考虑设备周期性维护的单机调度优化算法

吴玉洁，谢 勇
（华中科技大学自动化学院，武汉 430074）

摘要：为解决实际生产中生产调度与设备维护相互影响的问题，建立一种综合考虑单机调度和周期性预防维护的集成优化模型。针对模型的特点，提出一种两阶段启发式算法（two-stage heuristic algorithm，TSHA），先基于批次的最优排序规则，构建批次的初始调度序列，再通过对批次松弛时间的充分利用，在不增加最大拖期的前提下，使最大拖期工件前移或使最大拖期工件的开工时间提前，改善批次的调度序列。通过计算实验，与 CPLEX 最优解及已有启发式算法的解作对比，结果表明，TSHA 性能更加优异，能有效解决工件不可中断情况下的以周期性维护为资源约束的单机调度问题。

关键词：系统工程；单机调度；周期性维护；启发式算法
中图分类号：TP202+.7　　**文献标识码：**A　　**文章编号：**1674-2850(2017)10-1179-10

Single-machine scheduling optimization algorithm with periodic maintenance

WU Yujie, XIE Yong
(*School of Automation*, *Huazhong University of Science & Technology*, *Wuhan* 430074, *China*)

Abstract: In order to solve the interation problem between the production scheduling and equipment maintenance, an integrated optimization model was proposed. A two-stage heuristic algorithm (TSHA) was proposed to find an optimal schedule. In the first stage, an initial solution was obtained with the optimal batch ordering rule. In the second stage, through making full use of the batch relaxation time, under the premise of not increasing the maximum tardiness, the reach maximum tardiness workpiece or the starting time of the maximum tardiness workpiece was made in advance to improve the batch scheduling sequence. The performance of the TSHA was evaluated by comparing the solutions with those obtained from CPLEX and an existing heuristic algorithm. The experimental results show that the TSHA can solve the single-machine scheduling problem effectively subject to periodic maintenance.
Key words: systems engineering; single-machine schecluling; periodic maintenance; heuristic algorithm

0 引言

生产和维护计划的合理制定不仅可以提升企业的经济效益，还可以增强企业的市场竞争力。生产调度与设备维护之间存在耦合关系，生产调度作业会使机器负荷发生变化，降低机器的可靠性和稳定性；而设备维护虽然抢夺了机器的运行时间，但是保证了机器长时间的稳定运行。近年来，生产调度和设备维护的联合优化问题受到越来越多学者的关注。

根据维护时间段是否预先已知将这类问题分为两类，一类是维护时间点在调度之前已经确定的调度问题，另一类是维护时间点在调度之前不可确定的调度问题。在第一类的研究中，MANE 等[1]针对单机调度期间仅有一次维护活动的情况，以最小化提前、拖期、截止期总费用为目标，提出了一种多项式时

作者简介：吴玉洁（1993—），女，硕士研究生，主要研究方向：供应链管理、物流管理
通信联系人：谢勇，副教授，主要研究方向：供应链管理、物流管理. E-mail: xyhust@163.com

间算法确定最优的维护位置、公共交货期及加工序列。WAN[2]以炼钢工业的应用为实例，将其生产计划和维护计划的制定抽象为工件加工时间相同的单机调度与单次维护的集成优化问题并求解。CHEN[3]对单机调度和周期性预防维护进行了集成研究，以最小化拖期工件数量为优化目标，提出了一种基于 Moore 算法的启发式算法以获得最优工件加工顺序。LEE 等[4]研究了相同的问题，并建立了该问题的混合整数规划模型。SBIHI 等[5]以最小化最大拖期为目标函数，提出一种将分支定界法和启发式算法相结合的方法。刘碧玉等[6]针对单机生产系统中部分工件加工过程可中断和部分工件加工过程不可中断的特殊情况，提出了一种 LPT-LS 算法来安排工件的加工顺序，并分析该问题的最坏情况比。张思源等[7]将问题延伸至流水线车间，在考虑周期性维护的基础上，分别建立置换车间与非置换车间两种不同情形下的数学优化模型，并提出了一种混合遗传算法对问题进行优化求解。在第二类的研究中，LOW 等[8]考虑了弹性时间窗维护和工具更换维护共同约束下的单机调度问题，指出以最小化最大完工时间为目标时，该问题可以转化为离线的一维装箱问题。蒋志高[9]首次提出虚拟维护的概念，解决并维护具有柔性的时间窗是工件加工带有学习效应的单机调度问题。

