运筹学与控制论

DOI:10.11721/cqnuj20190614

工件可外包的单机准时排序问题*

李寒雪，樊保强，陈继文，郭志佳，杨燕英，李　欣
(鲁东大学 数学与统计科学学院，山东 烟台 264000)

摘要:【目的】研究一类单机准时排序问题，其中工件有公共的交货期和交货截止期，允许工件外包加工，外包加工将产生外包费用，目标是极小化总提前时间，总延迟时间与总外包费用之和。【方法】首先给出该问题的若干最优性性质，然后对于交货期和交货截止期都待定，以及给定交货截止期两种情形分别讨论。【结果】对于第一种情形，设计了多项式时间算法，对于第二种情形，证明了它是 NP-困难的并设计了伪多项式时间的动态规划算法。【结论】所讨论的单机准时排序问题所得到的结果为冷鲜食品生产管理者提供了有效决策支持。

关键词:准时排序;动态规划;外包费用;算法

中图分类号:O223　　**文献标志码**:A　　**文章编号**:1672-6693(2019)06-0015-07

1 研究背景

客户服务水平和生产成本控制问题是制造行业中普遍存在的问题，特别是在冷鲜肉加工行业中，由于产品的易逝性，即维持产品品质的周期短且销售周期也短，导致在生产和配送等供应链的各个阶段，对于产品时效性有更高的要求。另一方面，由于市场需求在时间上的高度一致性，导致一些企业不得不考虑外包加工客户工件，来提高客户服务水平。本文考虑一类来源于冷鲜肉加工业的工件可外包的单机准时排序问题，工件有公共的交货期和交货截止期，所有工件必须在交货截止期前加工完毕，允许工件外包加工，外包加工将产生外包费用。目标是确定外包工件集合，对于没有外包的工件给定加工顺序，使得总提前时间、总延误时间和总外包费用的加权和最小。

Emmons[1]最早研究了总延误问题。Lawler[2]给出了总延误问题一个运行时间为 $O\left(n^4 \sum p_i\right)$ 的伪多项式时间算法。基于该算法 Lawler[3]设计了一个 FPTAS 近似方案。Du 等人[4]证明了总延误问题是 NP-困难的。对于加权总延误问题，Lestra 等人[5]证明该问题是强 NP-困难的。对于有公共交货期的加权总延误问题，Lawler 等人[6]给出了一个时间为 $O(n^2 d)$ 的伪多项式时间算法，直到 1992 年，Yuan[7]证明了该问题是 NP-困难的。关于极小化总提前和总延误之和的准时排序问题，Hall 等人[8]证明了它是 NP-困难的，并给出了一个时间为 $O\left(n \sum p_i\right)$ 的伪多项式时间算法。

Chen 等人[9]较早研究了工件课外包的排序问题，其中每个工件可以在制造商机器上加工，又可以转包给若干承包商之一进行加工，目标是在保证工件最大完工时间不超过给定值的情况下，极小化加工与转包费用和。Chen 等人[10]研究了单机工件可转包排序，目标是极小化排序目标(最大完工时间或误工工件数)与加工、转包费用和。Chen 等人[11]研究了 m 台平行机环境下的外包排序问题，考虑了外包费用、运输延迟以及运输费用。工件可拒绝排序是一类与工件可外包排序类似的排序问题。Bartal 等人[12]首先研究了工件可拒绝平行机排序，目

* **收稿日期**:2019-01-30　　**修回日期**:2019-10-26　　**网络出版时间**:2019-11-25 10:34

资助项目:国家自然科学基金(No. 11801251)

第一作者简介:李寒雪，女，研究方向为排序论，E-mail:1049975558@qq.com;**通信作者**:樊保强，男，副教授，博士，E-mail:baoqiangfan@163.com

网络出版地址:http://kns.cnki.net/kcms/detail/50.1165.N.20191125.1034.016.html

标是极小化最大工件完工时间与总拒绝费用之和。对于工件可拒绝单机排序,当目标是极小化总加权完工时间与总拒绝费用之和时,Engels 等人[13]考虑了证明该问题是 NP-困难的,并给出了一个 FPTAS 算法,对于工件权重恒等情形,给出了多项式时间算法。对于工件有就绪时间,极小化最大工件完工时间与总拒绝费用之和问题,Zhang 等人[14]证明了该问题是 NP-难的,给出了两个伪多项式时间算法,一个 2-近似算法和一个 FPTAS 算法。

