_1 \wedge C_2 \wedge \cdots \wedge C_k$ 是 3-CNF 形式中一个具有 k 个子句的布尔公式，对 $r=1, 2, \cdots, k$，每个子句 C_r 中恰好有 3 个不同的文字 l_1、l_2 和 l_3

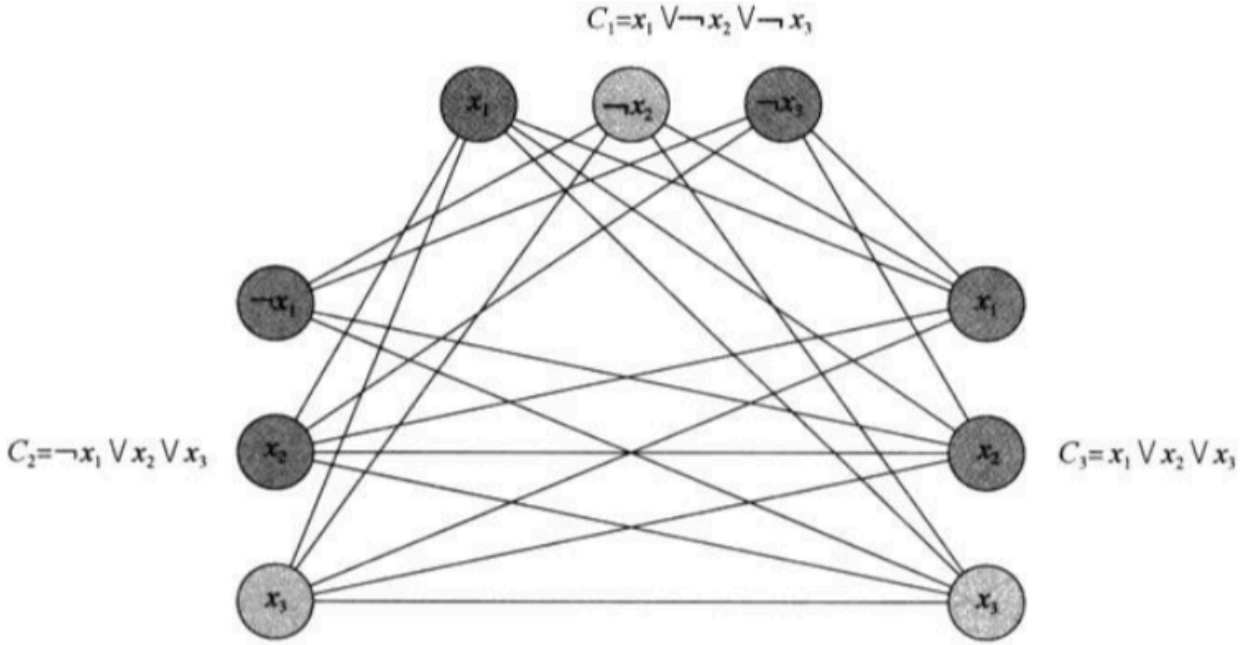


构造图

- 一个子句对应一组顶点
- 对于任意两个在不同组的顶点，如果满足“这两个顶点不是‘否’的关系”这一条件，就用一条边连接

团问题

$$\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$



$x_2 = 0, x_3 = 1$，x_1 为 0 或者 1　　是一组可满足赋值，其对应图中的灰色团

团问题

下面证明以上的变换是一个归约变换:

(1) 令 ϕ 是可满足的，即它有一个可满足的真值指派。ϕ 中每个子句至少有一个文字被指派 1。每个这样的文字对应有一个顶点。从每个子句中选出一个这样的文字所对应的顶点形成一个规模为 k 的顶点子集 V'，由 G 的构造，V' 是 G 中规模为 k 的团。

(2) 设 G 有一个规模为 k 的团。由于每个子句所对应的三个顶点彼此间无边连结。因此 k 个顶点分别来自 k 个子句所对应的三元组。因此这 k 个顶点多对应的文字是彼此相容的，对这些文字赋值 1，容易得到 ϕ 的一组可满足赋值。综上所述，ϕ 可满足 $\Leftrightarrow$ G 有一个规模为 k 的团。

