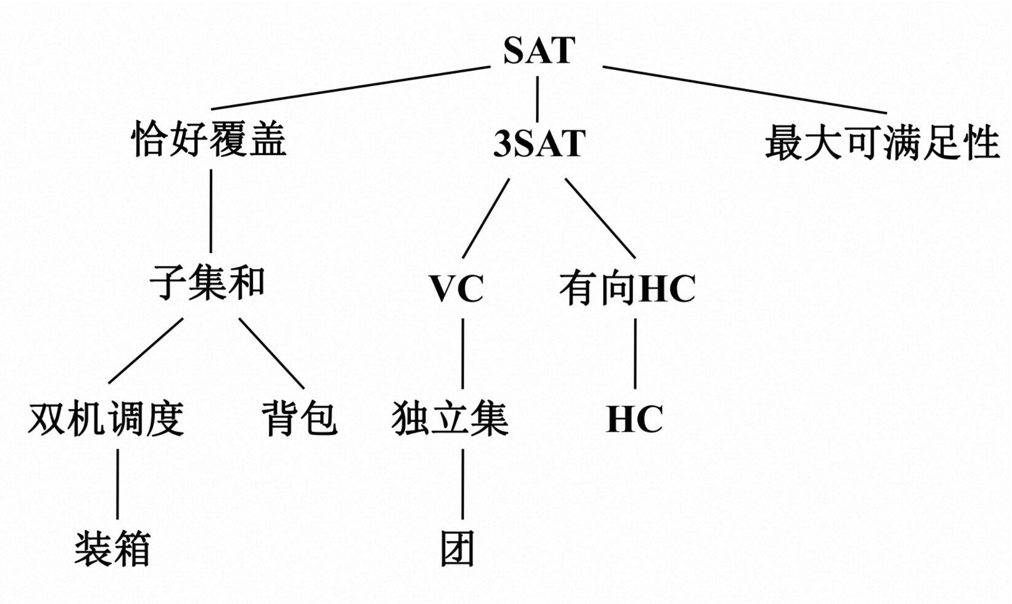
NP-complete

课上内容

- Cook-Levin定理: SAT问题是NP完全的
- 详细证明可参考计算理论导引 P304.



Karp’s 21 NP-complete problems

Satisfiability: the boolean satisfiability problem for formulas in conjunctive normal form (often referred to as SAT)

0–1 integer programming (A variation in which only the restrictions must be satisfied, with no optimization)

Clique (see also independent set problem)

Set packing

Vertex cover

Set covering

Feedback node set

Feedback arc set

Directed Hamilton circuit (Karp's name, now usually called **Directed Hamiltonian cycle**)

Undirected Hamilton circuit (Karp's name, now usually called **Undirected Hamiltonian cycle**)

Karp’s 21 NP-complete problems

Satisfiability with at most 3 literals per clause (equivalent to 3-SAT)

Chromatic number (also called the Graph Coloring Problem)

Clique cover

Exact cover

Hitting set

Steiner tree

3-dimensional matching

Knapsack (Karp's definition of Knapsack is closer to Subset sum)