本文考虑工件可外包,目标是极小化总提前、总延误和总外包费用的加权和的单机准时排序问题。第二节将给出问题的详细描述以及常用符号。第三节对于共同交货期和交货截止期的不同情形,分别讨论了它们的计算复杂性,给出了基于动态规划的有效算法。第四节是本文结论部分。

2 问题与符号

有 n 个工件 $N=\{1,2,\cdots,n\}$ 将在一台机器上加工,每个工件有一个加工时间 p_i,如果该工件不安排加工,则将外包给其他厂商加工,外包费用为 q_i。所有工件有一个公共交货期 d,一个最后交货期限 $\overline{d}$,有 $d<\overline{d}$。令 A 表示接受加工的工件集合,$\overline{A}$ 表示外包工件集合,则 $A\cup\overline{A}=N$。σ 表示 A 中工件的一个排序。本文将用到以下符号:$C_i(\sigma)$ 表示工件 J_i 在排序 σ 中的完工时间,简记为 C_i;$E_i(\sigma)$ 表示 J_i 的提前时间,$E_i(\sigma)=\max\{0,d-C_i(\sigma)\}$,简记为 E_i;$T_i(\sigma)$ 表示 J_i 的延误时间,$T_i(\sigma)=\max\{0,C_i(\sigma)-d\}$,简记为 T_i;$\sum_{i\in A}E_i$ 表示 A 中工件的总提前时间;$\sum_{i\in A}T_i$ 表示 A 中工件的总延误时间;$\sum_{i\in\overline{A}}q_i$ 表示 $\overline{A}$ 中工件的总外包费用。

问题是确定外包的工件集合 $\overline{A}$,即确定加工的工件集合 A,并给出 A 中工件的加工顺序,使总提前时间、总延误时间和总外包费用之和最小。该问题记为 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$,其中 α 和 β 分别表示所有接受工件和外包工件的权重。假设工件加工过程不允许中断,并且以上数据均为整数。

3 问题 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$

本节讨论问题 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$。对于 d 和 $\overline{d}$ 都待定的情形,设计多项式时间算法;对于给定 $\overline{d}$ 的情形,证明该问题是 NP-困难的,最后设计了伪多项式时间的动态规划算法。

对于经典准时排序问题 $1|d_i=d|\sum E_i+\sum T_i$,唐国春等人[15]给出了若干最优解性质。对于 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$,下面将给出类似的性质,在此之前先给出关键工件 J_k 的定义。

定义 1 设 σ 是 A 中工件的一个排序,如果存在工件 J_k,有 $C_k-p_k<d<C_k$,则称工件 J_k 为排序 σ 的关键工件。

注 对于 A 中工件的一可行排序,如果有工件在 d 时刻完工,那么该排序不存在关键工件。

引理 1 对于问题 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$,存在最优解满足下列条件:

1) 接受加工工件集合 A 中任意两个工件之间,机器无空闲。

2) 交货期 d 前(包括时刻)完工的工件是按 LPT 序加工,交货期 c 之后开始加工的工件是按 SPT 序加工。

根据引理 1,接受加工的工件集合 A 被分成了 3 部分,令 S_1 表示在 d 之前(包括时刻)完工的工件集合,S_2 表示在 d 之后(包括 d 时刻)开始加工的工件集合,关键工件 J_k(如果存在)在 d 之前开始加工,d 之后完工。这里 $S_1\subseteq A$,$S_2\subseteq A$,$A=S_1\cup S_2\cup\{k\}$,并且 S_1 中的工件按 LPT 序加工,S_2 中的工件按 SPT 序加工。称满足引理 1 条件的可行解 $\sigma=(S_1,J_k,S_2)$ 为 V-型解。下文只考虑 V-型可行解。

