



HITWH
SE

第九章

NP完全性与近似算法



- 时间复杂度被分为两种级别：
 - 一种是 $O(1)$, $O(\log(n))$, $O(n^a)$ 等，我们把它叫做多项式级的复杂度，因为它的规模 n 出现在底数的位置；
 - 另一种是 $O(a^n)$ 和 $O(n!)$ 型复杂度，它是非多项式级的，其复杂度计算机往往不能承受。
- 当我们在解决一个问题时，我们选择的算法通常都需要是多项式级的复杂度，非多项式级的复杂度需要的时间太多，往往会超时，除非是数据规模非常小



- 自然地，人们会想到一个问题：会不会所有的问题都可以找到复杂度为多项式级的算法呢？
- 答案是否定的。
- 例如：
 - Hamilton回路。
 - 问题是这样的：给你一个图，问你能否找到一条经过每个顶点一次且恰好一次(不遗漏也不重复)最后又走回来的路(Hamilton回路)。
 - 这个问题现在还没有找到多项式级的算法。事实上，这个问题就是我们后面要说的NPC问题。



HITWH
SE

提要

9.1 NP完全性

9.2 近似算法的基本概念与设计方法



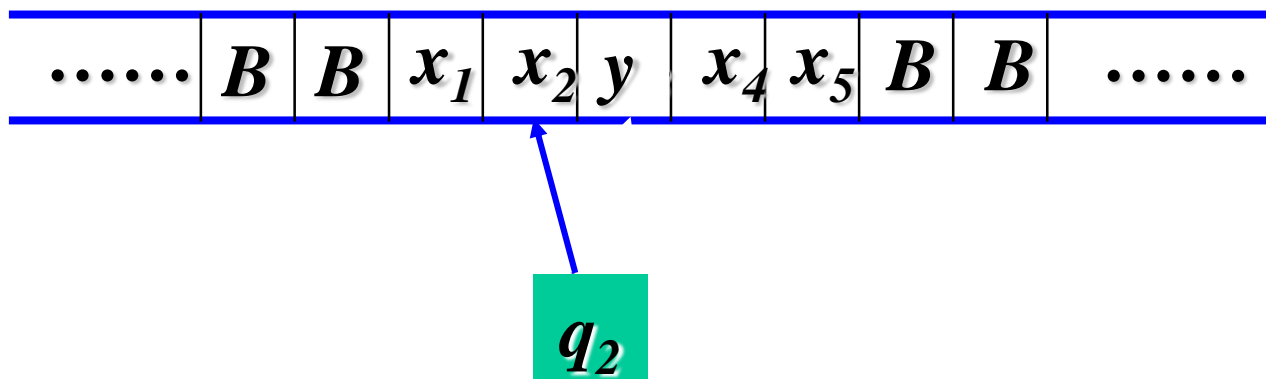
NP完全性

- NP完全问题的概念
- 几个重要的NP完全问题



NP 完全性相关概念

- 确定图灵机



$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q : 有穷状态集合

Σ : 有穷输入符号集合

Γ : 有穷带符号集合, $\Sigma \subseteq \Gamma$

$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, L\}$, $\delta(q, x) = (p, y, D)$, $D = L$ 或 R

q_0 : 起始状态

$B \in \Gamma - \Sigma$: 空白符号

$F \subseteq Q$: 终止或接受状态集合



NP 完全性相关概念

- 图灵机接受的瞬时描述
 - $X_1X_2...X_{i-1}qX_iX_{i+1}...X_n$
- 瞬时描述的迁移
 - $X_1X_2...X_{i-1}qX_iX_{i+1}...X_n \Rightarrow X_1X_2...pX_{i-1}YX_{i+1}...X_n$
 - $\alpha_1q\beta_1 \Rightarrow^* \alpha_2p\beta_2$
- 图灵机作为语言识别器
 - $M=(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ 接受的语言为
 $L(M)=\{ w \mid w \in \Sigma^*, q_0w \Rightarrow^* \alpha p \beta, p \in F \}$
 - 例1. 构造图灵机接受 $\{0^n1^m \mid n \geq 1, m \geq 1\}$
 - 例2. 构造图灵机接受 $\{0^n1^n \mid n \geq 1, m \geq 1\}$
- 图灵机作为计算器