对单机调度和设备维护的集成优化问题的研究已经有了部分成果，但还不够深入和全面，有待进一步补充。本文将对周期性维护和单机调度的集成优化问题进行深入研究，提出一种比已有成果更加优化的算法。

1 周期性维护下的单机调度问题建模

1.1 问题描述

假设有 n 个工件 $\{J_{[1]},J_{[2]},\cdots,J_{[n]}\}$ 在同一台机器上等待加工，每个工件都有各自的加工时间 p_j 和交货日期 d_j. 在加工过程中，机器并非一直持续可用，需要定期进行维护，维护活动只在固定时间段内进行，即连续两次维护活动之间的时间间隔是固定的，如图 1 所示。这个固定的时间段在工件调度之前就确定已知，因此可以通过机器不可用时间段的设立来描述维护计划对生产调度的影响。

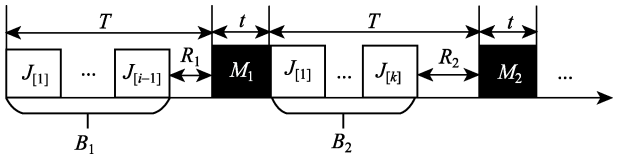

图 1 考虑周期性维护的单机调度

Fig. 1 Single-machine scheduling with periodic maintenance

按照 GRAHAM 等[10]提出的三阶段表示法，该问题可被表示为 $1|\text{npmpt}-\text{pm}|T_{\max}$，其中，1 表示机器类型为单机；npmpt 表示加工过程中工件不可中断；pm 表示加工过程受周期性预防维护的影响；$T_{\max}$ 表示问题的目标函数为最小化最大拖期。问题的具体假设条件如下：

1）工件之间的加工顺序无约束；

2）所有工件在零时刻均已就绪；

3）工件的加工不需要准备时间；

4）工件加工过程中不允许中断；

5）设备同一时刻最多只能加工 1 个工件；

6）设备周期性预防维护后修复如新；

7）工件加工过程中不会发生随机故障。

1.2 周期性维护下单机调度问题的数学模型

数学规划方法是求解机器调度问题的常用方法之一。近年来，随着计算机技术的发展，出现了诸如CPLEX、GLPK 等高性能的数学规划问题求解器，可以快速求解线性规划、混合整数规划、二次规划等一系列规划问题。针对问题$1\left|\text{npmpt}-\text{pm}\right|T_{\max}$，本节将建立其整数规划模型。

决策变量表示为

$$x_{ij}=\begin{cases}1, & \text{工件}j\text{在第}i\text{个位置加工},\\ 0, & \text{其他};\end{cases} \tag{1}$$

$$y_{[i]}=\begin{cases}1, & \text{加工第}i\text{个位置的工件后进行维护},\\ 0, & \text{其他}。\end{cases} \tag{2}$$

辅助决策变量表示如下：B_i为第i个加工批次；$[i]$为第i个加工位置；$i[j]$为批次B_i中的第j个加工位置；$J_{[i]}$为第i个位置加工的工件；$p_{[i]}$为$J_{[i]}$的加工时间；$d_{[i]}$为工件i的交货时间；$c_{[i]}$为$J_{[i]}$的完工时间；$L_{[i]}$为$J_{[i]}$的延迟时间；$T_{[i]}$为$J_{[i]}$的拖期时间；$e_{[i]}$为$J_{[i]}$的完工时间与上一次维护完成时间的差；$g_{[i]}$为工件$J_{[i]}$所在批次的松弛时间；$b_{[i]}$为工件$J_{[i]}$所在批次的总加工时间。它们存在以下关系：

$$p_{[i]}=\sum_{j=1}^{n}p_j x_{ij}\ , \tag{3}$$

$$d_{[i]}=\sum_{j=1}^{n}d_j x_{ij}\ , \tag{4}$$

$$e_{[1]}=p_{[1]}\ , \tag{5}$$

$$e_{[i]}+My_{[i-1]}\geqslant e_{[i-1]}+p_{[i]}\ , \tag{6}$$

$$e_{[i]}+M\left(1-y_{[i-1]}\right)\geqslant p_{[i]}\ , \tag{7}$$

$$e_{[i]}\leqslant T\ , \tag{8}$$

$$b_{[i]}\geqslant e_{[i]}-M\left(1-y_{[i]}\right)\ , \tag{9}$$

$$g_{[i]}\geqslant T-b_{[i]}-M\left(1-y_{[i]}\right)\ , \tag{10}$$

$$c_{[i]}=\sum_{k=1}^{i-1}b_{[k]}+\sum_{k=1}^{i-1}g_{[k]}+t\sum_{k=1}^{i-1}y_{[k]}+e_{[i]}\ , \tag{11}$$