3.1 d 和 $\overline{d}$ 待定的情形

当共同交货期 d 待定时,问题 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$ 有下面关键的最优解性质。

引理 2 对于问题 $1|d,\overline{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\overline{A}}q_i$,当共同交货期 d 待定时,存在 V-型最优解,其中不存在

关键工件，即 S_1 的完工时间为 d。

首先，根据加工时间给工件进行编号，使 $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$。由引理 2，令 $f_i(n_1, n_2)$ 表示前 i 个工件的最优 V-型解的目标值，其中：1）所有接受加工的工件由 S_1 和 S_2 组成；2）n_1 为 S_1 中包含的工件个数，n_2 为 S_2 中包含的工件个数。记 $f_i(n_1, n_2)$ 对应 i 阶段的状态变量为 (i, n_1, n_2)。则状态 (i, n_1, n_2) 是在 $i-1$ 阶段的状态 (i', n_1', n_2') 基础上，通过对工件 J_i 执行以下某个决策得到：

1）$i \in A, i \in S_1$，即把 J_i 放在 d 之前加工。因为 S_1 中的工件按 LPT 顺序加工，所以 J_i 为 S_1 中最后一个加工的工件，因此 $C_i = d$。由此产生的费用是 $\alpha(n_1 - 1)p_i$。

2）$i \in A, i \in S_2$，即把 J_i 放在 d 之后加工。因为 S_2 中的工件为 SPT 序，所以 J_i 为 S_2 中第一个加工的工件，因此 $C_i = d + p_i$。由此产生的费用是 $\alpha n_2 p_i$。

3）$i \in \overline{A}$，即工件 J_i 外包加工。因此产生的费用是 βq_i。

根据以上分析，下面给出问题 $1 | d, \overline{d} | \alpha \sum_{i \in A}(E_i + T_i) + \beta \sum_{i \in \overline{A}} q_i$ 的动态规划算法。

算法 DP1　初始条件：$f_0(n_1, n_2) = \begin{cases} 0, n_1 = n_2 = 0 \\ \infty, \text{其他} \end{cases}$；

迭代方程：对于 $i = 1, 2, \cdots, n$，有 $f_i(n_1, n_2) = \min \begin{cases} \alpha(n_1 - 1)p_i + f_{i-1}(n_1 - 1, n_2) \\ \alpha n_2 p_i + f_{i-1}(n_1, n_2 - 1) \\ \beta q_i + f_{i-1}(n_1, n_2) \end{cases}$；

最优值：$\min\{f_n(n_1, n_2) | 0 \leqslant n_1, n_2 \leqslant n\}$。

动态规划 DP1 共有 n 个阶段。因为 $n_1 \leqslant n, n_2 \leqslant n$，所以每个阶段的运行时间不超过 $O(n^2)$。那么 DP1 的时间复杂度为 $O(n^3)$。通过回溯法可以得到最优解。那么 $d = \sum_{i \in S_1} p_i$ 和 $\overline{d} = \sum_{i \in A} p_i$。

定理 1　当 d 和 $\overline{d}$ 待定时，问题 $1 | d, \overline{d} | \alpha \sum_{i \in A}(E_i + T_i) + \beta \sum_{i \in \overline{A}} q_i$ 可以由 DP1 在 $O(n^3)$ 时间内给出最优解。

3.2 $\overline{d}$ 给定的情形

本节对于 $\overline{d}$ 给定的情形，证明了问题 $1 | d, \overline{d} | \alpha \sum_{i \in A}(E_i + T_i) + \beta \sum_{i \in \overline{A}} q_i$ 是 NP-困难问题，然后就 d 是否待定分别设计了伪多项式时间的动态规划算法。

定理 2　问题 $1 | d, \overline{d} | \alpha \sum_{i \in A}(E_i + T_i) + \beta \sum_{i \in \overline{A}} q_i$ 是 NP-困难的。

