第3章

协同代理调度

3.1 引言

前一章研究了在机器计件维护情形下的单机调度问题.可在实际生产中,还会因为任务属于不同的代理而具有不同的调度目标,因此研究同时考虑协同代理和计件维护的最优调度方案问题更具有实际意义.

大量现有文献主要是考虑协同代理时考虑相互竞争代理的调度研究.刘鹏等(2011)用任务基本加工时间和任务开始加工时间的递减线性函数之积来表示任务的实际加工时间,并在一个代理目标不超过给定上限的另一个代理的任务处理时间表长及最大延迟最小化的目标要求下,得到了获得最优调度方案的多项式时间算法.Liu 等(2010)在线性函数形式的分组发动时间模型以及线性函数形式的任务处理时间模型的基础上,研究了最小化第一个代理总完工时间且用给定上限约束第二个代理总代价的单机分组调度,分析得到多项式时间算法求得最优的单机调度方案.Agnetis 等(2007,2009)、Cheng 等(2006)、Janiak 等(2006)都对此有相关研究.

和上述大量文献考虑相互竞争代理调度研究不同,Yu 等(2013)研究了考虑两个任务代理和机器计件维护的单机调度,在目标函数为任务完工时刻的整系数多项式函数的条件下,给出了求得最优调度方案的多项式时间算法.

本章研究的协同代理调度问题是:对给定的一批任务,任务分属于两个协同代理,任务处理时间在达到给定的任务排序位置时发生改变;考虑一台机器的情形,机器每次只能处理一项任务,分别考虑机器不施行机器维护情形和施行计件维护情形(在此情形下,机器每处理完一定数量的任务后进行一次维护,维

护后机器状态恢复如新).

本章首先在最小化任务处理时间表长的多项式函数的目标要求下,分别建立处理时间可变影响下的只考虑协同代理的情形和同时考虑协同代理和计件维护情形的单机调度模型,接着研究确定相应最优协同代理调度方案的多项式时间求解算法,并分析求解算法的计算复杂度.

3.2　问题描述及模型

基于生产调度研究的 $\alpha|\beta|\gamma$ 三阶段模型,本节接下来阐述协同代理的单机调度的相关符号,给出相应的三阶段调度模型.

所有的待处理的任务分属于两个代理,即代理 1 和代理 2.每个代理所包含的一组任务,需要在同一台机器上处理.相应于代理 1 和代理 2,把任务分成 $J^{(1)}$ 和 $J^{(2)}$ 两类,即 $J^{(i)}=\{J_1^{(i)},J_2^{(i)},\cdots,J_{n_i}^{(i)}\}$,$i=1,2$.在集合 $J^{(i)}$ 的任务被称为第 i 个代理的任务.任务 $J_j^{(i)}$ 的处理时间用一个正整数 $p_j^{(i)}$ 表示.所有任务的释放时间为零时刻,需要在同一台机器上处理,一个任务处理与相邻的任务处理之间没有空闲时间.任务在机器上处理的队列成为一个调度序列.在一个调度序列 σ 中,任务 $J_j^{(i)}$ 的完工时间记为 $C_j^{(i)}(\sigma)$.对于 $J_j^{(i)}$,用 $f_j^{(i)}(\cdot)$表示任务 $J_j^{(i)}$ 完工时间的一个单调不减的函数,这样的目标函数在现有的调度研究文献中被称为正则函数.把全部任务 $J^{(1)}$ 和任务 $J^{(2)}$ 的处理时间之和分别记为 τ,$\tau^{(1)}$ 和 $\tau^{(2)}$.在调度过程中,用 $\bar{\tau}$ 表示没有被安排的任务处理时间之和.