$$L_{[i]}=c_{[i]}-d_{[i]}\ , \tag{12}$$

$$T_{[i]}=\max\left\{0,L_{[i]}\right\}\ , \tag{13}$$

$$T_{[i]}\geqslant c_{[i]}-d_{[i]}\ , \tag{14}$$

$$T_{\max}\geqslant T_{[i]}\ , \tag{15}$$

$$e_{[i]},b_{[i]},g_{[i]},T_{[i]}\geqslant 0\ , \tag{16}$$

其中，$i,j=1,2,\cdots,n$；M 为一个足够大的正数；T 为连续两次维护活动之间的时间间隔；t 为完成一次维护活动所需要的时间。

模型具体描述如下：

$$\text{Minimize} \quad T_{\max}, \tag{17}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij}=1, \quad i=1,2,\cdots,n, \tag{18}$$

$$\sum_{i=1}^{n} x_{ij}=1, \quad j=1,2,\cdots,n, \tag{19}$$

$$x_{ij}=\begin{cases}1, \\ 0,\end{cases} \quad i,j=1,2,\cdots,n, \tag{20}$$

$$y_{[i]}=\begin{cases}1, \\ 0,\end{cases} \quad i=1,2,\cdots,n. \tag{21}$$

式（17）指出目标函数为最小化最大拖期，式（18）和式（19）表示工件和加工位置的一一对应关系。

2 基于最小批次松弛时间的 TSHA

对于小规模的问题，可以通过商业整数规划求解器来求解。但是，对于大规模问题，求解器难以在有限时间内获得最优解。启发式算法常用来解决大规模问题，它是面向具体问题的经验、规则启发出来的方法，在可接受的时间和空间内给出一个可行解，但这个解可能不是最优的。

文献[5]和文献[11]也研究了以最小化最大拖期为优化目标的集成调度问题，并提出了相应的启发式算法。但是由于维护周期的固定，它们求得的最优解往往还存在较多的机器空闲时间。这不仅会造成大量维护资源的浪费，也可能使得到的解不够完美。针对这个问题，本文提出了 TSHA，通过对松弛时间的充分利用，获得近似最优解。

2.1 构建批次的调度序列

记一个完整的调度序列为 S，调度子序列为 S_i，第 i 次维护为 M_i，B_i 的子集为 W_i，则 $S=(B_1,M_1,B_2,M_2,\cdots,B_l)$. 令 R_i 为批次 B_i 的松弛时间，CM_i 为第 i 次维护活动 M_i 的完成时间，PW_i 为工件集 W_i 的总加工时间，PB_i 为工件集 B_i 的总加工时间。

性质 1　最优调度序列中，批次 B_i 的松弛时间必须小于其后加工的任一工件的加工时间。即 $R_i<p_j$，对于任一工件 $J_j\in B_k$，$k=i+1,i+2,\cdots$

证明：令调度序列 $S=\left(S_1,M_{i-1},B_i,M_i,S_2,M_{k-1},W_{k1},J_j,W_{k2},M_k,S_3\right)$，且 $R_i\geqslant p_j$. 将工件 J_j 移至批次 B_i 最后一个加工工件之后的位置，获得新的调度序列可表示为 $S'=(S_1,M_{i-1},B_1,J_j,M_i,S_2,M_{k-1},W_{k1},W_{k2},M_k,S_3)$.

对于任一工件 $J_p\in W_{k2}$，S 和 S' 中 J_p 的延迟分别为

$$L_p=\mathrm{CM}_{k-1}+\mathrm{PW}_{k1}+p_j+\cdots+p_p-d_p, \tag{22}$$

$$L'_p=\mathrm{CM}_{k-1}+\mathrm{PW}_{k1}+\cdots+p_p-d_p. \tag{23}$$

因为 $p_j>0$，所以 $L_p'<L_p\leqslant T_{\max}$. 可知， S' 的最大拖期相较于 S 更小。证毕。

定理 最优调度序列中，每一批次 B_i 中的工件按交货期最早优先（earliest due date，EDD）规则排序。

证明：令调度序列 $S=(S_1,M_{l-1},W_{l1},J_i,J_{i+1},W_{l2},M,S_2)$，工件 J_i 的交货期晚于工件 J_{i+1}，即 $d_i>d_{i+1}$. 交换工件 J_i 和 J_{i+1} 的加工顺序后新的调度序列 $S'=(S_1,M_{l-1},W_{l1},J_{i+1},J_i,W_{l2},M,S_2)$.