证明　把 NP-困难的背包问题[16]变换到问题 $1 | d, \overline{d} | \alpha \sum_{i \in A}(E_i + T_i) + \beta \sum_{i \in \overline{A}} q_i$ 的一个特殊情形：$\alpha = 0, \beta = 1$。令 $\sum_{i \in A} p_i$ 表示接受工件的总加工时间，$\sum_{i \in A} q_i = \sum_{i \in \mathbf{N}} q_i - \sum_{i \in \overline{A}} q_i$。因此该情形等价于在 $\sum_{i \in A} p_i \leqslant \overline{d}$ 的约束下，极大化 $\sum_{i \in A} q_i$ 的优化问题。显然该问题等价于背包问题，因此是 NP-困难的。　证毕

下面给出 d 待定时，$1 | d, \overline{d} | \alpha \sum_{i \in A}(E_i + T_i) + \beta \sum_{i \in \overline{A}} q_i$ 的伪多项式时间动态规划算法。

由引理 2，S_1 中的工件在时刻 d 完工。类似于算法 DP1 中的定义，令 $f_i(n_1, n_2, t)$ 表示前 i 个工件的最优 V-型解的目标值，其中 n_1 为 S_1 中包含的工件个数，n_2 为 S_2 中包含的工件个数；t 表示 A 中所有工件的加工时间和 $t = \sum_{i \in A} p_i$。

算法 DP2　初始条件：$f_0(n_1, n_2, t) = \begin{cases} 0, n_1 = n_2 = t = 0 \\ \infty, \text{其他} \end{cases}$；

迭代方程：对于 $i = 1, 2, \cdots, n$，有：

$$f_i(n_1, n_2, t) = \min \begin{cases} \alpha(n_1 - 1)p_i + f_{i-1}(n_1 - 1, n_2, t - p_i), t \leqslant \overline{d} \\ \alpha n_2 p_i + f_{i-1}(n_1, n_2 - 1, t - p_i), t \leqslant \overline{d} \\ \beta q_i + f_{i-1}(n_1, n_2, t) \end{cases};$$

最优值：$\min\left\{f_n(n_1,n_2,t)\middle|0\leqslant n_1,n_2\leqslant n,0\leqslant t\leqslant\sum p_i\right\}$。

动态规划DP2共有n个阶段。因为$n_1\leqslant n,n_2\leqslant n,t\leqslant\sum p_i$，所以每个阶段的运行时间不超过$O\left(n^2\sum p_i\right)$。那么DP2的时间复杂度为$O\left(n^3\sum p_i\right)$。通过回溯法可以得到最优解，则有$d=\sum_{i\in S_1}p_i$。

定理3 当d待定时，$1|d,\bar{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\bar{A}}q_i$可以由DP2在$O(n^3\sum p_i)$时间内给出最优解。

现在考虑d给定的情形。首先给出$1|d,\bar{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\bar{A}}q_i$的一些最优解性质。

引理3 对于问题$1|d,\bar{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\bar{A}}q_i$，当共同交货期$d$和截止日期$\bar{d}$给定时，存在V-型最优解$\sigma=(S_1,J_k,S_2)$，满足下列条件之一：1) 关键工件$J_k$不存在，即$S_1$中的工件在$d$时刻完工；2) $d<C_k<d+p_k$，S_1在0时刻开始加工；3) $d<C_k<d+p_k$，S_2在$\bar{d}$时刻完工。

关于引理3中的关键工件J_k有下面的结论。

引理4 满足引理3中条件的关键工件J_k，如果$d<C_k<d+p_k$，则$p_k\leqslant\min\{p_i|i\in S_1\}$或$p_k\leqslant\min\{p_i|i\in S_2\}$。

证明 设$\sigma=(S_1,J_k,S_2)$是一最优V-型解。不失一般性，设$p_r=\min\{p_i|i\in S_1\}$，$r\in N$，$p_h=\min\{p_i|i\in S_2\}$，$h\in N$。由引理1知，在σ中J_r是J_k紧前工件，J_h是J_k紧后工件。下面用反证法来证明该引理结论。

假设$p_k>\min\{p_i|i\in S_1\}$且$p_k>\min\{p_i|i\in S_2\}$。注意到时刻d将J_k分为两部分，d时刻之前记为$p_{k,1}$，d之前记为$p_{k,2}$。下面按$p_{k,1}$和$p_{k,2}$的大小关系分两种情况讨论。

