第8章 计算复杂性理论

提 纲

- 易解的问题与难解的问题
- 判定问题
- NP类
- 多项式时间变换
- NP完全性及Cook-Levin定理

引言

❖计算复杂性

- 问题本身的内在复杂性决定了求解这个问题算法的计算复杂性。
- 从计算的角度如何度量问题的内在复杂性
- 许多实际问题无法确切了解其内在复杂性，只能将计算复杂性大致相同的问题归类进行研究

好算法的标准

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 $O(n2^n)$

输入规模与算法运行时间

天河二号（2015.12）

–峰值：5.49亿亿次（双精度浮点运算）

–持续：3.39亿亿次（双精度浮点运算）

Dijkstra最短路算法		最大团简单回溯算法	
顶点数	运行时间	顶点数	运行时间
1亿	300毫秒	20	1微妙
10亿	30秒	40	1毫秒
100亿	50分钟	60	1小时
1000亿	81小时	80	90年
1万亿	341天	100	1亿年

❖易解问题、难解问题

- 如果一个问题Π存在一个时间复杂性是$O(n^k)$的算法，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
- 通常将存在多项式时间算法的问题看作是易解问题，将不存在多项式时间算法的问题看作是难解问题。

多项式相关的概念

- 两个函数多项式相关
  - 设两个函数 $f, g: N \rightarrow N$，如果存在多项式p和q，使得对$\forall n \in N$

$$f(n) \leq p(g(n))$$

$$g(n) \leq q(f(n))$$

  - 则称f和g是多项式相关的

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.

问题实例的合理编码

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

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

算法运行时间是基本操作数

指令集不同，基本操作次数也会不同

指令集中的指令必须合理：　即每个指令的执行时间都应是常数

指令集间的等价性：每个指令都可以由另一个指令集中固定常数数量以内的指令模拟

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.

易解问题与难解问题……

- 不可计算问题

  丢番图方程是否有整数解，图灵机的停机问题、……

- 难解问题
  - 判定Presburger算术中的命题是否为真、……
  - 至少需要指数时间，或指数空间

- 易解问题
  - 排序、最小生成树、单源最短路等

- NPC问题（不知难还是易）
  - 既没有找到多项式时间算法、又没能证明是难解的问题.
  - 哈密尔顿回路问题、货郎问题、背包问题等

提 纲

- 易解的问题与难解的问题
- 判定问题
- NP类
- 多项式时间变换
- NP完全性及Cook-Levin定理

判定问题

- 直观概念
  - 答案只有两个“是(Yes)”和“否(No)”的问题
- 形式化定义
  - 判定问题$\boldsymbol{\Pi}$为有序对（$\boldsymbol{D_{\Pi}},\boldsymbol{Y_{\Pi}}$），其中
  - $\boldsymbol{D_{\Pi}}$是实例的集合，由$\boldsymbol{\Pi}$的所有可能的实例组成
  - $\boldsymbol{Y_{\Pi}}\subseteq\boldsymbol{D_{\Pi}}$由所有答案为“Yes ”的实例组成

排序问题

判定问题：给定一个整数数列，是否可以按升序排列

搜索问题：将一个整数数列调整为升序序列

- 如果多项式时间算法***A*能求解搜索问题**
  - 能对整数数列排序，输出排序序列
  - 不能排序， ***A***输出“No”
- 可**基于*A*构造算法*B***
  - 如果***A***输出排序序列，***B***输出“Yes”
  - 如果***A***输出“No” ， ***B***输出“No”

哈密尔顿回路（HC）

- 任给无向图G，问G是哈密尔顿图吗？
  - D_{HC}由所有的无向图组成
  - Y_{HC}包括所有哈密尔顿图
  - 只需要回答G中是否存在哈密尔顿回路，不需要给出G的具体哈密尔顿回路
- 对应的搜索问题

–为G找出一条哈密尔顿回路(如果存在)

–回答 “No” (如果不存在)