在 S 中， J_i 和 J_{i+1} 的延迟分别为

$$L_i=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_i-d_i, \tag{24}$$

$$L_{i+1}=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_i+p_{i+1}-d_{i+1}. \tag{25}$$

在 S' 中， J_i 和 J_{i+1} 的延迟分别为

$$L_i'=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_{i+1}+p_i-d_i, \tag{26}$$

$$L_{i+1}'=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_{i+1}-d_{i+1}. \tag{27}$$

因为 $p_{i+1}>0$，$p_i>0$， $d_i>d_{i+1}$，所以 $T_{\max}\geqslant L_{i+1}>L_i'>L_i$， $L_{i+1}>L_{i+1}'$. 因此，调度序列 S' 的最大拖期相较于 S 更小。

也就是说，同批次的相邻工件，交货期早的工件应该优先被加工。不断调整相邻工件的位置，直至不存在逆序，这便为最优的批次工件顺序。证毕。

根据性质 1 和定理，设计了第一阶段的算法，构建批次的调度序列，具体步骤如下。

步骤 1：工件排序。将工件按照 EDD 规则排序，如果工件的交货期相同，再按照最短加工时间优先（shortest processing time，PT）规则排序。

步骤 2：构建批次，生成调度序列，具体步骤如下。

1）初始化。令 $i=1$， $j=1$， $R_i=T$.

2）如果 $R_i-p_{[j]}\geqslant 0$，则将工件 $J_{[j]}$ 加入批次 B_i， $R_i=R_i-p_{[j]}$，转至 5）；否则，转至 3）。

3）在未处理工件中，选择处理时间最长且小于等于 R_i 的工件 J_p，将其加入批次 B_i， $R_i=R_i-p_p$. 重复该步骤，直至不存在满足条件的工件。

4）安排一次维护活动，开始一个新批次。令 $i=i+1$， $R_i=T$，将工件 $J_{[j]}$ 加入批次 B_i， $R_i=R_i-p_{[j]}$.

5）如果 $j=n$，终止；否则，令 $j=j+1$，转至 2）。

2.2 改善批次的调度序列

在 2.1 节获得的初始批次调度序列的基础上，改善批次调度序列的基本思想是：在不增加最大拖期的前提下，基于最小批次松弛时间思想，调整工件的加工顺序，使最大拖期工件前一批次有足够的空闲时间，再将最大拖期工件左移，或将与最大拖期工件位于同一批次且先于其加工的工件左移，从而获得更小的最大拖期。

性质 2 令 $J_p\in B_k$， $J_q\in B_l$， $J_j\in B_l$， $k<l$，工件 J_j 位于工件 J_q 之后。

1）当工件 J_q 为最大拖期工件时，若 $p_p-d_p<p_q-d_q$，且交换工件 J_p 和 J_q 后最大拖期出现的位置不变，则交换工件 J_p 和 J_q 后的调度序列更优。

2）当工件 J_j 为最大拖期工件时，若 $p_p<p_q$，且交换工件 J_p 和 J_q 后最大拖期出现的位置不变，则

交换工件 J_p 和 J_q 后的调度序列更优。

证明：令调度序列 $S=\left(S_1,M_{k-1},W_{k1},J_p,W_{k2},M_k,S_2,M_{l-1},W_{l1},J_q,W_{l2},M_l,S_3\right)$，$J_i\in W_{k2}$，$J_j\in W_{l2}$. 因为维护活动的开始和结束时间固定，所以当最大拖期工件位于 $\{S_1,S_2,S_3,W_{k1},W_{l1}\}$ 时，交换工件 J_p 和 J_q 后最大拖期不变。交换工件 J_p 和 J_q 后的调度序列为 $S'=(S_1,M_{k-1},W_{k1},J_q,W_{k2},M_k,S_2,M_{l-1},W_{l1},J_p,W_{l2},M_l,S_3)$.