1) $p_{k,2}\geqslant p_{k,1}$。

i) 若$p_{k,1}\geqslant p_h$，则交换p_k与p_h，得到一新排序，它的目标函数值将减小$p_{k,2}-p_{k,1}+p_h$。显然$p_{k,2}-p_{k,1}+p_h>0$，此结论与σ为最优解矛盾。

ii) 若$p_{k,1}<p_h$，则交换p_k与p_h，得到一新排序，它的目标值将减小p_k-p_h。因为$p_k>p_h$，所以$p_k-p_h>0$与σ为最优解矛盾。

2) $p_{k,2}<p_{k,1}$。

i) 若$p_{k,1}\geqslant p_r$，则交换p_k与p_r，得到一新排序，它的目标值将减小$p_{k,1}-p_{k,2}+p_r$。易知$p_{k,1}-p_{k,2}+p_r>0$，此结论与σ为最优解矛盾。

ii) 若$p_{k,1}<p_r$，则交换p_k与p_r，得到一新排序，它的目标值将减小p_k-p_r，因为$p_k>p_r$，所以$p_k-p_r>0$与σ为最优解矛盾。

综上所述，假设不成立，因此有$p_k\leqslant\min\{p_i|i\in S_1\}$或$p_k\leqslant\min\{p_i|i\in S_2\}$。 证毕

下面对引理3中的3种情形分别进行讨论。

情形1 关键工件J_k不存在，即S_1中的工件在时刻d完工。

根据工件加工时间给工件进行编号，使$p_1\leqslant p_2\leqslant\cdots\leqslant p_n$。

令$f_i(t_1,t_2)$表示前i个工件的最优V-型解的目标值，其中t_1为S_1中工件总加工时间，即$t_1=\sum_{i\in S_1}p_i$；t_2为S_2中工件总加工时间，即$t_2=\sum_{j\in S_2}p_j$。$f_i(t_1,t_2)$对应i阶段的状态(i,t_1,t_2)。则状态(i,t_1,t_2)是在$i-1$阶段的状态(i',t_1',t_2')基础上，通过对工件J_i执行以下某个决策得到：

1) $i\in A,i\in S_1,t_1\leqslant d$。因为$S_1$中的工件按LPT顺序加工，所以$J_i$为$S_1$中第一个加工的工件，对其他工件的完工时间没有影响，由此产生的费用是$\alpha(t_1-p_i)$。

2) $i\in A,i\in S_2,t_2\leqslant\bar{d}-d$。因为$S_2$中的工件为SPT序，所以$J_i$为$S_2$中最后一个加工的工件，对其他工件的完工时间也没有影响。由此产生的费用是αt_2。

3) $i\in\bar{A}$，即工件J_i外包加工。因此产生的费用是βq_i。

根据以上分析，下面给出情形 1 的动态规划算法。

算法 DP3　初始条件：$f_0(t_1,t_2)=\begin{cases}0, t_1=t_2=0\\ \infty,\text{其他}\end{cases}$；

迭代方程：对于 $i=1,2,\cdots,n$，有：

$$f_i(t_1,t_2)=\min\begin{cases}\alpha(t_1-p_i)+f_{i-1}(t_1-p_i,t_2), t_1\leqslant d\\ \alpha t_2+f_{i-1}(t_1,t_2-p_i), t_2\leqslant \bar{d}-d\\ \beta q_i+f_{i-1}(t_1,t_2)\end{cases};$$

最优值：$\min\left\{f_n(t_1,t_2)\,\middle|\,0\leqslant t_1,t_2\leqslant \sum p_i\right\}$。

动态规划 DP3 共有 n 个阶段。因为 $t_1\leqslant \sum p_i$，$t_2\leqslant \sum p_i$，所以第 i 阶段运行时间不超过 $O\left(\left(\sum p_i\right)^2\right)$。因此，DP3 的时间复杂度为 $O\left(n\left(\sum p_i\right)^2\right)$。