HC对应的搜索问题不比HC更容易

- 如果多项式时间算法$\boldsymbol{A}$能求解搜索问题
  - 当$\boldsymbol{G}$是哈密尔顿回路时， $\boldsymbol{A}$输出$\boldsymbol{G}$的一条哈密尔顿回路；
  - 当$\boldsymbol{G}$不是哈密尔顿回路时， $\boldsymbol{A}$输出“No”
- 可基于$\boldsymbol{A}$构造算法$\boldsymbol{B}$
  - 如果$\boldsymbol{A}$对$\boldsymbol{G}$输出的是回路， $\boldsymbol{B}$输出“Yes”
  - 如果$\boldsymbol{A}$输出“No” ， $\boldsymbol{B}$输出“No”
  - 则显然，如果$\boldsymbol{A}$多项式时间算法, $\boldsymbol{B}$也是多项式时间算法
- 于是 搜索问题是易解的⇒HC是易解的

  HC是难解的⇒其搜索问题是难解的

货郎问题(TSP)

- 任给n个城市，以及城市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？

- 对应的搜索问题

– 为G找出一条最短哈密尔顿回路(如果存在)

– 回答“No” (如果不存在)

提 纲

- 易解的问题与难解的问题
- 判定问题
- NP类
- 多项式时间变换
- NP完全性及Cook-Levin定理

P类与NP类

- P类（Polynomial）
  - 所有多项式时间可解的判定问题 组成的问题类

- NP类（Nondeterministic Polynomial）
  - 所有多项式时间可验证的判定问题 组成的问题类

多项式时间可验证

•设判定问题$\boldsymbol{\Pi}$=（$\boldsymbol{D,Y}$）

•如果存在有两个输入变量的多项式时间算法 $\boldsymbol{A}$和多项式 $\boldsymbol{p}$，对$\forall$ $\boldsymbol{I}\in\boldsymbol{D}$，

•$\boldsymbol{I} \in \boldsymbol{Y}$ 当且仅当 $\exists \boldsymbol{t}$，$\boldsymbol{t}\leq\boldsymbol{p}(|\boldsymbol{I}|)$，$\boldsymbol{A}$对输入 $\boldsymbol{I}$和 $\boldsymbol{t}$输出“Yes”，

•则称$\boldsymbol{\Pi}$是多项式时间可验证的

•$\boldsymbol{A}$是$\boldsymbol{\Pi}$的多项式时间验证算法

•而当$\boldsymbol{I}\in\boldsymbol{Y}$时，称t是$\boldsymbol{I}\in\boldsymbol{Y}$的证据

NP-非确定性多项式时间算法

•把多项式时间验证算法看成一种以不确定的方式 搜索整个证据空间：

•对给定实例$I \in D$，首先“猜想”一个t， $t \le p(|I|)$

•然后检查t是否是证明$I \in Y$的证据

•猜想和验证都可在多项式时间完成

•$I \in Y \Leftrightarrow$能够正确地猜想到一个证据t

这种不确定的搜索方式称作非确定性多项式时间算法

哈密尔顿回路(HC)的NP算法

- 对于给定的图G
  - –任意猜想一个所有顶点的排列
  - –检查这个排列是否构成一条哈密尔顿回路 即任意相邻两点及首尾两点间是否都有边
  - –若是，回答“Yes”
  - –否则回答“No”
- 上述猜想和验证都可在多项式时间内完成
- 图G可验证$\Leftrightarrow$图G存在哈密尔顿回路
- 因此 HC $\in$ NP

0-1背包问题∈ NP

- 给定n个物品和一个背包
- 物品i的重量为w_i，价值为v_i，$1 \le i \le n$
- 背包的重量限制为B，目标价值为K
- 问能否在背包中装入总价值不小于K，总重量不超过B 的物品吗？
- 即是否存在子集$T \subseteq \{1,\ldots,n\}$，使得 $\sum_{i \in T} v_i \ge K$ 并且 $\sum_{i \in T} w_i \le B$
- 任意猜想一个子集T并在多项式时间内验证上式
- 成立则回答“Yes”，否则回答“No”
- 这是0-1背包问题的多项式时间验证算法

定理 P ⊆ NP

证明：设Π=（D,Y），A是Π的多项式时间算法；

构造算法B，对于每一个$I \in D$和任意的t，

算法B对于I和t与算法A对于I的计算过程完全相同，

显然B是多项式时间的算法，且

$I \in Y$时，A回答“Yes”，取$t=t_0$，B也回答“Yes”

$I \notin Y$时，A回答“No”，无论怎样取t，B总回答“No”