团问题

- 还有一个疑问：归约为一个特殊的图，能说明一般图的团问题也是NP完全的吗？

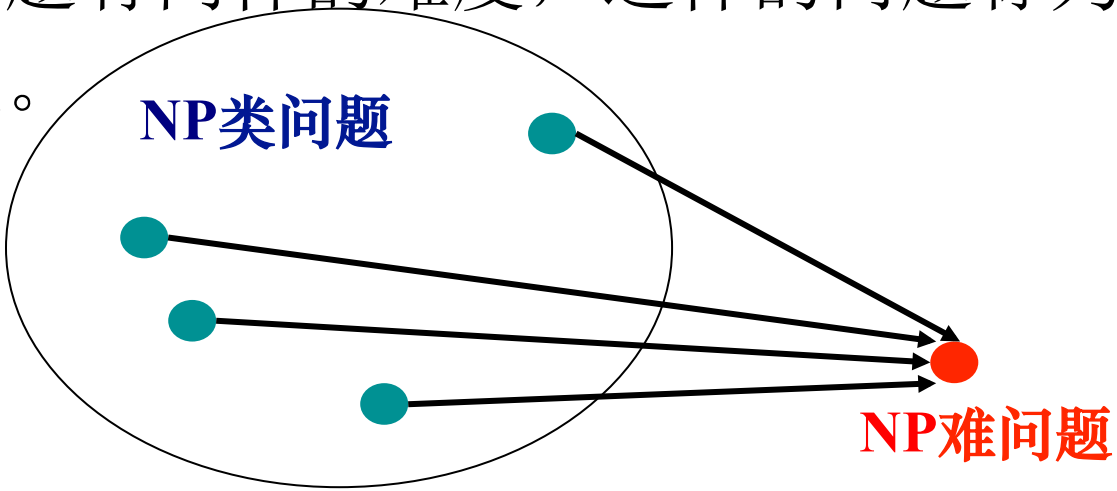
事实上，我们的确仅证明了 CLIQUE 在这种受限的情况下才是 NP 难度的，但是，这一证明足以证明在一般的图中，CLIQUE 也是 NP 难度的。这是为什么呢？如果有一个能在一般的图上解决 CLIQUE 问题的多项式时间算法，那么它就能在受限的图上解决 CLIQUE 问题。

更多NP完全问题

- 3着色问题
- 3维匹配问题
- 哈密尔顿路径问题
- 划分问题
- 背包问题
- 装箱问题
- 集合覆盖
- 多处理机调度
- 最长路径问题

NP难问题

难解问题中还有一类问题，虽然也能证明所有的NP类问题可以在多项式时间内变换到问题Π，但并不能证明Π也是NP类问题，所以Π不是NP完全的。但问题Π至少与任意NP类问题有同样的难度，这样的问题称为NP难问题。



