

chap 3 排序与装箱组合优化

chap 3 排序与装箱组合优化

排序与装箱问题

代表性问题:

1. Stanta\ Claus问题

2. 订单管理问题

$\sum T_{ij}$ 的伪多项式时间算法

装箱问题

Next Fit

First Fit Decreasing

装箱问题的最坏情况界不会低于 $3/2$

渐近性能比

渐近近似方案

排序与装箱问题

可以使用的资源称为机器，需要完成的任务称为工件。

排序问题数量众多，每一个排序问题可以用机器环境，工件特征与目标函数3个具体要素来表示。一般用三参数表示法 $(\alpha, |\beta|, \gamma)$ 来表示各种排序问题。三个参数分别对应于具体的机器环境，工件特征与目标函数。

机器环境有很多机器，一般包括以下几种需要记忆的:

1. 单台机，意思是工件加工一次的一台机器，表示为1

2. 平行机， m 台($m \geq 2$)功能相同的机器，机件在任意一台上加工一次

平行机又包括同型机，即所有机器都完全相同，同一工件在各台机器上加工时间都相同，可以用 Pm 来表示

同类机，也就是同一工件在各台机器上加工所需时间是成比例的。该比例对所有的工件都相同，可以记作 Qm

不同类机，不同工件在各台机器上加工时间不存在特定关系

对于作业类型还可以分为流水作业(每个工件需要多道相同工序进行加工，每道工序需要在 m 台机器上指定一台加工 Fm)，工件作业（每个工件有不同工序，工序按固定顺序在相应机器上加工 Jm)，自由作业（每个工件工序各自不同，可以按照任意顺序加工 Om)

工件特征也反应工件的基本信息，给出工件在加工过程中所需要满足的限制。一般记工件为 (J_1, J_2, \dots, J_n) ，我们能掌握的有用的信息有：

加工时间： p_j 工件在机器上加工所需要的标准时间

释放时间： r_j 自零时刻开始到工件可以加工所需要的时间

交工期： d_j 是工件预定的完工交付时间

然后我们可以优化的东西大概有以下几种：

1.总完工时间

可以定义为 $\sum_{j=1}^n C_j$

2.工件最大完工时间

可以定义为 $\max_{1 \leq j \leq n} C_j = C_{\max}$

3.最大延迟

定义为 $\max_{1 \leq j \leq n} (C_j - d_j) = L_{\max}$

4.总延误时间

$\sum_{j=1}^n T_j$, 其中 $T_j = \max(C_j - d_j, 0)$

5.误工的工件数目

6.机器的最小负载

记录机器的负载为在机器上加工工件时间之和

$\min_{1 \leq i \leq m} L_i$

记作 C_{\min}

代表性问题：

1. *Stanta Claus*问题

圣诞老人有n个礼物，分给m个小朋友。第i个小朋友对礼物j的满意度为 $p_{i,j}$,而获得多件礼物的小朋友总满意度为他得到的各个礼物的满意度之和。

我们优化的目的是让满意度最小的小朋友满意度尽量大，也就是说 $\max_{1 \leq i \leq m} C_{\min}$

2. 订单管理问题

某个企业有m间分厂，每一家分厂固定生产某种商品，每一个订单有不同的商品需要在不同企业生产。现在有m个订单，完成所有零件就可以交工。如何安排使得总交工时间最小。

相当于每个订单是一个多工序的零件，每个工序没有顺序，不同工件也没有顺序

经典排序问题：

$$(一) 1||C_{max}, 1||\sum C_j$$

第一个很简单嘛，任意排列

第二个的话就是先加工小的后加工大的

$$(二) 1||L_{max}, 1||\sum U_j, 1||\sum T_j$$

第一个就是最大延误时间最小

第二个就是延误工件数目最小

第三个就是总延误时间最小

EDD：

将工件的交工期按非降顺序排列，也就是说令排列后的工件满足 $d_1 \leq d_2 \leq d_3 \leq \dots \leq d_n$ ，然后依照顺序加工即可。解决了 L_{max} 问题