第 i 个代理的目标函数记为 $f_{\max}^{(i)}(\sigma)=\max\limits_{1\leqslant j\leqslant n_i} f_j^{(i)}(C_j^{(i)}(\sigma))$.函数 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$是 $f_{\max}^{(1)}(\sigma)$和 $f_{\max}^{(2)}(\sigma)$的复合目标函数,且它关于 $f_{\max}^{(1)}(\sigma)$和 $f_{\max}^{(2)}(\sigma)$是严格单调递增的.依据 Graham 等(1979)提出的调度问题记法,把所研究的考虑两个协同代理的单机调度问题记为 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$,其中调度目的是找到一个调度方案 σ 使得目标函数 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$最小.研究的第二个调度模型 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 的决策目标是在满足约束条件 $f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 前提下找到一个调度方案 σ 使得目标函数 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$最小.结合两个相似的约束条件 $f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 和 $f_{\max}^{(1)}(\sigma)\leqslant Q_1$,$f_{\max}^{(2)}(\sigma)\leqslant Q_2$,本章相应地研究以下两个相似的包含协同代理的调度问题:

$$1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(2)}(\sigma)\leqslant Q_2 \quad 和$$

$$1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1,\quad f_{\max}^{(2)}(\sigma)\leqslant Q_2.$$

针对任务处理时间关于给定排序位置可发生改变的情形,这里给出任务的处理时间函数形式及符号说明.有 n 个任务需要在机器上加工处理,机器只能逐

个处理任务，且单个任务处理开始后不允许中断切换到其他的任务处理. 任务 j $(j=1,2,\cdots,n)$所需的基本处理时间记为 p_j. 本节考虑的一般性的关于位置的时间突变的任务处理时间为

$$p_j^r=\varphi_j^*(p_j,r)=\begin{cases}p_j, & 1\leqslant j\leqslant n,1\leqslant r\leqslant B\leqslant n,\\ +\infty, & 1\leqslant j\leqslant n,r>B.\end{cases}\tag{3-1}$$

在式(3-1)中，函数 $\varphi_j^*(p_j,r)$是关于任务排序位置的分段函数，用来描述任务处理时间关于给定排序位置可发生改变的情况. 当累计处理任务的件数不超过上限 B 时，任务的实际处理时间等于基本任务处理时间 p_j；当累计处理任务的件数超过上限 B 时，任务的实际处理时间突然变得极长(用$+\infty$表示).

针对任务处理时间关于给定排序位置发生突变的情形，本章采用计件维护技术预防任务的实际处理时间的突变. 接下来阐述本章所考虑的机器计件维护及相关符号. 在调度过程中，机器每处理或处理完 l 项任务或产品就需要进行一次保养维护，每次机器维护的时间长度为 t. 在机器维护的过程中，需要关闭机器，任务处理过程停止. 假设机器经过维护后恢复到初始的状态. 因此，在任意一个调度方案 σ 中机器维护嵌入到任务处理序列中，从而在每个调度方案中任务处理被机器维护分隔成一个个的任务处理组. 不失一般性，把第 i 组记为 G_i，即 $G_i=[J_{i1},J_{i2},\cdots,J_{in_i}]$，其中 $1\leqslant i\leqslant k+1$，$J_{ir}$是任务处理组 G_i 中的第 r 个任务. 因此，整个调度方案 σ 可以表示为$\sigma=[G_1,M,G_2,M,\cdots,M,G_{k+1}]$，其中 M 表示机器维护，k 表示机器维护的次数. 因为机器每处理或处理完 l 项任务或产品就需要进行一次保养维护，所以除了最后一个任务处理组，每个组中的任务处理数都为 l 个. 图 3-1 是固定维护时间的计件维护调度时序示意图.

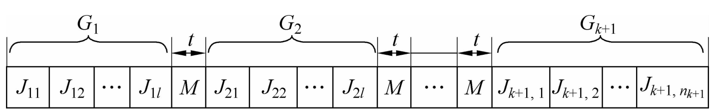

图 3-1 维护时间固定的计件维护调度时序示意图

在 3.3 节中将研究的处理时间可变影响下的不考虑计件维护的协同代理单机调度可记为 $1|p_j^r=\varphi_j^*(p_j,r)|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$，其研究的决策目标是找到一个调度方案 σ 使得 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$最小化. 分别结合 $f_{\max}^{(1)}(\sigma)\leqslant Q_1$，$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 和($f_{\max}^{(1)}(\sigma)\leqslant Q_1$，$f_{\max}^{(2)}(\sigma)\leqslant Q_2$)等约束条件下的调度问题记为如下三阶段模型：

$1\mid p_j^r=\varphi_j^*(p_j,r)\mid g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$：$f_{\max}^{(1)}(\sigma)\leqslant Q_1$，

$1\mid p_j^r=\varphi_j^*(p_j,r)\mid g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$：$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 和

$1\mid p_j^r=\varphi_j^*(p_j,r)\mid g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$：$f_{\max}^{(1)}(\sigma)\leqslant Q_1$， $f_{\max}^{(2)}(\sigma)\leqslant Q_2$.