NP 完全性相关概念

- 不确定图灵机的定义

- $M=(Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q : 有穷状态集合

Σ : 有穷输入符号集合

Γ : 有穷带符号集合, $\Sigma \subseteq \Gamma$

~~$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, L\}, \delta(q, x) = (p, Y, D), D=L \text{ 或 } R$~~

q_0 : 起始状态

$B \in \Gamma - \Sigma$: 空白符号

$F \subseteq Q$: 终止或接受状态集合

$\delta: \delta(q, x) = \{ (p, Y, D) \mid p \in Q, Y \in \Gamma, D \in \{L, R\} \}$



NP 完全性相关概念

- 不确定图灵机作为语言识别器
 - $M=(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ 接受的语言为
 $L(M)=\{ w \mid w \in \Sigma^*, \text{ 存在一个移动序列 } q_0 w \Rightarrow^* \alpha p \beta, p \in F \}$
- 不确定图灵机作为计算器
- 确定图灵机与非确定图灵机区别
 - 确定性图灵机：每一步只有一种选择。
 - 非确定图灵机：每一步可以有多种选择



HITWH
SE

NP 完全性相关概念

- 图灵机的扩展
 - 多道图灵机
 - 多维图灵机
 - 多带图灵机
 - 具有半无限带的图灵机
 -



NP 完全性相关概念

- 各种图灵机的计算等价性

定理1. 所有前面介绍的图灵机计算能力都是等价的。

- 确定图灵机与计算机的等价性

定理2. 确定图灵机与计算机是等价的，而且在多项式时间内等价。



NP 完全性相关概念

- P类问题
 - 确定图灵机在多项式时间内能够求解的全部问题
 - 排序、矩阵链乘、最小生成树等
- NP类问题
 - 非确定图灵机在多项式时间内能够求解的全部问题
 - 旅行商问题、团问题等
- 显然的结论: $P \subseteq NP$
- 悬而未决的问题: $P \stackrel{?}{=} NP$

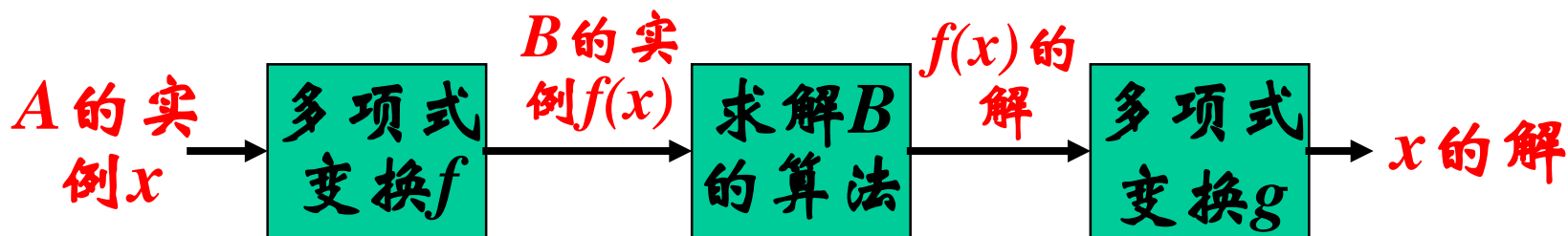


NP 完全性相关概念

- 多项式时间规约

— 问题A在多项式时间内可归约为问题B, 如果存在具有多项式时间复杂性的变换 f 和 g 使得:

- x 是A的实例 iff $f(x)$ 是B的实例;
- s 是B实例 $f(x)$ 的解 iff $g(s)$ 是A实例 x 的解.





NP 完全性相关概念

- 多项式时间规约的应用

定理. 设问题A可以在多项式时间内归约为问题B. 如果 $B \in P$, 则 $A \in P$.

推论. 设问题A可以在多项式时间内归约为问题B. 如果 $A \notin P$, 则 $B \notin P$.



