集合与实数集

luojunxun

2023 年 10 月 26 日

组合优化问题：从有限个可行解中找出最优可行解的优化问题称为组合优化问题

计算复杂性

问题是带有若干参数的一个提问，给这些参数特定的值就得到一个实例，算法是求解该问题的通用步骤描述。（这里的算法和数据结构中的算法不同，后者是程序执行的总步骤，而这里的算法是步骤的描述，可以通俗化）

规模 (size): 描述一个实例所需要的字节数，比如存储一个整数 k 所需要的字节数为 $([log_2 k] + 1) + 1$(标志结尾的空格)

最大数：实例的最大数就是在实例中出现过的最大整数

算法时间复杂性：算法的时间复杂性是一个关于实例规模 x 的函数 f(x)，他表示用该算法处理所有规模为 x 的实例中所需基本运算最多的那个实例的基本运算次数

如果函数 f 是一个多项式时间函数，称对应的算法是多项式时间算法，不能被次限制的称为指数时间算法

如果算法时间复杂性是关于实例规模 x 和最大数 B 的二元函数 f(x,B)=O(P(x,b)), 但算法不是多项式时间的，称为伪多项式时间算法，如 O(nB)；这类算法的特点就在于实例规模比较小的时候计算速度很快，但是随着实例规模的增加，算法复杂度也会快速增加

有多项式算法的问题称为多项式时间可解问题类：记为 $\mathcal{P}$

NP-完全性理论

只回答是否的问题称为判定问题，判定问题和优化问题可以相互转化（NP-完全性理论研究判定问题（或者优化问题的判定形式））

非确定性算法多项式时间可解问题类：对一个问题，存在一个算法，使得对于任何一个回答为是的实例，该算法能猜出一个可行解，其算法规模，也就是描述可行解的字节数不超过实例的多项式函数，并且在不超过实例的多项式时间内能验证猜想正确，就称该问题为非确定性多项式时间可解问题类（nondeterministic polynomial solvable problem），记为 $\mathcal{NP}$ 类

P 问题是多项式时间可解问题，而 NP 问题是多项式时间可验证问题，自然有 $P \subset NP$

NP 类是非确定性图灵机多项式时间可解类

NP-C：NP 完全问题:$p_1 \in \mathcal{NP}, \forall P_2 \in \mathcal{NP}, P_2 \leq_p P_1 \Rightarrow p_1 \in \mathcal{NP} - C$

NP-C 是 NP 中最难的问题，表现为如果某一个 NP-C 有多项式时间算法，那么每个 NP 问题都有多项式时间算法，且 NP-C 问题都是同等难度的，用归约证明即可

NP-C 判定定理:$p \in \mathcal{NP}, \exists p_0 \in \mathcal{NP} - C, s.t. p \leq_p p_0 \Rightarrow p \in \mathcal{NP} - C$

子集和问题是 NP-C 问题：如果有一个极大化背包算法，怎么得到极小化背包最优解?

第 k 个最大子集和问题可能不属于 NP

强 NP-C

一个 NP-C 问题的所有最大数不超过规模的某个固定的多项式函数的实例都是 NP-C 的，那么这个 NP-C 问题称为强 NP-C 问题

强 NP-C 在 $P \neq NP$ 的条件下是没有伪多项式时间算法的

证明强 NP-C: 1. 问题是 NP-C 并且所有的实例 $B(I) \leq p(size(I))$:Hamilton 圈问题，SAT 问题；

2. 归约证明某问题的 NP-C 性质时，所有实例都满足 $B(I) \leq p(size(I))$；

3. 从一个已知的强 NP-C 问题伪多项式归约

3-划分问题

存在伪多项式时间算法的 NP-C 问题不是强 NP-C 问题，称为普通意义下的 NP-C 问题：例如背包问题，划分问题

至少和 NP-C 问题一样难的问题称为强 NP-C 问题

线性规划和整数规划

解决线性规划的单纯性法；椭球法，内点法（线性规划多项式时间算法）

整数线性规划是 NP-C：割平面法，分支定界法

组合优化问题的解决方法

设计出多项式时间算法：改进算法性能

or 复杂性未决问题：在多项式内近似求解，在指数时间求最优解，研究特殊解

or 证明为 NP-C

决策变量不放在下标中

1 动态规划

动态规划 dynamic programming 是求解多阶段决策优化问题的数学方法和算法思想

将求解的实例转化为规模比较小的实例，利用递推关系导出最优解之间的关系

最优化原理：n 个决策过程中，如果第一个决策不管如何决策，后 n-1 个决策都能决策出第一个决策结果造成的总体最优解（最优解的部分必须要是最优解）

2 分支定界法

背包问题的分支定界法：高度为 n 的二叉树，每个分支表示一个物品放与不放，剪枝：该枝不存在可行解

定界：任何一个已经找到的可行解的目标值都是最优解的下界（全局），上界：每一枝，可行解的目标值的上界，如果上界不大于下界，那么该枝不存在更好的可行解：（可去整数线性规划的松弛线性规划的最优值，或者已经放入物品的价值加上未放入物品的价值和）