在 3.4 节中将研究的处理时间可变影响下的同时考虑计件维护和两个协同代理的单机调度问题可记为 $1|p_j^r=\varphi_j^*(p_j,r),\mathrm{PRM}=l|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$，其中 PRM 表示计件维护，而 PRM$=l$ 表示每处理完 l 个任务机器就进行维护一次. 研究的决策目标是找到一个调度方案 σ 使得 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 最小化. 类似地，分别结合 $f_{\max}^{(1)}(\sigma)\leqslant Q_1$，$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 和（$f_{\max}^{(1)}(\sigma)\leqslant Q_1$，$f_{\max}^{(2)}(\sigma)\leqslant Q_2$）等约束条件下的调度问题记为如下三阶段模型：

$$1\mid p_j^r=\varphi_j^*(p_j,r),\quad \mathrm{PRM}=l\mid g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))\colon f_{\max}^{(1)}(\sigma)\leqslant Q_1,$$

$$1\mid p_j^r=\varphi_j^*(p_j,r),\quad \mathrm{PRM}=l\mid g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))\colon f_{\max}^{(2)}(\sigma)\leqslant Q_2 \quad 和$$

$$1\mid p_j^r=\varphi_j^*(p_j,r),\quad \mathrm{PRM}=l\mid g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))\colon f_{\max}^{(1)}(\sigma)\leqslant Q_1,\quad f_{\max}^{(2)}(\sigma)\leqslant Q_2.$$

3.3 调度模型分析

本节对给出的协同代理的调度模型进行分析，给出构建模型求解算法的基本思路. 为了求解协同代理的调度模型，首先介绍文献 Agnetis 等（2004）中两个竞争代理的调度问题及其研究结论.

约束优化问题的决策目标是在满足约束条件 $f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 前提下找到一个调度方案 σ 使得目标函数 $f_{\max}^{(1)}$ 最小. 本节把这个约束优化问题记为 $1\parallel f_{\max}^{(1)}\colon f_{\max}^{(2)}\leqslant Q_2$. 对称地，调度问题 $1\parallel f_{\max}^{(2)}\colon f_{\max}^{(1)}\leqslant Q_1$ 的目的是在满足约束条件 $f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 前提下找到一个调度方案 σ 使得目标函数 $f_{\max}^{(2)}$ 最小.

基于 Lawler（1973）给出的算法，Agnetis 等（2004）给出了问题 $1\parallel f_{\max}^{(1)}\colon f_{\max}^{(2)}\leqslant Q_2$ 的一个多项式算法. 下面给出其算法的简略叙述，令

$$f_j(x)=\begin{cases} f_j^1(x), & \text{如果订单是属于代理 } J^{(1)}, \\ +\infty, & \text{如果订单是属于代理 } J^{(2)} \text{ 且 } f_j^2(x)>Q_2, \\ -\infty, & \text{如果订单是属于代理 } J^{(2)} \text{ 且 } f_j^2(x)\leqslant Q_2, \end{cases} \quad 其中\ x\geqslant 0.$$

如果对于 $1|\mathrm{prec}|f_{\max}$ 存在一个可行的调度方案 σ^*，其中目标函数值 $f_{\max}^*$ 是有限的，则 σ^* 是调度问题 $1\parallel f_{\max}^{(1)}\colon f_{\max}^{(2)}\leqslant Q_2$ 的一个可行且最优的调度方案.

具体来说，在每一步，算法在还没有被安排的任务中选择合适的放到调度序列的最后面. 本节用 τ 表示没有安排的任务需要的处理时间总和. 对于任意一个没有被安排的且属于代理 $J^{(2)}$ 的任务 $J_k^{(2)}$，如果满足 $f_k^2(t)\leqslant Q_2$，则把这个任务安排在时刻 τ. 如果没有属于代理 $J^{(2)}$ 的任务 $J_k^{(2)}$，可以选择代理 $J^{(1)}$ 中使得 $f_k^{(1)}(\tau)$ 最小的任务 $J_k^{(1)}$ 安排在时刻 τ. 如果所有属于代理 $J^{(1)}$ 的任务都被安排，没有合适的属于代理 $J^{(2)}$ 的任务可供安排，则这时调度问题没有可行解.