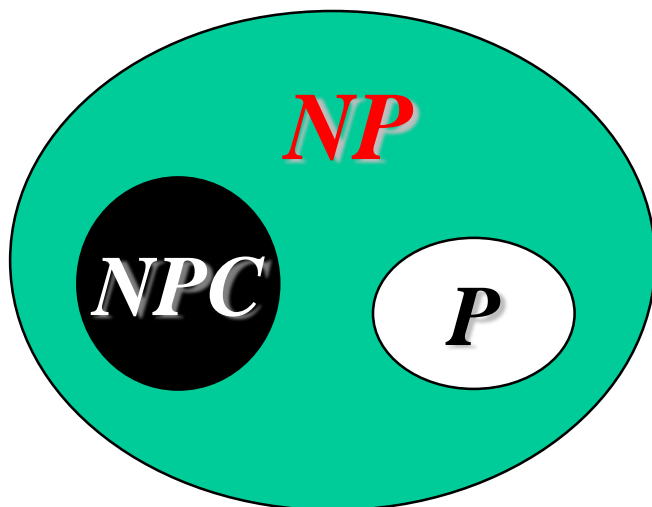
NP 完全性相关概念

- NP-Complete 问题

问题 A 称为 NP-Complete, 如果

- $A \in NP$,
- $\forall B \in NP$, B 可以多项式地归约为 A。

* 似乎任意一个 NP-complete 问题都属于 NP-P?





NP 完全性相关概念

- NP-Complete 问题

问题 A 称为 NP-Complete, 如果

- $A \in \text{NP}$,
- $\forall B \in \text{NP}$, B 可以多项式地归约为 A 。

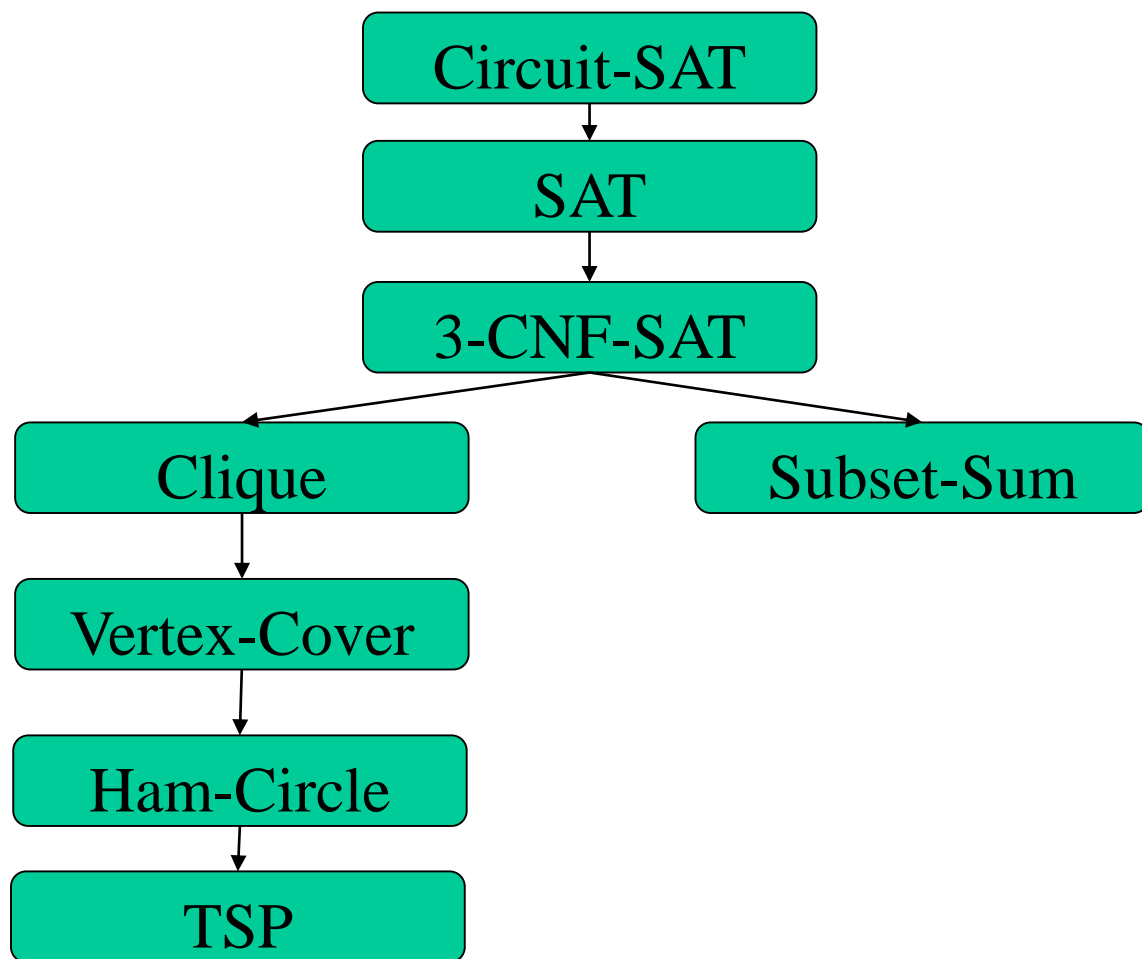
定理. 如果 p_1 是 NP-Complete, $p_2 \in \text{NP}$, 而且 p_1 可以在多项式时间内归约到 p_2 , 则 p_2 是 NP-Complete.