定理 4　对于不存在关键工件的情形，$1|d,\bar{d}|\alpha\sum\limits_{i\in A}(E_i+T_i)+\beta\sum\limits_{i\in\bar{A}}q_i$ 可以由 DP3 在 $O\left(n\left(\sum p_i\right)^2\right)$ 时间内给出最优解。

情形 2　$d<C_k<d+p_k$，且 S_1 在 0 时刻开始加工。

考虑 V-型可行解 $\sigma=(S_1,J_k,S_2)$。由引理 4，如果 $p_k\leqslant\min\{p_i|i\in S_1\}$，则令 $S_1'=S_1\cup\{k\}$，$S_2'=S_2$。若 $p_k\leqslant\min\{p_i|i\in S_2\}$，则令 $S_1'=S_1$，$S_2'=S_2\cup\{k\}$。令 $d'=d+s$ 为 S_2' 的开始加工时间，这里 $0<|s|\leqslant p_{\max}$，$p_{\max}=\max\{p_i|i\in N\}$。则 S_1' 表示在 d' 之前包括在 d' 时刻完工的工件集合，S'_2 表示在 d' 之后包括在 d' 时刻开始加工的工件集合。

下面对于给定的 s，令 $d'=d+s$，$f_i^s(t_1,t_2)$ 表示前 i 个工件的最优 V-型解的目标值。调用算法 DP3，在执行 DP3 过程中，d' 替换 d，并分别记录 S_1' 和 S_2' 中的工件个数 n_1 和 n_2。则有：

$$f_i^s(t_1,t_2)=\begin{cases}f_i(t_1,t_2)+(n_2-n_1+2)s, s>0\\ f_i(t_1,t_2)+(n_2-n_1)s, s<0\end{cases}.$$

此时，情形 2 的最优值是 $\min\left\{f_n^s(t_1,t_2)\,\middle|\,t_1=d+s,0\leqslant t_2\leqslant\sum p_i,0<|s|\leqslant p_{\max}\right\}$。

情形 3　$d<C_k<d+p_k$，且 S_2 在 $\bar{d}$ 时刻完工。

考虑 V-型可行解 $\sigma=(S_1,J_k,S_2)$。类似于情形 2 的讨论，定义 S_1'，S_2'。令 $d'=d+l$ 为 S_1' 的完工时间，$0<|l|\leqslant p_{\max}$。则 S_1' 表示在 d' 之前包括在 d' 时刻完工的工件集合，S_2' 表示在 d' 之后包括在 d' 时刻开始加工的工件集合。

下面对于给定的 l，令 $d'=d+l$，$f_i^l(t_1,t_2)$ 表示前 i 个工件的最优 V-型解的目标值。调用算法 DP3，在执行 DP3 过程中，d' 替换 d，并分别记录 S_1' 和 S_2' 中的工件个数 n_1 和 n_2。则有：

$$f_i^l(t_1,t_2)=\begin{cases}f_i(t_1,t_2)+(n_2-n_1+2)l, l>0\\ f_i(t_1,t_2)+(n_2-n_1)l, l<0\end{cases}.$$

此时，情形 3 的最优值是 $\min\left\{f_n^l(t_1,t_2)\,\middle|\,t_2=\bar{d}-(d+l),0\leqslant t_1\leqslant\sum p_i,0<|l|\leqslant p_{\max}\right\}$。

综上所述，现给出问题 $1|d,\bar{d}|\alpha\sum\limits_{i\in A}(E_i+T_i)+\beta\sum\limits_{i\in\bar{A}}q_i$ 在 d 和 $\bar{d}$ 给定时的有效算法。

算法 1　第 1 步：根据工件加工时间给工件进行编号，使 $p_1\leqslant p_2\leqslant\cdots\leqslant p_n$。

第 2 步：调用算法 DP3，得到排序 σ_1。

第 3 步：对于给定的 $s(0<|s|\leqslant p_{\max})$，针对新的交货期 $d'=d+s$，调用算法 DP3，在执行 DP3 过程中，记录 S_1' 和 S_2' 的工件个数分别为 n_1 和 n_2，令：

$$f_i^s(t_1,t_2)=\begin{cases}f_i(t_1,t_2)+(n_2-n_1+2)s, s>0\\ f_i(t_1,t_2)+(n_2-n_1)s, s<0\end{cases}$$