相对称地，以上算法也适用于问题 $1\parallel f_{\max}^{(2)}\colon f_{\max}^{(1)}\leqslant Q_1$.

非被主导调度是指，如果找不到一个调度方案 σ'，使得 $f_{\max}^{(1)}(\sigma')\leqslant f_{\max}^{(1)}(\sigma)$，$f_{\max}^{(2)}(\sigma')\leqslant f_{\max}^{(2)}(\sigma)$，其中至少一个不等式是严格小于的，则调度方案 σ 是非被主导的调度方案，$(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 是一组非被主导的目标函数值.

帕雷托优化问题是要找到问题的所有非被主导的目标函数值数组及其对应的代理 $J^{(1)}$ 和代理 $J^{(2)}$ 调度安排. 在文献 Agnetis 等(2004)中，帕雷托优化问题被记为 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$. 所研究的考虑两个协同代理的单机调度问题要求找到调度方案使得 $f_{\max}^{(1)}(\sigma)$ 和 $f_{\max}^{(2)}(\sigma)$ 的复合函数 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 最小化，接下来考虑调度问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$.

定理 3.1 如果调度方案 σ 是调度问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的最优调度，则 σ 必然是调度问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的一个非被主导调度.

证明(反证法) 假设调度方案 σ 是 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的一个最优调度，但不是问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的非被主导调度方案，则至少存在一个调度方案 σ' 满足以下的条件：

$$f_{\max}^{(1)}(\sigma')\leqslant f_{\max}^{(1)}(\sigma),\quad f_{\max}^{(2)}(\sigma')\leqslant f_{\max}^{(2)}(\sigma),$$

并且以上两个不等式是严格小于的. 不失一般性，假设成立以下不等式：

$$f_{\max}^{(1)}(\sigma')\leqslant f_{\max}^{(1)}(\sigma),\quad f_{\max}^{(2)}(\sigma')< f_{\max}^{(2)}(\sigma).$$

因为函数 $g(f_{\max}^{(1)}(\cdot),f_{\max}^{(2)}(\cdot))$ 关于 $f_{\max}^{(1)}(\sigma)$ 和 $f_{\max}^{(2)}(\sigma)$ 是严格单调递增的，所以可以得到 $g(f_{\max}^{(1)}(\sigma'),f_{\max}^{(2)}(\sigma'))<g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$. 即调度方案 σ' 的目标函数值比调度方案 σ 的目标函数值要小. 这与调度方案 σ 是问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的最优调度方案的假设矛盾，证毕. □

引理 3.1(Agnetis 等，2004)[232-233] 问题 $1\parallel f_{\max}^{(1)}: f_{\max}^{(2)}\leqslant Q_2$ 求解的计算复杂度为 $O(n_1^2+n_2\log n_2)$.

根据引理 3.1，可以由对称性得到问题 $1\parallel f_{\max}^{(2)}: f_{\max}^{(1)}\leqslant Q_1$ 的求解复杂度也是 $O(n_2^2+n_1\log n_1)$.

任务的实际处理时间 $p_j^{(i)}$ 都是整数，因此函数 $f_{\max}^{(i)}(\sigma)=\max\limits_{1\leqslant j\leqslant n_i} f_j^{(i)}(C_j^{(i)}(\sigma))$ 的函数值也是整数. 这里设定决策目标中函数 $f_j^{(i)}(\cdot)$ 和函数 $g(\cdot,\cdot)$ 都是整系数的多项式函数. 因此，$f_{\max}^{(i)}(\sigma)$ 和 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的函数值一定是整数和任务处理时间总和 τ 的多项式函数倍.

根据以上的分析，这里给出构建求解两个协同代理的调度问题求解算法的基本思路：先找到帕雷托优化问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的所有可能的非被主导调度方案，然后选择在约束条件限制的范围内使得目标函数值 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 最小，最小目标函数值对应的调度方案即为调度问题的最优调度方案.