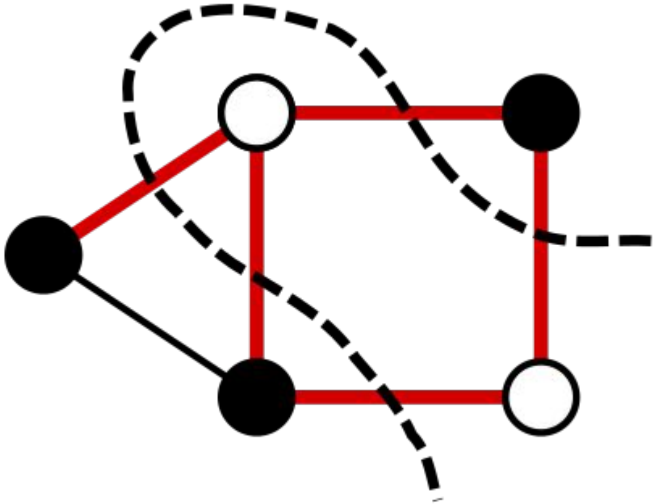
Job sequencing

Partition

Max cut

Max Cut最大割问题

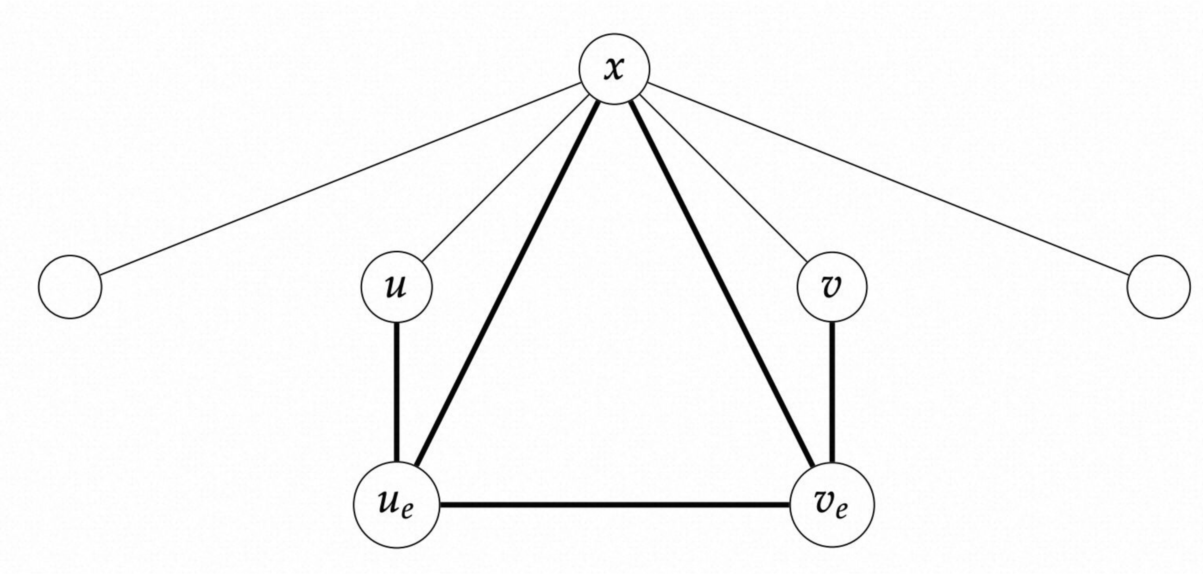
- 将图G中的节点分为S, T两个互补集合，使得ST之间的边的数量最大
- 该问题为NP-hard，其decision problem为NP-complete



Max Cut最大割 证明思路

- 考虑规约到最大独立集问题。
- 令G = (V, E)代表最大独立集问题中的图
- 构造G’ = (V’, E’):
    - 加入节点x与V中所有点连边
    - 对于e = {u, v}属于E, 创建ue, ve新节点，并加入{u, ue}, {v, ve}, {ue, ve}, {x, ue}, {x, ve}

Max Cut最大割 证明思路



构件法

Max Cut最大割 证明思路

- 证明独立集问题存在 I >= k 当且仅当 存在割 Cut(S) >= k + 4*|E|
- 具体讨论参考Max Cut NP-complete.

往年题 - 2022

由于疫情原因，小明所在校区被管控，所以小明每天都只能在校园里活动。按照要求，小明每天都要去学校门口做核酸检测。为了尽可能利用做核酸的机会多活动，小明想设计一条从小明所住的宿舍楼到核酸检测点的路线，使得路线的长度最长，并且不要走重复的地方。假定学校里的地图可以假设为一张无向简单图 G，边的权重代表长度，小明的宿舍楼和核酸检测点是其中的两个顶点。请问这个问题是否是 NP 难问题？并给出证明。

（提示：证明中可以使用的 NPC 问题是：顶点覆盖(VC)、哈密尔顿回路(HC)、团、SAT、多机调度）

往年题 - 2022

参考答案：

可以考虑从哈密尔顿回路(HC)问题构造一个到题目中问题的归约

对于图 G = <V, E>，将 V 中编号为 1 的点拆成入点 S 、中间点 M 和出点 T，将 E 中与 1 相关的边拆成到 S 和 T 分别连边，并连边 <S, M>, <M, T>,新图中所有的边边权为 1，则原图存在哈密尔顿回路等价于新图中 S 到 T 的最长简单路径长度为 n = |V|。

因此题目中问题是 NP-hard 问题。

往年题 - 2020

九、NP 完全性（共 10 分）

证明最小平方和问题是 NP 完全的。

输入：一个有n个整数的集合A、待划分的子集个数K、以及整数L。

输出：能否将A划分成K个不相交的子集$A_1, A_2, \ldots, A_K$且$A = \bigcup_{i=1}^{K} A_i$，使得$\sum_{i=1}^{K}\left(\sum_{a \in A_i} a\right)^2 \leq L$。

往年题 - 2020

参考答案：

先证明该问题是NP的；

从子集和问题归约，令K’=2，L‘ = L*L + (Sum-L) * (Sum - L)，就完成了归约；

因此这个问题是NPC的。

往年题 - 2019

六、NPC 证明（15）

证明 Monotone SAT 是 NPC 问题。

Monotone SAT：F 的所有析取范式中只有 positive 项（没有取否操作），问是否存在一个最多有 k 个 True 的成真赋值。

往年题 - 2019

参考答案：

首先证明是NP问题；

其次构造一个从SAT到原问题的多项式规约，我们引入一个新变量z_i来代表 $\neg x_i$。对于给定的公式ϕ，我们通过将每个$\neg x_i$的出现替换为z_i，并添加$x_i \vee z_i$的子句来创建一个新的公式ϕ'。