在 S 中，工件 J_p、J_i、J_q、J_j 的延迟时间可以表示为

$$L_p=\mathrm{CM}_{k-1}+\mathrm{PW}_{k1}+p_p-d_p, \tag{28}$$

$$L_i=\mathrm{CM}_{k-1}+\mathrm{PW}_{k1}+p_p+\cdots+p_i-d_i, \tag{29}$$

$$L_q=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_q-d_q, \tag{30}$$

$$L_j=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_q+\cdots+p_j-d_j. \tag{31}$$

在 S' 中，工件 J_p、J_i、J_q、J_j 的延迟时间可以表示为

$$L_p'=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_p-d_p, \tag{32}$$

$$L_i'=\mathrm{CM}_{k-1}+\mathrm{PW}_{k1}+p_q+\cdots+p_i-d_i, \tag{33}$$

$$L_q'=\mathrm{CM}_{k-1}+\mathrm{PW}_{k1}+p_q-d_q, \tag{34}$$

$$L_j'=\mathrm{CM}_{l-1}+\mathrm{PW}_{l1}+p_p+\cdots+p_j-d_j. \tag{35}$$

当工件 J_q 为最大拖期工件时，因为 $p_p-d_p<p_q-d_q$，所以 $L_p'<L_q=T_{\max}$，调度序列 S' 较 S 更优。当工件 J_j 为最大拖期工件时，因为 $p_p<p_q$，所以 $L_j'<L_j=T_{\max}$，调度序列 S' 较 S 更优。证毕。

根据定理和性质 2，设计了第二阶段算法，改善第一阶段构建的批次调度序列，具体步骤如下。

步骤 1：最大化最大拖期工件前一批次的空闲时间。在不增大 $T_{\max}$ 的前提下，前移加工时间较长的工件，直至最大拖期工件的前一批次。具体步骤如下。

1）令 $i=1$，$j=1$，最大拖期工件处于批次 B_m.

2）从批次 B_j 和 B_i 中选择两工件，使得交换后 B_i 的松弛时间最小。

3）将两工件交换，并对批次 B_j 和 B_i 按照 EDD 规则重新排序，计算批次 B_j 和 B_i 中工件的最大拖期，记最大值为 $T_{\max}$.

4）若 $T\geqslant T_{\max}$，将调度序列恢复为交换前的状态。

5）如果 $j=m-1$，停止；否则，令 $i=i+1$，$j=j+1$，转至 2）。

步骤 2：在不增大 $T_{\max}$ 的前提下，前移最大拖期工件。具体步骤是将最大拖期工件移至前一批次，再对批次 B_{m-1} 和批次 B_m 分别进行 EDD 排序，产生新的最大拖期为 $T_{\max 1}$.

步骤 3：在不增大 $T_{\max}$ 的前提下，提前最大拖期工件的开始加工时间。具体步骤与步骤 1 类似，将批次 B_m 中位于最大拖期工件之前的工件与批次 B_{m-1} 的工件交换，使得交换后 B_{m-1} 的松弛时间最小，产生新的最大拖期为 $T_{\max 2}$.

步骤 4：比较。如果 $T_{\max 1}\geqslant T_{\max 2}$，则新的最大拖期为 $T_{\max 2}$；否则，新的最大拖期为 $T_{\max 1}$.

2.3 数值实例

根据 2.1 节和 2.2 节的内容，绘制了 TSHA 的流程图，如图 2 所示。

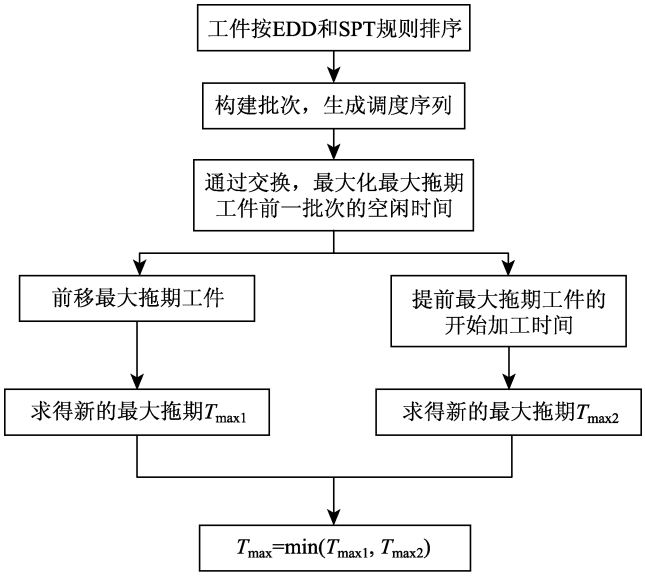

图 2 TSHA 流程图

Fig. 2 Flow chart of TSHA

分别应用 TSHA 算法和 LIAO 等[11]的启发式算法求解文献[11]中的数值实例，工件的处理时间和交货期如表 1 所示，求解结果分别如表 2 和表 3 所示。