3.4 不考虑计件维护的求解算法分析

本节将研究考虑两个协同代理但不考虑机器维护的单机调度，分析求得最优调度方案的求解算法，并分析相关算法的计算复杂度.

在给出求解算法之前，先将几个需要用到的函数值列举出来，计算以下代理 $J^{(2)}$ 目标函数值.

函数 $f_{\max}^{(i)}(\sigma)$ 的上限：$Q_i^{\max}=\max\limits_{1\leqslant j\leqslant n_2} f_j^{(i)}(\tau)$，

函数 $f_{\max}^{(i)}(\sigma)$ 的下界：$Q_i^{\min}=\min\limits_{1\leqslant j\leqslant n_2} f_j^{(i)}(\tau^{(i)})$，其中 $i=1,2$。

接下来，给出调度问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的求解算法.

算法 3.1 问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的求解算法.

步骤 1 计算 $Q_i^{\max}$，$Q_i^{\min}$，$i=1,2$.

步骤 2 令 $Q^*=\begin{cases} Q_1, & (Q_1^{\max}-Q_1^{\min})\leqslant(Q_2^{\max}-Q_1^{\min}), \\ Q_2, & (Q_1^{\max}-Q_1^{\min})>(Q_2^{\max}-Q_2^{\min}). \end{cases}$

令 $Q^*=Q_2$，调度方案集合 $s=\varnothing$，$i=0$，转步骤 3.

步骤 3 如果使用 Agnetis 等(2004)[233] 的算法能找到问题 $1\parallel f_{\max}^{(1)}(\sigma)$：$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的最优调度，进而找到问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的一个非被主导调度 σ^i，则问题 $1\parallel f_{\max}^{(1)}(\sigma)$：$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 是可解的，令 $i=i+1$，进入步骤 4. 否则，问题 $1\parallel f_{\max}^{(1)}(\sigma)$：$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 是不可解的，进入步骤 5.

步骤 4 令 $s=s\cup\sigma^i$，$Q_2'=f_{\max}^{(2)}(\sigma^i)$，$Q_2=Q_2'-1$，转入步骤 3. 如果 $Q_2<Q_2^{\min}$，转入步骤 5.

步骤 5 记录所有的非被主导的调度方案集合 s。计算集合内每个方案对应的目标函数值 $g(f_{\max}^{(1)}(\cdot),f_{\max}^{(2)}(\cdot))$. 具有最小目标函数值 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的调度方案 σ 即为所求得的最优调度方案.

引理 3.2(Agnetis 等，2004)[240] 调度问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 最多有 n_1n_2 个非被主导的调度方案.

定理 3.2 如果 $Q^*=Q_2$，则问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 求解的计算复杂度为 $O((n_1^2+n_2\log n_2)(Q_2^{\max}-Q_2^{\min})+n_1n_2)$；如果 $Q^*=Q_1$，则问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 求解的计算复杂度为 $O((n_2^2+n_1\log n_1)(Q_1^{\max}-Q_1^{\min})+n_1n_2)$.

证明 基于文献 Agnetis 等(2004)，可以得到求解调度问题 $1\parallel f_{\max}^{(1)}(\sigma)$：$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的计算复杂度为 $O(n_1^2+n_2\log n_2)$. 在算法 3.1 中，循环迭代的次数不超过 $(Q_2^{\max}-Q_2^{\min})$. 由引理 3.2 可知，$1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 最多有 n_1n_2 个非被主导的调

度方案，在其中找对应目标函数值最小的一个，于是可以得到用算法 3.1 求解问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的计算复杂度为 $O((n_1^2+n_2\log n_2)(Q_2^{\max}-Q_2^{\min})+n_1n_2)$. 综合起来，得到定理的结论. □

接下来考虑问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的两个衍生问题：

$$1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1 \quad 和$$

$$1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_2).$$

在算法 3.1 的基础上结合约束条件作修改(主要在步骤 2 和步骤 4 中修改)就可以得到这两个问题的求解算法.

下面给出求解问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 的算法.

算法 3.2 问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 的求解算法.

步骤 1 计算 $Q_1^{\max}$，$Q_1^{\min}$.

步骤 2 令 $Q_1^*=\min(Q_1,Q_1^{\max})$，调度方案集合 $s=\varnothing$，$i=0$，转步骤 3.