令 K 的值为原 SAT 中变量的总数，则构造的 mono SAT 可满足当且仅当原 SAT 可满足。即完成了从 SAT 到 mono SAT 的归约。

往年题 - 2018

得分

六、NPC 证明（15 分）

给定图$G=(V, E)$ 和整数k。如果任意两个结点$v, u \in I$，边$(v, u) \notin E$，并且也没有从v到u的两条边的路径，即没有结点w使得$(v, w) \in E \wedge (w, u) \in E$，则称集合$I \subseteq V$是强独立的。强独立集问题是要确定$G$是否有一个大小不小于$k$的强独立集。

1. （5 分）请证明强独立集是 NP。
2. （10 分）请证明强独立集是 NP 难。（提示：可以利用下图中的 NPC 问题进行证明）

往年题 - 2018

参考答案：

1. 任意给定一个强独立集猜想，解释验证这个猜想只需要多项式时间，得满分。如果没有说明多项式时间，扣 2 分。
2. 证明一：说明独立集是强独立集的子问题，得满分。

   证明二：

利用独立集问题进行证明。

任给一个独立集问题的实例图 G，在每条边的中间加一个顶点 u_i，所有新加的顶点之间都连边，{u_i}形成完全图。然后证明在新的图 G’ 中，一个强独立集不能同时选择两个 u’，因为任何两个 u’ 有边相连。也不能同时选择一个新顶点和一个旧顶点，因为任意一个新顶点距离任意一个旧顶点距离为 2，所以只能都选旧顶点。新图中只由旧顶点构成的强独立集等价于旧图中的独立集。

往年题 - 2018

证明三：利用独立集问题进行证明。

任给一个独立集问题的实例图 G，在每条边的中间加一个顶点 u_i，所有新加的顶点之间都和一个公共顶点 s 相连，并且 s 与另外一个顶点 s’相连。也可以证明二者的等价性。和证明二类似。

往年题 - 2016

7.NPC 证明：　图 G 中取 a 个点，至少有 b 条边两个端点都在这 a 个点中

往年题 2016

参考答案：

先证明原问题是NP的；

接下来构造一个从独立集问题到原问题的多项式时间归约，令 $b = 0$ 即可；

因此原问题是NPC的。

一些选择 / 判断题

1. 如果 P=NP，那么每一个 NPC 问题都可以在多项式时间内求解。(✓)

2. NP-hard 问题一定是 NPC 问题。(×)

一些选择 / 判断题

6. 0-1 背包问题可以用动态规划方法求解，因此应该不属 NPC 类问题。(×)

一些选择 / 判断题

5. 以下问题中属于 NPC 的是（ B ）。

A. 最长公共子序列 B. 双机调度 C. 最小生成树 D. 快速排序

补充内容 - Co-NP

Co-NP问题的定义：

如果一个语言 L 的补集 ¬L 属于NP，则称L属于Co-NP。

另一种定义方式：

语言L属于Co-NP当且仅当：

$\exists$ a polynomial function $P : N \rightarrow N$ and a polynomial time TM M s.t. $\forall x \in \{0,1\}^*$

$$x \in L \quad \text{iff} \quad \forall u \in \{0,1\}^{P(|x|)} \text{ s.t. } M(x,u) = 0$$

Co-NP不是NP的补集！

补充内容 - Co-NP

Co-NP语言的一个例子是 ¬SAT，这个语言由所有不可满足的布尔表达式组成。

一个布尔表达式被称为不可满足的，当且仅当没有任何一组赋值使得这个布尔表达式的值为真。

补充内容 - Co-NP

我们无法用证明 SAT 是 NP 的的方法证明 ¬SAT 也是 NP 的，这是因为：

如果给出一组赋值，我们确实可以在多项式时间内验证这组赋值使得某布尔表达式为假，但是这样并不能说明不存在一组其他的赋值使得该布尔表达式为真。实际上，按照这种赋值+验证的想法，我们可能需要穷举所有的赋值，才可以说明一个布尔表达式是不可满足的，这个验证需要的时间显然超出了多项式时间的范畴。

可以证明：¬SAT问题是Co-NP complete的，即所有的 Co-NP中最复杂的

补充内容 - Co-NP

通过 ¬SAT 的例子，我们看似说明了 Co-NP 和 NP 是两个不同的复杂度类，但实际上 NP 是否等于 Co-NP 还是一个open problem，就如同 P 和 NP 的关系一样。

目前我们已知的有 $P \subseteq NP \cap \text{Co-NP}$