得到排序 σ_2，使得 σ_2 的目标函数值为 $\min\left\{f_n^s(t_1,t_2)\,\middle|\,t_1=d+s,0\leqslant t_2\leqslant\sum p_i,0<|s|\leqslant p_{\max}\right\}$。

第4步：对于给定的 $l(0<|l|\leqslant p_{\max})$，这对新的交货期 $d'=d+l$，调用算法DP3，在执行DP3过程中，记录 S_1' 和 S_2' 的工件个数分别为 n_1 和 n_2，令：

$$f_i^l(t_1,t_2)=\begin{cases}f_i(t_1,t_2)+(n_2-n_1+2)l,l>0\\ f_i(t_1,t_2)+(n_2-n_1)l,l<0\end{cases}$$

得到排序 σ_3，使得 σ_3 的目标函数值为 $\min\left\{f_n^l(t_1,t_2)\,\middle|\,t_2=\bar{d}-(d+l),0\leqslant t_1\leqslant\sum p_i,0<|l|\leqslant p_{\max}\right\}$。

第5步：记 σ_1，σ_2 和 σ_3 中目标函数值最小的排序为 σ。

算法1的时间复杂度为 $O\left(n\left(\sum p_i\right)^2 p_{\max}\right)$。

定理5 当 d 和 $\bar{d}$ 给定时，$1|d,\bar{d}|\alpha\sum_{i\in A}(E_i+T_i)+\beta\sum_{i\in\bar{A}}q_i$ 可以由算法1在 $O\left(n\left(\sum p_i\right)^2 p_{\max}\right)$ 时间内给出最优解。

下面通过例子对算法1进行验证。

首先，令 $\alpha=1,\beta=1,d=10,\bar{d}=17$，工件集合 $N=\{1,2,3,4\}$，参数见表1。

首先根据工件加工时间从小到大的顺序给工件进行编号为 n_2,n_4,n_3,n_1。如果一一计算，版面过于大，现只呈现一部分计算过程及结果(每次都选取将工件放在交货期之前的位置进行计算)。

$$f_0(0,0)=0,f_1(t_1,t_2)=\min\begin{cases}f_1(2,0)=0\\ f_1(0,2)=2\\ f_1(0,0)=1\end{cases},f_2(t_1,t_2)=\min\begin{cases}f_2(7,0)=2\\ f_2(2,5)=5\\ f_2(2,0)=2\end{cases},$$

$$f_3(t_1,t_2)=\min\begin{cases}f_3(13,0)=7\\ f_3(7,6)=8\\ f_3(7,0)=6\end{cases},f_4(t_1,t_2)=\min\begin{cases}f_4(21,0)=13\\ f_4(13,8)=15\\ f_4(13,0)=13\end{cases}。$$

表1 参数

Tab.1 Parameters

参数类别	取值			
n_i	1	2	3	4
p_i	8	2	6	5
q_i	6	1	4	2

同理可得其他结果，最终排序工件 n_2,n_3,n_4 外包，而工件 n_1 在机器上加工。因此，极小化总提前时间，总延迟时间与总外包费用之和为7。

4 结论

本文研究了工件可外包的单机准时排序问题，目标是确定外包工件集合，对于没有外包的工件给定加工顺序，使得总提前时间、总延误时间和总外包费用的加权和最小。对于 d 和 $\bar{d}$ 都待定的情形，设计了多项式时间算法；对于给定 $\bar{d}$ 的情形，证明该问题是NP-困难的，最后针对一般情形设计了两个伪多项式时间的动态规划算法。对于该问题，今后可以在以下几个方面进行深入研究：

1) 既然该问题是NP-困难的，可以考虑设计有效的基于连续优化技术的近似算法；

2) 考虑其他目标函数，例如，误工工件数和总外包费用的加权和等；

3) 对于 d 和 $\bar{d}$ 都待定的情形，将进一步研究具有可转移特征函数的排序博弈问题，为决策者制定更合理的订单加工次序，设计科学合理的分配方案，提高联盟的凝聚力，达到社会收益最大。

参考文献：

[1] EMMONS H. One-machine sequencing to minimize certain functions of job tardiness[J]. Operations Research, 1969, 17(4): 701-715.