步骤 3 如果使用 Agnetis 等(2004)[233] 的算法能找到问题 $1\parallel f_{\max}^{(2)}(\sigma):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 的最优调度，进而找到问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的一个非被主导调度 σ^i，则问题 $1\parallel f_{\max}^{(2)}(\sigma):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 是可解的，令 $i=i+1$，进入步骤 4. 否则，这时问题 $1\parallel f_{\max}^{(2)}(\sigma):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 是不可解的，进入步骤 5.

步骤 4 令 $s=s\cup\sigma^i$，$Q_1'=f_{\max}^{(1)}(\sigma^i)$，$Q_1=Q_1'-1$，转入步骤 3；如果 $Q_1<Q_1^{\min}$，转入步骤 5.

步骤 5 记录所有的非被主导的调度方案集合 s，计算集合内每个方案对应的目标函数值 $g(f_{\max}^{(1)}(\cdot),f_{\max}^{(2)}(\cdot))$. 具有最小目标函数值 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的调度方案 σ 即为所求的最优调度方案.

类似地，给出求解问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的算法 3.3.

算法 3.3 问题 $1\parallel g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的求解算法.

步骤 1 计算 $Q_2^{\max}$，$Q_2^{\min}$.

步骤 2 令调度方案初始集合 $s=\varnothing$，$i=0$，记 $Q_2^*=\min(Q_2,Q_2^{\max})$，转步骤 3.

步骤 3 如果使用 Agnetis 等(2004)[233] 的算法能够找到问题 $1\parallel f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的最优调度，进而找到问题 $1\parallel f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的一个非被主导调度 σ^i，则问题 $1\parallel f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 是可解的，令 $i=i+1$，进入步骤 4. 否则，这时问题 $1\parallel f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 是不可解的，进入步骤 5.

步骤 4 令 $s=s\cup\sigma^i$，$Q_2'=f_{\max}^{(2)}(\sigma^i)$，$Q_2=Q_2'-1$，转入步骤 3；如果 $Q_2<Q_2^{\min}$，转入步骤 5.

步骤 5 记录所有的非被主导的调度方案集合 s，计算集合内每个方案对应

的目标函数值 $g(f_{\max}^{(1)}(\cdot),f_{\max}^{(2)}(\cdot))$. 具有最小目标函数值 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的调度方案 σ 即为所求的最优调度方案.

可以用类似于分析算法 3.1 的复杂度的分析方法,分别得到求解问题 $1\|g(f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 和问题 $1\|g(f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的计算复杂度.

定理 3.3 求解问题 $1\|g(f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 的计算复杂度为 $O((n_2^2+n_1\log n_1)(Q_1^*-Q_1^{\min})+n_1n_2)$;求解问题 $1\|g(f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的计算复杂度为 $O((n_1^2+n_2\log n_2)(Q_2^*-Q_2^{\min})+n_1n_2)$,其中 $Q_1^*=\min(Q_1,Q_1^{\max})$,$Q_2^*=\min(Q_1,Q_2^{\max})$.

调度问题 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1,f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 在已研究问题 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$ 的基础上增加了约束条件 $f_{\max}^{(1)}(\sigma)\leqslant Q_1,f_{\max}^{(2)}(\sigma)\leqslant Q_2$,在算法 3.1 基础上加入约束条件就可以得到问题的求解算法.

算法 3.4 问题 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1,f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的求解算法.

步骤 1 计算 $Q_i^{\max},Q_i^{\min},i=1,2$.

步骤 2 计算初始的上限 Q^*:

令 $a=\min\{(Q_1^{\max}-Q_1^{\min}),(Q_1-Q_1^{\min})\}$,$b=\min\{(Q_2^{\max}-Q_2^{\min}),(Q_2-Q_2^{\min})\}$,则

$$Q^*=\begin{cases}Q_1, & a\leqslant b,Q_1\leqslant Q_1^{\max},\\ Q_1^{\max}, & a\leqslant b,Q_1>Q_1^{\max},\\ Q_2, & a>b,Q_2\leqslant Q_2^{\max},\\ Q_2^{\max}, & a>b,Q_2>Q_2^{\max}.\end{cases}$$

不失一般性,假设 $Q^*=Q_1$,令 $s=\varnothing,i=0$,进入步骤 3.