所以B是Π的多项式时间验证算法。因此

$\Pi \in \mathrm{P} \Rightarrow \Pi \in \mathrm{NP}$

那么问题是 **P=NP？**

如何比较两个问题的难度？

- P=NP？
  - 很可能是不成立的，但如何证明不成立？
  - NP中最难的问题应该是难解的
  - 怎么知道一个问题是NP中最难的问题？
  - 多项式时间变换用于比较问题的难度

提 纲

- 易解的问题与难解的问题
- 判定问题
- NP类
- 多项式时间变换
- NP完全性及Cook-Levin定理

多项式时间变换

定义 设判定问题$\boldsymbol{\Pi_1}$=（$\boldsymbol{D_1,Y_1}$），$\boldsymbol{\Pi_2}$=（$\boldsymbol{D_2,Y_2}$），如果存在函数$\boldsymbol{f:D_1 \to D_2}$满足条件:

① $\boldsymbol{f}$ 是多项式时间可计算的

②对所有的$\boldsymbol{I \in D_1}$， $\boldsymbol{I \in Y_1 \Leftrightarrow f}$（$\boldsymbol{I}$）$\boldsymbol{\in Y_2}$（正确性）

则称 $\boldsymbol{f}$ 是$\boldsymbol{\Pi_1}$到$\boldsymbol{\Pi_2}$的多项式时间变换。

- 如果存在$\boldsymbol{\Pi_1}$到$\boldsymbol{\Pi_2}$的多项式时间变换，

  则称$\boldsymbol{\Pi_1}$可多项式时间变换到$\boldsymbol{\Pi_2}$，

  记作$\boldsymbol{\Pi_1 \le_p \Pi_2}$。

例：HC $\leq_p$ TSP

证：设计HC到TSP的多项式时间变换$\boldsymbol{f}$：

对HC的每个实例定义$\boldsymbol{I}$， $\boldsymbol{I}$是一个无向图$\boldsymbol{G}$=（$\boldsymbol{V}$,$\boldsymbol{E}$）

TSP对应的实例$\boldsymbol{f}$（$\boldsymbol{I}$）为：

城市集$\boldsymbol{V}$，任意两个不同的城市$\boldsymbol{u}$和$\boldsymbol{v}$之间的距离

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

以及界限$\boldsymbol{D}$=|$\boldsymbol{V}$|。

显然f是多项式时间可计算的；

$f(I)$中的巡回路线有$|V|$条边，每条边长1或2，

因而长度不超过D的巡回路线长度只能恰好为D，

当且仅当它的每条边的长度都为1，

也即当且仅当它是G的一条哈密尔顿回路，从而

$I \in Y_{\mathrm{HC}} \Leftrightarrow f(I) \in Y_{\mathrm{TSP}}$

例：最大生成树$\le_p$最小生成树

最小生成树：

任给连通网G=(V,E,w）以及正整数B，其中权w:$E \to Z^+$，问有权值和**小于等于B**的生成树吗？

最大生成树：

任给连通网G=（V,E,w）以及正整数D，其中权w:$E \to Z$+，问有权值和**大于等于D**的生成树吗？

证：

任给最大生成树的实例I：连通网G=（V,E,w）和正整数D，

最小生成树的对应实例$f(I)$： 图G'=（V,E,w'）和正整数B=（n−1）M−D，其中n=|V|， M=max{$w(e)$| $e\in E$}+1，$w'(e)$=M−$w(e)$，

如果存在G的生成树T，使得 $\sum_{e\in T} w(e) \geq D(I\in Y_{大})$，则图$G'$中对应的生成树有，

$$\sum_{e\in T} w'(e) = \sum_{e\in T}\big(M - w(e)\big) = (n-1)M - \sum_{e\in T} w(e)$$

$$\leq (n-1)M - D = B \ (f(I)\in Y_{小})$$

反之亦然

$\leq p$的性质： $\leq p$具有传递性

定理 如果$\boldsymbol{\Pi}_1 \leq_p \boldsymbol{\Pi}_2$， $\boldsymbol{\Pi}_2 \leq_p \boldsymbol{\Pi}_3$，则$\boldsymbol{\Pi}_1 \leq_p \boldsymbol{\Pi}_3$