co-NP类问题

co-NP类由它们的补属于NP类的那些问题组成。例如：

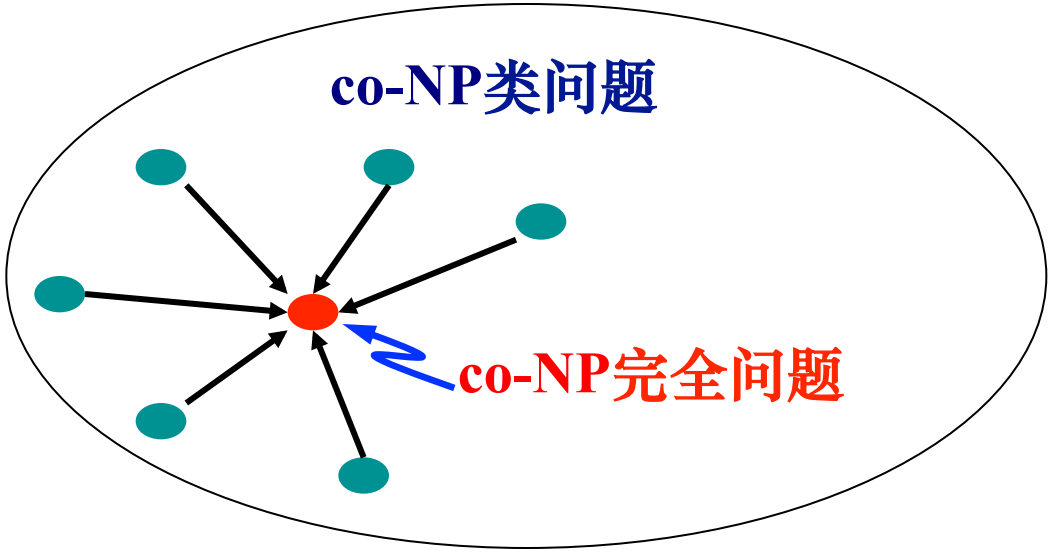
- 旅行商问题的补：给出n个城市和它们之间的距离，不存在长度为k或更少的任何旅程，情况是那样吗?
- 可满足性问题(SAT)的补：给出一个公式f，不存在使得f为真的布尔变量指派，是吗? 换言之，f是不可满足的吗?

换个角度来看， co-NP类问题也是NP完全问题。

co-NP类问题的定义

定义(co-NP完全问题): 问题Π对于co-NP类是完全的，若

- Π在co-NP中；
- 对于co-NP中的每个问题Π’，有Π’ $\propto_{poly}$ Π。



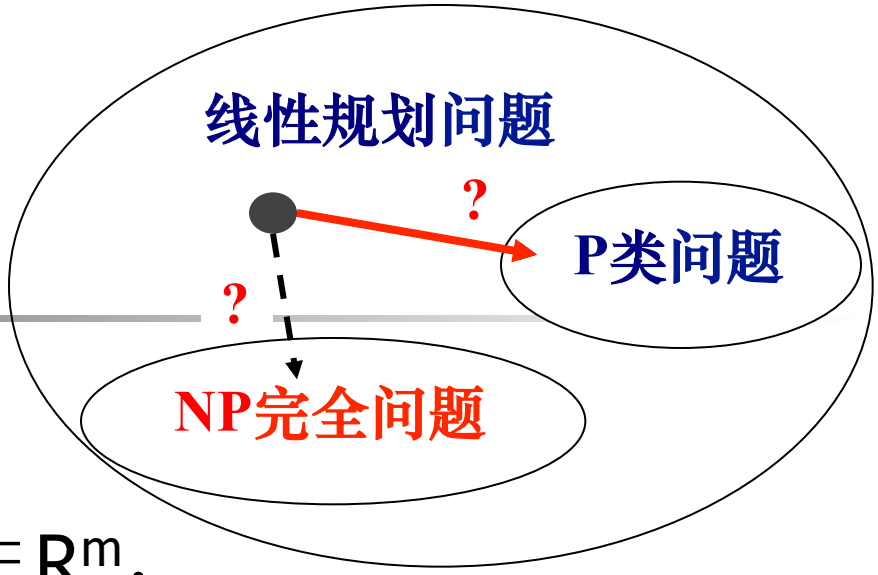
NPI类问题

- NP类问题中还有一些问题，人们不知道是属于P类还是属于NP完全问题，还有待于证明其归属。
- 这些问题是NP完全问题的可能性非常小，也因为不相信他们在P中，我们人为地增加另一问题类来接纳这类问题，这个类称为NPI(NP-Intermediate)类。

关于NPI类问题的评述

- NPI类是一个人为定义的、动态的概念，随着人们对问题研究的深入，许多NPI类问题逐渐被明白无误地证明他们原本属于P类问题或NP完全问题。
- 例如：线性规划问题、素数判定问题等，在二者没有被证明他们均属于P类问题之前，人们一直将他们归于NPI类问题。