步骤 3 如果使用 Agnetis 等(2004)[233]的算法能够找到问题 $1\|f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_1$ 的最优调度,进而找到问题 $1\|f_{\max}^{(1)}\circ f_{\max}^{(2)}$ 的一个非被主导调度 σ^i,则问题 $1\|f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_1$ 是可解的,令 $i=i+1$,进入步骤 4. 否则,这时问题 $1\|f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 是不可解的,进入步骤 5.

步骤 4 令 $s=s\cup\sigma^i,Q_1'=f_{\max}^{(1)}(\sigma^i),Q_1=Q_1'-1$. 如果问题 $1\|f_{\max}^{(2)}(\sigma):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 不可行或者 $Q_1<Q_1^{\min}$,进入步骤 5.

步骤 5 记录非被主导集合 s,计算集合 s 中每个调度方案对应的目标函数值 $g(f_{\max}^{(1)}(\cdot),f_{\max}^{(2)}(\cdot))$,比较得到拥有最小目标函数值的调度方案 σ 即为最优的调度方案. 如果 $s=\varnothing$,则表明问题没有可行的调度方案.

可以利用算法 3.1 计算复杂度的分析方法得到算法 3.4 的计算复杂度.

定理 3.4 如果 $a\leqslant b$,则调度问题 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$,$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 求解的计算复杂度为 $O((n_1^2+n_1\log n_1)a+n_1n_2)$;如果 $a>b$,则调度问题求解的计算复杂度为 $O((n_1^2+n_2\log n_2)b+n_1n_2)$.

证明 由引理 3.2 可知,非被主导调度的目标函数值只有 n_1n_2 种可能值,计算得到这些函数值复杂度为 $O((n_2^2+n_1\log n_1)a+n_1n_2)$.

如果 $a\leqslant b$,需要求解问题 $1\|f_{\max}^{(2)}(\sigma):f_{\max}^{(1)}(\sigma)\leqslant Q_1$. 由引理 3.1 及其推论知道,求解问题 $1\|f_{\max}^{(2)}(\sigma):f_{\max}^{(1)}(\sigma)\leqslant Q_1$ 的计算复杂度为 $O(n_2^2+n_1\log n_1)$,整个循环迭代的次数不超过 a. 因此,问题 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1$,$f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 在 $a\leqslant b$ 时求解的计算复杂度为 $O((n_2^2+n_1\log_1)a+n_1n_2)$.

如果 $a>b$,由引理 3.1 及其推论知道,求解问题 $1\|f_{\max}^{(1)}(\sigma):f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 的计算复杂度为 $O(n_1^2+n_2\log n_2)$. 整个循环迭代的次数不超过 b. 因此,问题 $1\|g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma)):f_{\max}^{(1)}(\sigma)\leqslant Q_1,f_{\max}^{(2)}(\sigma)\leqslant Q_2$ 在 $a\leqslant b$ 时求解的计算复杂度为 $O((n_1^2+n_2\log n_2)b+n_1n_2)$. □

3.5 考虑计件维护的求解算法分析

本节将研究同时考虑两个协同代理和机器计件维护的单机调度,分析其求解算法及计算复杂度. 运用类似于 2.2 节的方法,使得考虑任务的实际处理时间分段函数 $\varphi_j^*(p_j,r)$的协同代理调度问题转化为不含分段函数的任务处理时间模型的协同代理调度问题. 为了简便起见,用 OBJ 表示所考虑的调度问题的通用目标函数,不特指某一特定目标.

引理 3.3 调度问题 $1|p_j^r=\varphi_j^*(p_j,r)$,PRM$=l|$OBJ 与调度问题 $1|$PRM$=l|$OBJ 在 $l\leqslant B$ 时有相同的最优调度.

证明 类似于引理 2.1 的证明,分两种情形讨论和分析.

当 $l\leqslant B$ 时,每处理完 k 项任务,机器就维护到初始状态,所有被安排入序列的任务的实际处理时间不会发生突变(对于任意一个任务 j,其实际处理时间为 $p_j^r=p_j$),对应有限的数值 $f_{\max}^{(1)}(\sigma)$和 $f_{\max}^{(2)}(\sigma)$,相关约束条件在这种情形下才有可能满足,调度决策目标中复合函数 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$也有可能达到最优.