[2] LAWLER E L. A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness[J]. Annals of Discrete Mathematics, 1977, 1(8): 331-342.

[3] LAWER E L. A fully polynomial approximation scheme for the total tardiness problem[J]. Operations Research Letters, 1982, 1(6): 207-208.

[4] DU J Z, LEUNG J Y T. Minimizing total tardiness on one single machine is NP-hard[J]. Mathematics of Operations Research, 1990, 15(3): 483-495.

[5] LENSTRA J K, RINNOOY KAN A H G, BRUCKER P. Complexity of machine scheduling problems[J]. Annals of Discrete Mathematics, 1977, 1: 343-362.

[6] LAWER E L, MOORE J M. A functional equation and its application to resource allocation and sequencing problems [J]. Management Science, 1969, 16(1): 77-84.

[7] YUAN J J. The NP-hardness of the single machine common due date weighted tardiness problem[J]. Systems Science and Mathematics Sciences, 1992, 5(4): 328-333.

[8] HALL N G, POSNER M E. Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date[J]. Operations Research, 1991, 39(5): 836-846.

[9] CHEN Z L, LI C L. Scheduling with subcontracting options [J]. IIE Transactions, 2008, 40: 1171-1184.

[10] 陈荣军,唐国春.单机上的排序与转包问题[J].应用数学学报,2017,40(2):170-178.

CHEN R J, TANG G C. Scheduling with outsourcing on single machine[J]. Acta Mathematicae Applicatae Sinica, 2017, 40(2): 170-178.

[11] CHEN L Z, YANG D, ZHANG Y Z. The outsourcing problem in supply chain scheduling [J]. Journal of Chongqing Normal University (Natural Science), 2013, 30 (5): 21-26.

[12] BARTAL Y, LEONARDI S, SPACCAMELA A M, et al. Multiprocessor scheduling with rejection[J]. SIAM Journal on Discrete Mathematics, 2000, 13(1): 64-78.

[13] ENGELS D W, KARGER D R, KOLLIOPOULDS S G, et al. Techniques for scheduling with rejection[J]. Journal of Algorithms, 2003, 49(1): 175-191.

[14] ZHANG L Q, LU L F, YUAN J J. Single machine scheduling with release dates and rejection[J]. European Journal of Operational Research, 2009, 198(3): 975-978.

[15] 唐国春,张峰,罗收成,等.现代排序论[M].上海:上海科学普及出版社,2003.

TANG G C, ZHANG F, LUO S C, et al. Modern scheduling theory[M]. Shanghai: Shanghai Science Popularization Publishers, 2003.

[16] GÜNTZER M M, JUNGNICKEL D. Approximate minimization algorithms for the 0/1 knapsack and subset-sum problem[J]. Operations Research Letters, 2000, 26(2): 55-66.

Operations Research and Cybernetics

Single-Machine Just-in-Time Scheduling with Outsourcing

LI Hanxue, FAN Baoqiang, CHEN Jiwen, GUO Zhijia, YANG Yanying, LI Xin

(School of Mathematics and Statistics, Ludong University, Yantai Shandong 264000, China)

Abstract: [**Purposes**]Motivated by the time-sensitive production process, a single-machine just-in-time scheduling problem with outsourcing is studied. In such a problem, there is a common due date and a deadline. A job is either outsourced to process, in which case an outsourcing cost has to be paid, or processed on the machine. The objective is to minimize the sum of the total earliness, the total tardiness and the total outsourcing cost. [**Methods**] First, some optimality properties of the problem are given, and then the delivery deadline and delivery deadline are discussed respectively. [**Findings**] For the first case, the polynomial time algorithm is designed. For the second case, it is proved to be NP hard and a dynamic programming algorithm of pseudo polynomial time is designed. [**Conclusions**] The results of the single machine on-time scheduling problem provide effective decision support for cold and fresh food production managers.

Keywords: scheduling; dynamic programming; outsourcing; algorithm

(责任编辑 黄 颖)