一个例子



- 线性规划问题

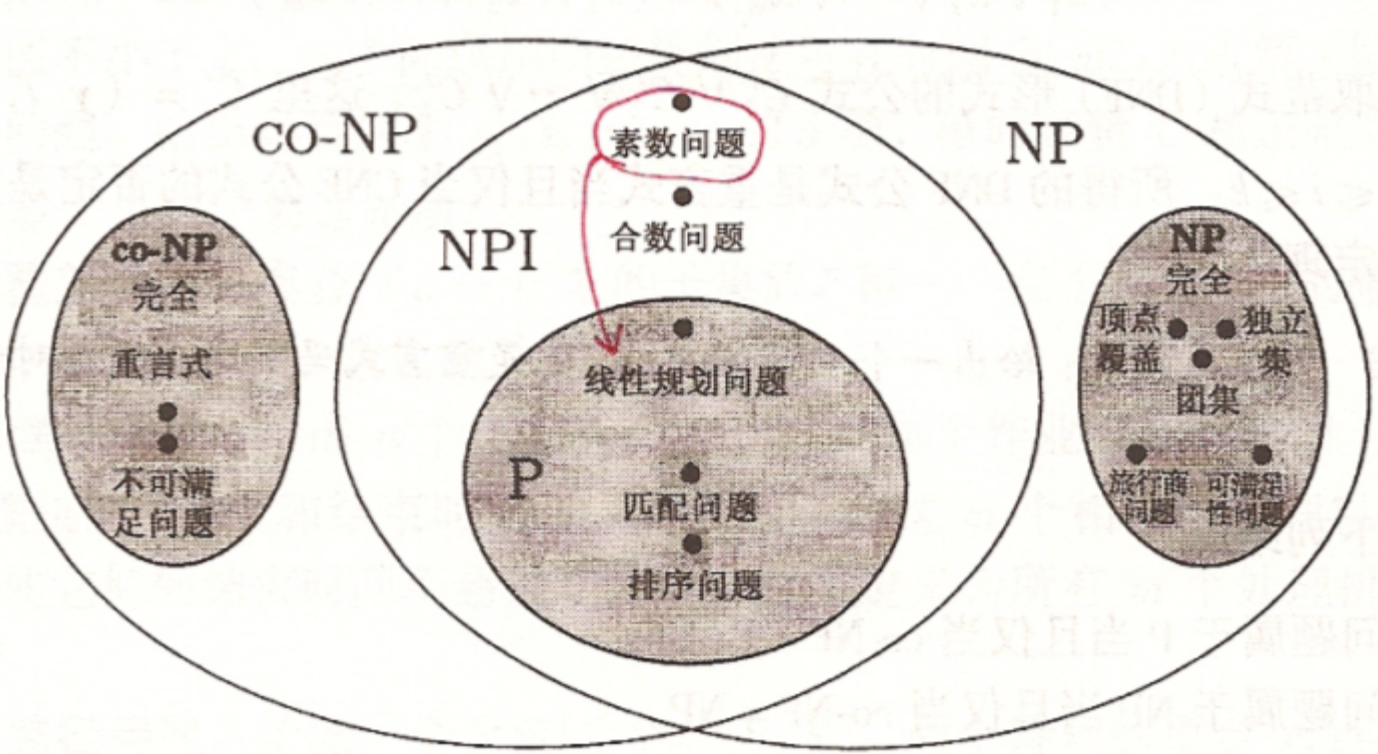
  设$A \in R^{m \times n}$, $x \in R^n$, $b \in R^m$,

  在满足$Ax=b$，$x \geq 0$约束下，

  使目标函数$c^T x$达到最大值，其中$c \in R^n$.