定理. 如果一个 NP-Complete 问题 $p \in P$, 则 $P = \text{NP}$.



HITWH
SE

NP 完全性相关概念



几个经典NP完全问题的规约路线



NP 完全性相关概念

- NP-Hard 问题

问题A称为NP-Hard, 如果

– $\forall B \in \text{NP}$, B可以在多项式时间内归约为A.

*注意:

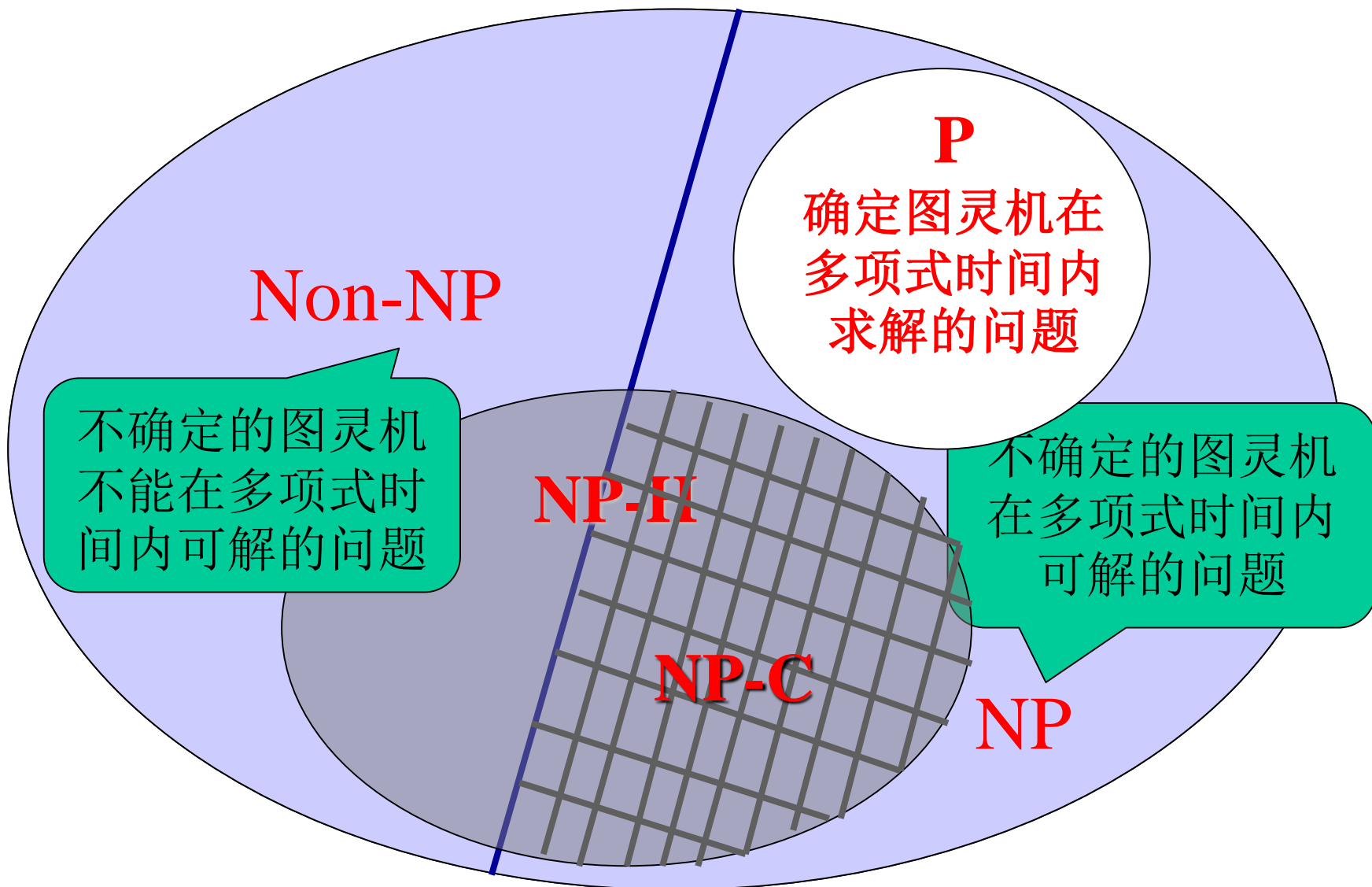
➤ NP-难问题不一定属于NP问题

➤ NP-难问题似乎比NP-完全问题更难



HITWH
SE

NP 完全性相关概念





HITWH
SE

提要

近似算法的基本概念与设计方法



近似算法的基本概念

- 近似算法的基本思想
 - 很多实际应用中问题都是NP-完全问题
 - NP-完全问题的多项式算法是难以得到的
 - 求解NP-完全问题的方法：
 - 如果问题的输入很小,可以使用指数级算法圆满地解决该问题
 - 否则使用多项式算法求解问题的近似优化解
 - 什么是近似算法
 - 能够给出一个问题的近似解的算法
 - 常用来解决**优化问题**



近似算法的性能分析

- 近似算法的时间复杂性
 - 分析目标和方法与传统算法相同
- 近似算法解的近似度
 - 本节讨论的问题是**优化问题**
 - 问题的每一个可能的解都具有一个正的代价
 - 问题的优化解可能具有最大或最小代价
 - 我们希望寻找问题的一个近似优化解
 - 我们需要分析近似解代价与优化解代价的差距
 - Ratio Bound
 - 相对误差
 - $(1 \pm \varepsilon)$ -近似



• Ratio Bound

定义1(Ratio Bound) 设 A 是一个优化问题的近似算法, A 具有ratio bound $p(n)$, 如果

$$\max\left\{\frac{C}{C^*}, \frac{C^*}{C}\right\} \leq p(n)$$

其中 n 是输入大小, C 是 A 产生的近似解的代价,

C^* 是优化解的代价.

- 如果问题是最大化问题, $\max\{C/C^*, C^*/C\} = C^*/C$
- 如果问题是最小化问题, $\max\{C/C^*, C^*/C\} = C/C^*$