表 1 工件的处理时间与交货日期（T=12，t=3）

Tab. 1 Processing time and due date of workpiece (T=12, t=3)

J_i	p_i	d_i
J_1	3	5
J_2	4	10
J_3	4	32
J_4	4	12
J_5	7	32
J_6	2	16
J_7	4	18
J_8	5	36
J_9	4	19
J_{10}	3	20
J_{11}	3	40

表 2 TSHA 求得的调度序列

Tab. 2 Schedule obtained by TSHA

J_i	T_i
J_2	0
J_4	0
J_7	0
M_1	—
J_1	13
J_9	3
J_8	0
M_2	—
J_6	16
J_{10}	15
J_5	10
M_3	—
J_3	**17**
J_{11}	12

注：黑体值为获得的最大拖期 T_{max}

表 3 LIAO 等[11]求得的调度序列

Tab. 3 Schedule obtained by LIAO et al[11]

J_i	T_i
J_1	0
J_2	0
J_6	0
J_{10}	0
M_1	—
J_4	7
J_5	0
M_2	—
J_7	16
J_9	**19**
M_3	—
J_3	17
J_8	18
J_{11}	17

注：黑体值为获得的最大拖期 T_{max}

由表 2 和表 3 可知，TSHA 算法获得的调度序列的最大拖期为 17，最大拖期之前批次的松弛时间为{0,0,0}，而 LIAO 等[11]求得的最大拖期为 19，最大拖期之前批次的松弛时间为{0,1,4}. 由此可知，TSHA 获得的解更优。

3 仿真实验与结果分析

为证明 TSHA 的有效性，利用不同的工件规模和参数设置设计实验，将 TSHA 与 SBIHI 等[5]的启发式算法进行对比分析，再将 TSHA 求得的解与规划求解器 CPLEX 求得的最优解作比较。

实验参数的生成方式与 SBIHI 等[5]相同，设置如下：工件的处理时间为区间[1, 10]上的离散平均随机变量，交货期为区间$[(1-C-Q/2)\sum_{i=1}^{n}p_i,(1-C+Q/2)\sum_{i=1}^{n}p_i]$上的离散平均随机变量，其中，$C$ 为拖期因子，Q 为交货期幅度。为避免算法涉及的主要参数的不同取值对算法的优劣性判断有影响，拖期因子 C、交货期幅度 Q、维护周期 T、维护时间 t 取多组值。交货期幅度 $Q=\{0.2,0.6\}$，拖期因子 $C=\{0.2,0.6\}$，它们共有 4 种组合，分别为 $C=0.2$、$Q=0.2$，$C=0.2$、$Q=0.6$，$C=0.6$、$Q=0.2$，$C=0.6$、$Q=0.6$. 维护周期和维护时间考虑两种情况，$T=10$、$t=2$ 和 $T=18$、$t=4$. 对于每一种问题参数 C、Q、n、T、t 的组合，随机产生 50 个问题实例。

实验中，算法均使用 Matlab R2016a 编写，在配置为 2.71 GHz 处理器、4 GB RAM 的 DELL-Vostro 上运行。为简便起见，下文将 SBIHI 等[5]的启发式算法简记为算法 H1.

3.1 与 H1 的比较实验

分别在工件数为{10, 20, 30, 40, 50, 80, 110, 150, 170, 200}的 10 种情况下，将 TSHA 和 H1 进行实验对比分析，解的优劣性和计算时间如表 4 和表 5 所示。表中 TSHA 列下的“No”表示 TSHA 的解比 H1

表 4 TSHA 与 H1 的解的优劣性比较（*C*=0.2，*Q*=0.2）

Tab. 4 Comparison between TSHA and H1 in quality of solutions (*C*=0.2, *Q*=0.2)