当 $l>B$ 时,由 $B<n$ 可以得到方案中至少有一个任务 j 的排序位置 $r>B$,则有 $p_j^r=+\infty$. 由于调度决策目标 OBJ 中包含关于任务实际处理时间 $f_{\max}^{(1)}(\sigma)$,$f_{\max}^{(2)}(\sigma)$的严格单调增加的复合函数 $g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$,当存在一个任务的实际处理时间 $p_j^r=+\infty$就会有 $f_{\max}^{(1)}(\sigma)$和 $f_{\max}^{(2)}(\sigma)$达到最大值,超出约束条件,$g(f_{\max}^{(1)}(\sigma),f_{\max}^{(2)}(\sigma))$取得最大的可能值,因此此时不会得到问题的最优解.

综合比较 $l>B$ 和 $l\leqslant B$ 两种情形,调度问题 $1|p_j^r=\varphi_j^*(p_j,r)$,PRM=

$l|\mathrm{OBJ}$ 的最优调度只可能产生在 $l\leqslant B$ 的情形中，可能取得最优解，因此与调度问题 $1|\mathrm{PRM}=l|\mathrm{OBJ}$ 在 $l\leqslant B$ 时有相同的最优调度. □

基于引理 3.3，下面在 $l\leqslant B$ 条件下进行讨论和分析. 为了求解以上问题，先考虑一个辅助问题 $1|\mathrm{PRM}=l|f_{\max}$，并基于 Lawler(1973)提出的算法可以改进问题 $1|\mathrm{PRM}=l|f_{\max}$ 的求解算法. 调度问题 $1|\mathrm{PRM}=l|f_{\max}$ 中暂没有考虑两个代理，因此把要处理的任务集记为 $J=\{J_1,J_2,\cdots,J_n\}$，所有任务都要依次在一台机器上进行处理. 任务 J_1 处理时间为一个正整数 p_j，其处理完工时间为 C_j. 给出任务处理完工时间的一个单调不减的函数 $f_j(\cdot)$. 调度问题 $1|\mathrm{PRM}=l|f_{\max}$ 的目标函数可以表示为 $f_{\max}(\sigma)=\max\limits_{1\leqslant j\leqslant n}f_j(C_j(\sigma))$. 机器每处理完 l 个任务就进行维护一次，每次机器维护的时间长度为 t.

基于文献 Lawler(1973)中的算法，下面给出了求解问题 $1|\mathrm{PRM}=l|f_{\max}$ 的求解算法.

算法 3.5　问题 $1|\mathrm{PRM}=l|f_{\max}$ 的求解算法.

步骤 1　在未被安排的任务中选择合适的任务从后面开始安排到调度序列. 令 $\bar{\tau}$ 为被安排的任务所需处理时间之和，s 表示已安排的任务的个数. s 的初始值记为零. 在未被安排的任务选择 $f_k\left(\bar{\tau}+\left[\frac{n-s}{l}\right]t\right)$ 值最小的任务 J_k 排在最后. 通过这种方法，把所有任务逐个安排进调度序列.

步骤 2　从上一步骤得到的任务处理序列，每隔 l 个任务插入一次机器维护.

定理 3.5　算法 3.5 可以得到调度问题 $1|\mathrm{PRM}=l|f_{\max}$ 的一个最优解.

证明　基于 Lawler(1973)[544] 给出了排队定理的证明，这里将给出本定理的证明. 若有 s 个任务尚未安排，则需考虑 s 个任务和 $\left[\frac{n-s}{l}\right]$ 次机器维护组成的序列 σ_1. 在未被安排的任务选择 $f_k\left(\bar{\tau}+\left[\frac{n-s}{l}\right]t\right)$ 值最小的任务 J_k. A 和 B 表示调度序列中除了任务 J_k 和 $J_{k'}$ 之外的部分序列，调度序列 σ_1 可以通过图 3-2 表示.

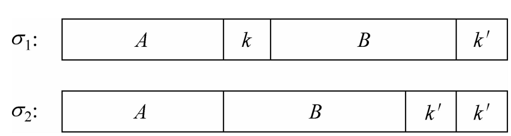

图 3-2　调度问题 $1|\mathrm{PRM}=l|f_{\max}$ 的序列调整示意图