 - Ratio Bound不会小于1
 - Ratio Bound越大, 近似解越坏



- 相对误差

定义2(相对误差) 对于任意输入, 近似算法的相对误差定义为 $|C-C^*|/C^*$, 其中 C 是近似解的代价, C^* 是优化解的代价.

定义3(相对误差界) 一个近似算法的相对误差界为 $\varepsilon(n)$, 如果 $|C-C^*|/C^* \leq \varepsilon(n)$.



结论1. $\varepsilon(n) \leq p(n)-1$.

证. 对于最小化问题

$$\varepsilon(n) = |C - C^*| / C^* = (C - C^*) / C^* = C / C^* - 1 = p(n) - 1.$$

对于最大化问题

$$\begin{aligned} \varepsilon(n) &= |C - C^*| / C^* = (C^* - C) / C^* = (C^* / C - 1) / (C^* / C) \\ &= (p(n) - 1) / p(n) \leq p(n) - 1. \end{aligned}$$

- 对于某些问题, $\varepsilon(n)$ 和 $p(n)$ 独立于 n , 用 p 和 ε 表示之.
- 某些 NP-完全问题的近似算法满足: 当运行时间增加时, Ratio Bound 和 相对误差 将减少.
- 结论1表示, 只要求出了 Ratio Bound 就求出了 $\varepsilon(n)$



• 近似模式

定义4 (近似模式) 一个优化问题的近似模式是一个以问题实例 I 和 $\varepsilon > 0$ 为输入的**算法族**. 对于任意固定 ε , 近似模式是一个 $(1 \pm \varepsilon)$ -近似算法.

定义5 一个近似模式 $A(I, \varepsilon)$ 称为一个多项式时间近似模式(*PTAS*), 如果对于任意 $\varepsilon > 0$, $A(I, \varepsilon)$ 的运行时间是关于输入实例大小 $|I|$ 的多项式.

定义6 一个近似模式称为完全多项式时间近似模式(*FPAS, FPTAS*), 如果它的运行时间是关于 $1/\varepsilon$ 和输入实例大小 n 的多项式.