提 纲

- 易解的问题与难解的问题
- 判定问题
- NP类
- 多项式时间变换
- NP完全性及Cook-Levin定理

NP完全性及其性质

1.问题能多项式时间变换到P类问题，则它是P类问题

定理：设$\boldsymbol{\mathit{\Pi}}_1 \le_p \boldsymbol{\mathit{\Pi}}_2$，$\boldsymbol{\mathit{\Pi}}_2 \in \mathbf{P} \Rightarrow \boldsymbol{\mathit{\Pi}}_1 \in \mathbf{P}$

2.难解问题能多项式时间变换到的问题都是难解问题

推论：设$\boldsymbol{\mathit{\Pi}}_1 \le_p \boldsymbol{\mathit{\Pi}}_2$，$\boldsymbol{\mathit{\Pi}}_1$是难解的，则$\boldsymbol{\mathit{\Pi}}_2$也是难解的

例如：

因为"最大生成树$\le_p$最小生成树"，则

最小生成树$\in$**P**$\Rightarrow$最大生成树$\in$**P**

因为"**HC**$\le_p$**TSP**"，则

HC是难解的 $\Rightarrow$ **TSP**也是难解的

NP难与NP完全

如果对于 $\forall \boldsymbol{\Pi}' \in \mathbf{NP}$，$\boldsymbol{\Pi}' \leq_p \boldsymbol{\Pi}$，则称$\boldsymbol{\Pi}$是NP难的；

如果$\boldsymbol{\Pi}$是NP难的，且$\boldsymbol{\Pi} \in \mathbf{NP}$，则称$\boldsymbol{\Pi}$是NP完全的。

NP完全的问题就是NP中最难的问题。

关于P与NP

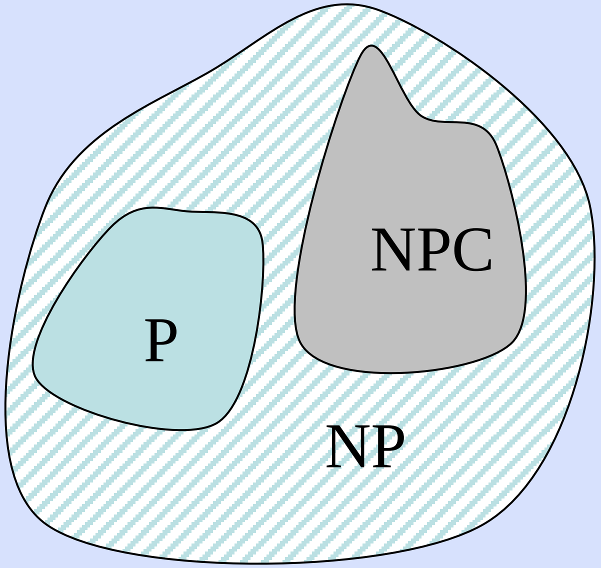
定理： 如果存在NP难的问题$\boldsymbol{\Pi}\in \mathbf{P}$，则P=NP。

证：因为$\boldsymbol{\Pi}$是NP难的，因此$\forall \boldsymbol{\Pi}' \in \mathbf{NP}$，$\boldsymbol{\Pi}' \le_p \boldsymbol{\Pi}$

又因，$\boldsymbol{\Pi}\in \mathbf{P}$，推出$\boldsymbol{\Pi}'\in \mathbf{P}$，即有NP⊆P，

又已知P⊆NP，所以有P=NP。

推论： 假设P≠NP，那么，如果$\boldsymbol{\Pi}$是NP难，则$\boldsymbol{\Pi}\notin \mathbf{P}$。



如何证明问题是NP完全的

定理：如果存在NP难的问题Π'，使得则$\Pi' \leq_p \Pi$， 则Π是NP难的。

推论：如果$\Pi \in$ NP，并且存在NP完全的问题Π'，使得则$\Pi' \leq_p \Pi$，则Π是NP完全的。

证明Π是NP完全的一般步骤：

1.证明$\Pi \in$ NP；

2.找一个已知的NP完全的问题Π'，证明$\Pi' \leq_p \Pi$

哪里去找这“第一个”NP完全的问题？

第一个NP完全问题

Cook-Levin定理

- 可满足性（SAT）

　任给一个合取范式F，如果F存在成真赋值，则称F是可满足的。问F是可满足的吗？

- Cook-Levin定理：SAT是NP完全的。

　SAT显然是NP问题

　如何证明SAT是NP难的？

- 定理：P=NP的充要条件是存在NP完全问题$\Pi \in$ P


谢谢大家！