最大延迟，误工工件数目与总延误时间这三个目标不能相互替代

$1||\sum T_j$ 的伪多项式时间算法

基本概念：

多项式时间近似方案：

(Polynomial Time Approximation Scheme, PTAS)

如果对于任意给定的 $\epsilon > 0$ 算法 A_ϵ ，求解问题 II 的最坏情况界 $r_{A_\epsilon} = 1 + \epsilon$ ，且 A_ϵ 的时间复杂性 $f(n) = O(p(n^{1/\epsilon}))$ ， p 是土匪多项式

完全多项式时间近似方案 (FPTAS), $f(n, \epsilon) = O(p(n, \frac{1}{\epsilon}))$

装箱问题

将一系列物品放入容量一定的若干个箱子里，放入每个箱子的物品大小之和不超过箱子容量，目标为所用的箱子数目尽量少

子问题：

1. 下料问题：给定生产一批产品所需要的某种材料的大小与数量列表，如何从一定规格的原料中下料，使得所用的原料最少。（这里就是装箱了，原料就是箱子，如何把材料大小填入箱子里面

2. 一维装箱问题：箱子的容量与物品大小都是有理数
3. 二维装箱，二维条状装箱，三维装箱

如何表达一个问题呢？

一维装箱的表达：

记录箱子的容量为 C ，物品 j 的大小为 $w_j, 0 \leq w_j \leq C, j = 1, 2, \dots, n$

用 B_i 表示按顺序第 i 个启用，同时表示当前放入该箱子的物品大小之和

Next Fit

设当前装箱的物品为 j ，最后启用的箱子为 B_i ，也就是说 $i = \max\{k | B_k^{j-1} > 0\}$

反正箱子数目是无限多的，然后此时我们最后用到第 i 个箱子，此时如果 $w_j + B_i^{j-1} \leq C$ ，说明可以放，将 j 放入 B_i 中，如果大于 C 了，放入下一个箱子，然后 B_i 不放东西了

最坏情况界为2

First Fit Decreasing

将物品按照大小非增的顺序重新排序，

装箱问题的最坏情况界不会低于3/2

采用划分集问题来归约

假设低于，那么有 $\frac{C^A(I)}{C^*(I)} < \frac{3}{2}$

那么构造划分集问题即可

渐近性能比

渐近性能比是为了衡量一个算法在实例较大的那些情况下的性能的。

算法A的渐近性能比记作

$$r_A^\infty = \inf\{r \geq 1 \mid \text{存在整数 } N, \text{ 使得对任意满足 } C^*(I) \geq N \text{ 的实例 } I, \text{ 存在 } \frac{C^A(I)}{C^*(I)} \leq r\}$$

也就是说是一个渐近最坏情况界

如果说存在与实例无关的常数 c ，使得对于所有的实例 I ，满足 $C^A(I) \leq kC^*(I) + c$ ，那么A的渐近性能比不超过 k

同样如果我们不规定N的话，直接把 $r_A = \inf\{r \geq 1 | C^A(I) \leq r * C^*(I), \forall I\}$ 为算法A的绝对性能比

一般来说渐近肯定小于绝对。绝对意味着最坏情况，渐近则是实例很大的情况

渐近近似方案

渐近多项式时间近似方案 (APTAS)

也就是说， $C^{A_\epsilon} \leq (1 + \epsilon)C^* + 1$ ，它的时间复杂性为n的多项式，但是是 $1/\epsilon$ 的指数函数

渐近完全多项式时间近似方案 (AFPTAS)

$C^{A_\epsilon} \leq (1 + \epsilon)C^* + O(\frac{1}{\epsilon^2})$ ，时间复杂性为n与 $\frac{1}{\epsilon}$ 的多项式