NP-Completeness of SAP

tqyaaaaang

Tsinghua University

4/18/2019

P, NP, NP-Hard and NP-Complete

这里我们只考虑判定性问题，即判断是否存在一个满足某个条件的解。那么 P、NP、NP-Hard 和 NP-Complete 分别定义如下：

- P：能在多项式时间内解决的问题
- NP：不确定是否能在多项式时间内解决，但能在多项式时间内验证的问题
- NP-Hard：任何 NP 问题都能在多项式时间内归约到它的问题，至少比 NP 问题要强
- NP-Complete：NP 问题与 NP-Hard 问题的交集，即能在多项式时间验证，但是能被所有 NP 问题在多项式时间内规约到的问题

这里在多项式时间内解决指相对于 输入长度规模
解决一个 NP-Complete 问题就意味着我们可以解决所有 NP 问题。

常见的 NP-Complete 问题 [4]

- 哈密尔顿回路问题
- 旅行商问题
- 3-SAT 问题
- 背包问题
- 最大独立集问题
- 图着色问题
- ……

判定性问题和最优化问题

很多时候可能的答案集合是一个和输入规模的指数规模相关的集合。

例如旅行商问题求最小解，但若有 m 条边，答案只会有 2^m 种情况。可以对这些情况做二分答案，就可以把它的判定性问题规约到最优化问题。

Weakly NP-Hard and Strongly NP-Hard

我们称这种关于输入<u>数值</u> 规模而不是输入<u>长度</u> 规模多项式的算法称作伪多项式算法 (Pseudo-polynomial time)

我们称 Partition Problem 这种存在 Pseudo-polynomial time 算法的 NP-Hard 问题称作 Weakly NP-Hard。否则称作 Strongly NP-Hard。

Partition Problem

给定一个大小为 n 的集合 S，求集合是否存在一个子集 $T \subseteq S$，满足 $sum(T) = sum(S - T)$。

存在 $O(n \cdot sum(S))$ 的动态规划算法，因此是 Weakly NP-Hard 的。

Bin Packing Problem

将 n 个大小分别为 s_i 的物品放到 k 个大小为 V 的箱子里，问是否可行。

这个问题是 Strongly NP-Hard 的。

内存分配问题

有 n 个内存分配的申请，有一些内存，要求为每个申请分配内存中连续的一段，使得内存分配不会出现冲突。

- 离线问题：我们事先知道所有申请序列
- 在线问题：动态的对每个申请进行处理

优化目标 - 最大化峰值利用率

这里我们考虑的是最大化峰值利用率 [3]。即若我们假定 n 个分配和释放请求按照顺序处理，在处理完第 i 个情况之后，它的有效载荷 (payload) 是 P_i，即当前被分配的所有块的大小和为 P_i，而此时堆的大小是 H_i。则第 i 个时刻的峰值利用率 U_i 定义为

$$U_i = \frac{\max_{j \le i} P_j}{H_i}$$

我们分配器的目标就是最大化 U_n。

而一般我们假设 H_i 是单调不减的，而由于每个时刻我们 P_i 都是确定且已知的，因此 $\max\limits_{i \le n} P_i$ 也是已知的。因此我们最大化 U_n，等于最小化 H_n。

离线问题

假定有 m 个单位空间的内存（代表一个大小为 m 的堆）。给定 n 个内存分配的申请，第 i 个申请有一个申请大小 $size_i$，以及存在的时间区间 $[tl_i, tr_i]$。要为每个内存分配请求分配空间中一个连续的区间 $[l_i, r_i]$，区间长度为 $size_i$，即满足 $r_i - l_i + 1 = size_i$。要求不存在两个分配 i 和 j 满足 $[tl_i, tr_i]$ 和 $[tl_j, tr_j]$ 相交且 $[l_i, r_i]$ 和 $[l_j, r_j]$ 相交。

求最小的 m 使得存在一个可行的 $[l_i, r_i]$ 序列满足上述条件。

离线问题

假定有 m 个单位空间的内存（代表一个大小为 m 的堆）。给定 n 个内存分配的申请，第 i 个申请有一个申请大小 $size_i$，以及存在的时间区间 $[tl_i, tr_i]$。要为每个内存分配请求分配空间中一个连续的区间 $[l_i, r_i]$，区间长度为 $size_i$，即满足 $r_i - l_i + 1 = size_i$。要求不存在两个分配 i 和 j 满足 $[tl_i, tr_i]$ 和 $[tl_j, tr_j]$ 相交且 $[l_i, r_i]$ 和 $[l_j, r_j]$ 相交。

求最小的 m 使得存在一个可行的 $[l_i, r_i]$ 序列满足上述条件。

答案肯定是 $[0, \sum size_i]$ 的整数，转化成判定性问题求解。

问题转化

将每个内存分配请求放到平面上，每个内存分配请求是一个以 $[tl_i, tr_i]$ 为横坐标区间，$[l_i, r_i]$ 为纵坐标区间的矩形。

内存分配不能冲突的限制就等价与 n 个内存分配请求对应的矩形不相交。而对于每个矩形，$[tl_i, tr_i]$ 是请求的性质，是固定的，而你要求的就是一个可行的 $[l_i, r_i]$ 序列。