- 长期以来，线性规划问题没有多项式时间解法，也无法证明它是NP完全问题。直到20世纪80年代，这个问题得到解决，发现了多项式时间算法。

P、NP、co-NP、NPI类之间的区别与联系



NP完全问题的计算机处理技术

NP完全问题是计算机难以处理的，但是实际中经常遇 到，我们无法回避这些问题。因此，人们提出了解决NP完全问题的各种方法：

- 采用先进的算法设计技术
  - 问题规模不是很大时，采用动态规划法、回溯法、分枝限界法等。

NP完全问题的计算机处理技术

- 近似算法
  - 很多问题的解允许有一定的误差，只要给出的解是合理的、可接受的。
- 随机算法
  - 例如随机采样算法，穷举+挑一 vs. 随机化采样，$\Theta(n)$
  - 或以较小的错误概率为代价，获得计算时间的大幅减少。

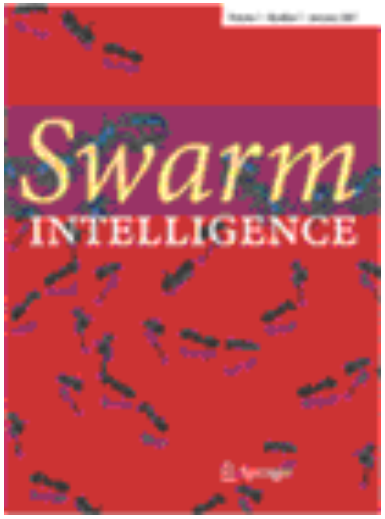
NP完全问题的计算机处理技术

- 并行计算
  - 利用多台CPU共同完成一项计算，虽然从原理上说，增强计算性能并不能从根本上解决NP完全问题，但这也是解决NP完全问题的措施之一。近年来的成就如129位(bit)大整数的分解、“深蓝”下棋程序等。

NP完全问题的计算机处理技术

智能算法

- 遗传算法
- 人工神经网络
- 模拟退火算法
- 禁忌搜索算法
- DNA计算
- 人工免疫算法
- 蚁群算法



Swarm Intelligence: a new journal dedicated to reporting on developments in the discipline of swarm intelligence. Published by Springer. **Editor-in-Chief: Marco Dorigo.**

（蚁群算法小游戏）_http://www.atlas-zone.com/complex/temp/ant/ant.htm

NP完全问题的求解

- 减少搜索量

  简单算法是穷举搜索，时间为指数

  减少搜索量：分枝限界法，隐枚举法、动态规划等

  可以提高效率，但时间复杂度不变

- 优化问题

  降低优化要求，求近似解，以得到一个多项式时间的算法。即：找寻在容许的时间内得到容许精度的近似最优解的算法

NP-完全性理论

- Cook的贡献：第一个NPC问题

史提芬·库克(Stephen Arthur Cook，1939—)NP完全性理论的奠基人，他在1971年论文”The Complexity of Theorem Proving Procedures”中，给出了第一个NP完备的证明，即Cook定理： 可满足性问题（Satisfiablity problem） 是NP完全问题，亦即 SAT∈NPC。且证明了：

SAT∈P当且仅当P=NP

Cook于1961年获Michigan大学学士学位，1962和1966年分获哈佛大学硕士与博士学位。1966-1970，他在UC Berkeley担任助教授；1970年加盟多伦多大学，现为该校CS和数学系教授，他的论文开启了NP完备性的研究，令该领域于之后的十年成为计算机科学中最活跃和重要的研究。因其在计算复杂性理论方面的贡献，尤其是在奠定NP完全性理论基础上的突出贡献而荣获1982年度的图灵奖。

Karp的贡献

理查德·卡普（Richard Karp，1935-）1972年论文”Reducibility among Combinatorial Problems”发展和加强了由库克提出的“NP完全性”理论。尤其是库克仅证明了命题演算的可满足问题是NP完全的，而卡普则证明了从组合优化中引出的大多数经典问题(背包问题、覆盖问题、匹配问题、分区问题、路径问题、调度问题等)都是NP完全问题。只要证明其中任一个问题是属于P类的，就可解决计算复杂性理论中最大的一个难题，即P=?NP。

Karp于1955、1956和1959年分别获哈佛大学文学学士、理学硕士和应用数学博士学位，现任UC Berkeley计算机科学讲座教授，美国科学院、美国工程院、美国艺术与科学院、欧洲科学院院士。因其在计算机科学领域的基础贡献曾获图灵奖(1985)、冯诺依曼奖、美国国家科学勋章、哈佛大学百年奖章等奖项.