顶点覆盖问题

输入：无向图 $G=(V, E)$

输出： $C \subseteq V$, 满足

- (1). $\forall (u, v) \in E, u \in C, v \in C$ 或 $\{u, v\} \subseteq C$
- (2). C 是满足条件(1)的最小集合。

理论上已经证明：

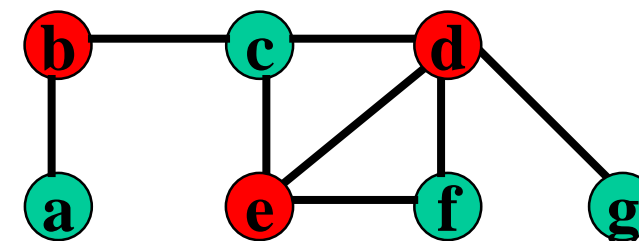
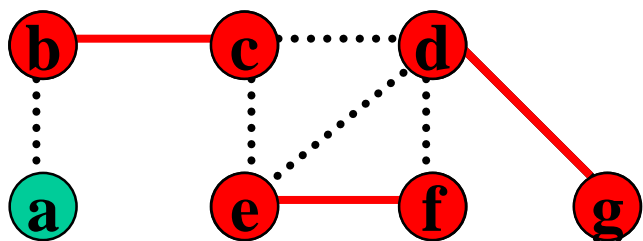
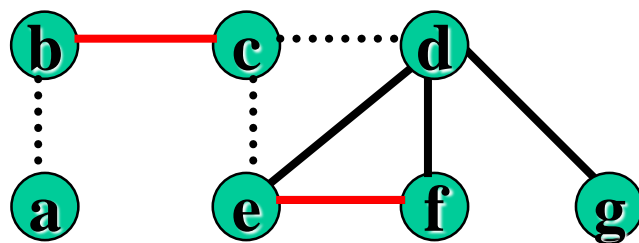
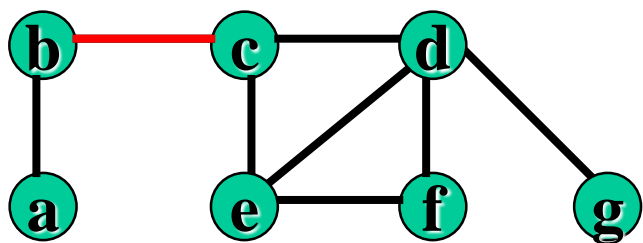
最优顶点覆盖问题是NP-完全问题。



近似算法的设计

• 算法的基本思想

每次任取一条边，结点加入C，删除相关边，重复该过程



算法解: $\{b, c, e, f, d, g\}$

优化解: $\{b, e, d\}$



- 算法

APPROX-Vertex-Cover (G)

时间复杂性

$$T(G)=O(|E|)$$

1. $C=0$
2. $E'=E[G];$
3. While $E'\neq 0$ DO
4. 任取 $(u, v)\in E';$
5. $C=C\cup\{u, v\};$
6. 从 E' 中删除所有与 u 或 v 相连的边;
7. Return C



• Ratio Bound

定理. Approx-Vertex-Cover的Ratio Bound为2.

证. 令 $A=\{(u, v) \mid (u, v) \text{ 是算法第4步选中的边}\}$.

若 $(u, v) \in A$, 则与 (u, v) 邻接的边皆从 E' 中删除.

于是, A 中无相邻接边.

第5步的每次运行增加两个结点到 C , $|C|=2|A|$.

设 C^* 是优化解, C^* 必须覆盖 A .

由于 A 中无邻接边,

C^* 至少包含 A 中每条边的一个

结点. 于是,

$$|A| \leq |C^*|,$$

$$|C|=2|A| \leq 2|C^*|,$$

$$\text{即 } |C|/|C^*| \leq 2.$$

APPROX-Vertex-Cover (G)

1. $C=\emptyset$
2. $E'=E[G]$;
3. While $E' \neq \emptyset$ DO
4. 任取 $(u, v) \in E'$;
5. $C=C \cup \{u, v\}$;
6. 从 E' 中删除所有与 u 或 v
 相连的边;
7. Return C



- 基于组合优化的近似算法
- 基于贪心策略的近似算法
- 基于局部优化的近似算法
- 基于动态规划的近似算法
- 基于线性规划的近似算法
- 随机算法、Online算法
-



其它算法设计方法

- 智能优化算法
 - 遗传算法
 - 蚁群算法、鱼群算法
 - 模拟退火算法
- 并行与分布式算法
-



HITWH
SE

各位同学的算法课没有结束.....