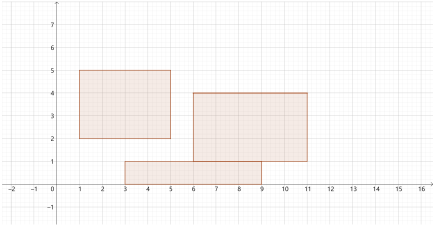

Figure: 内存分配图示例

问题转化

在二维平面上，给定 n 个矩形，问是否存在一种将所有矩形都上下平移的方案，使得所有矩形的纵坐标都在 $0 \sim m$ 之间，且各个矩形不相交。

这个问题是 NP-Complete 的。

证明

我们考虑下面这个内存分配图（若有边横坐标相同重合表示它们的先后顺序不影响）：

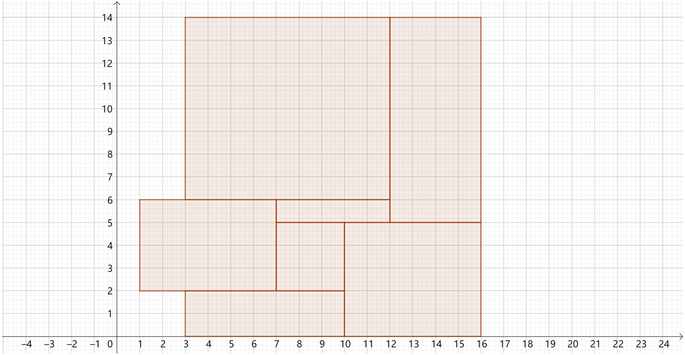

Figure: 构造内存分配图

证明

那么我们考虑对于一个 Partition Problem，给定一个 n 个元素的集合 S，S 中所有元素的和为 $2K$，则我们在上面图的基础上添加如下一些存在时间在 $[tl_i, tr_i] = [1, 2]$ 的内存申请：

1. 一个大小为 $6K$ 的申请
2. 对于集合 S 中的每个元素 a，添加一个大小为 $2a$ 的申请

将第二部分的每个申请等分成上下两部分。转化为 Partition Problem。

在 [6] 中还提出了一个从 Knapsack Problem 归约的证明。

优化目标 - 最大化权值和

在 [1] 中提出了另一种优化目标，尝试最大化权值和，但是在这种目标下判定问题是 Strongly NP-Hard 的。

在第 i 个时刻，可用的内存空间最多只有 c_i 个单位大小，第 i 个申请有个重要性权值 w_i，要选择权值和尽可能大的申请集合，使得能够被合法地分配。

证明

假设对于一个 n 个大小分别为 s_i 的物品放到 k 个大小为 1 的箱子里 Bin Packing Problem，我们考虑将其归约到我们这个问题上。

构造一个长度为 $2k+1$ 的时间线，第 c_i 满足如下式子：

$$c_i = \begin{cases} 2i-1 & i \le k \\ 2k-1 & k+1 \le i \le k+2 \\ 2(2k+2-i) & i \ge k+3 \end{cases}$$

证明

而构造如下 $2k-1$ 个申请：

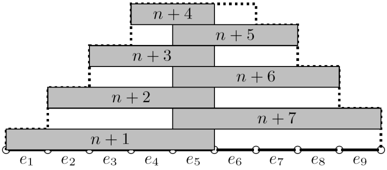

Figure: 构造内存分配图

即 $[1,k+1],[2,k+1],[3,k+1],\ldots,[k,k+1]$ 和 $[k+1,k+3],[k+1,k+4],[k+1,k+5],\ldots,[k+1,2k+1]$ 的一些大小为 1 的申请

证明

而构造如下 $2k-1$ 个申请：

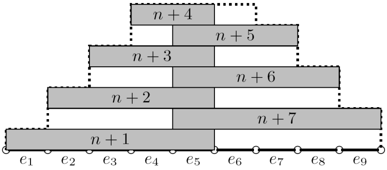

Figure: 构造内存分配图

再在 $[k+2, k+2]$ 的位置放 n 个申请，分别大小是 s_i，则等同于将这 n 个申请放到 k 个大小为 1 的缝隙中，即为 Bin Packing Problem。

近似算法

- 2013 年 Bar-Yehuda 等人提出了 $(9+\epsilon)$ 近似比的多项式算法 [1]
- 2015 年 Batra 等人提出了 $(2+\epsilon)$ 近似比的多项式算法 [5]

常用在线算法

在线的内存分配算法就更为困难了。因此一般都采用贪心或启发式的方法：

- Sequential-fit Methods：贪心策略，最先/最佳/最差匹配，循环匹配
- Buddy Methods：分块策略，Buddy System
- Segregated-storage Methods：启发式策略，Slab Allocator [2]

Reuven Bar-Yehuda, Michael Beder, and Dror Rawitz.
A constant factor approximation algorithm for the storage allocation problem.
Algorithmica, 77(4):1105–1127, 2017.

Jeff Bonwick et al.
The slab allocator: An object-caching kernel memory allocator.
In *USENIX summer*, volume 16. Boston, MA, USA, 1994.

Randal E. Bryant and David R. O'Hallaron.
Computer Systems: A Programmer's Perspective Plus MasteringEngineering with Pearson eText – Access Card Package.
Pearson, 3rd edition, 2015.

Richard M Karp.
Reducibility among combinatorial problems.
In *Complexity of computer computations*, pages 85–103. Springer, 1972.

Tobias Mömke and Andreas Wiese.
A $(2+\epsilon)$ -approximation algorithm for the storage allocation problem.

2015.

J. M. Robson.
Storage allocation is np-hard.
Information Processing Letters, 11(3):119–125, 1980.