T	*t*	*n*	TSHA			H1		
			No	Max	Mean	No	Max	Mean
10	2	10	25	16	5.00	6	13	3.33
		20	39	23	9.48	3	5	3.00
		30	44	36	12.65	4	12	6.25
		40	42	46	19.95	1	2	2.00
		50	49	52	26.38	1	4	4.00
		80	50	72	46.02	0	0	0
		110	50	120	65.80	0	0	0
		150	50	133	92.40	0	0	0
		170	50	156	108.00	0	0	0
		200	50	201	134.96	0	0	0
18	4	10	25	20	5.04	10	14	7.50
		20	36	30	9.97	3	8	4.00
		30	44	44	15.38	2	6	3.50
		40	48	39	19.68	2	13	8.00
		50	46	47	23.23	1	4	4.00
		80	50	80	42.20	0	0	0
		110	50	109	64.62	0	0	0
		150	50	135	86.40	0	0	0
		170	50	152	98.52	0	0	0
		200	50	163	126.74	0	0	0

注：实验发现，解的优劣性与拖期因子 *C* 及交货期幅度 *Q* 的取值关系不大，为展示的简洁性，仅列出(*C*, *Q*)=(0.2, 0.2)的实验结果

的解更优的实例数目，“Max”和“Mean”分别表示 TSHA 比 H1 更优解中两解的最大差值和平均差值。同样地，表 4 中 H1 列下的“No”、“Max”和“Mean”分别表示算法 H1 的解比 TSHA 的解更优的实例数目、两解的最大差值及平均差值。

由表 5 可以看出，比较两个算法获得的优解数目、解的最大差值和平均差值三个方面，TSHA 较 H1 均更优，且这种优异性随着问题规模的增大更加突显。当工件数 n 超过 50 时，TSHA 获得更优解的比例接近 100%. 这是因为 TSHA 充分利用了每个批次的松弛时间，使工件的加工更为紧凑，接近于无维护活动干扰情况下的单机调度。

由表 5 可以看出，虽然 TSHA 比 H1 的计算时间稍长一些，但两者的差距最多不超过 0.06 s，而且在问题规模为 n=200 的情况下，TSHA 花费的时间仅为 0.079 9 s. 因此，H1 的求解速度虽然比 TSHA 更快，但差距并不明显。

表 5 TSHA 与 H1 的计算时间比较（C=0.2，Q=0.2）（s）

Tab. 5 Comparison between TSHA and H1 in computation time (C=0.2, Q=0.2) (s)

n	TSHA	H1
10	0.028 1	0.015 9
20	0.032 0	0.019 7
30	0.045 0	0.020 2
40	0.053 5	0.020 7
50	0.055 6	0.021 5
80	0.055 9	0.022 0
110	0.065 7	0.023 1
150	0.071 3	0.023 5
170	0.078 9	0.024 1
200	0.079 9	0.025 8

总的来说，从实验结果可以看出，相较于 SBIHI 等[5]提出的启发式算法 H1，本文提出的启发式算法 TSHA 性能更优。

3.2 与 CPLEX 的比较实验

分别在工件数为{10,15, 20}的情况下，将算法 TSHA 和 CPLEX 得到的最优解进行比较，实验结果如表 6 所示。表 6 中“No”表示 TSHA 求得最优解的实例数目，“Max”和“Mean”分别表示 TSHA 和 CPLEX 得到的最优解的最大偏差和平均偏差。

$$\text{Relative Error}=\frac{T_{\max}(\text{H1})-\text{opt}\_T_{\max}}{\text{opt}\_T_{\max}}. \tag{36}$$

表 6 TSHA 与 CPLEX 的性能比较

Tab. 6 Comparison between TSHA and CPLEX in performance

(C, Q)	(T, t)	n=10			n=15			n=20		
		No	Max	Mean	No	Max	Mean	No	Max	Mean
(0.2, 0.2)	(10, 3)	43	0.636 4	0.235 0	43	0.352 9	0.236 7	46	0.114 8	0.065 9
	(10, 6)	42	0.565 2	0.233 5	43	0.258 1	0.121 3	44	0.154 6	0.064 2
	(15, 3)	41	0.857 1	0.341 3	42	0.600 0	0.405 8	44	0.360 0	0.169 0
	(15, 6)	41	0.800 0	0.267 2	42	0.488 4	0.210 1	44	0.300 0	0.105 6
	(20, 3)	42	0.687 5	0.203 3	45	0.523 8	0.189 1	46	0.161 3	0.084 7
	(20, 6)	42	0.888 9	0.379 7	44	0.348 3	0.188 3	46	0.107 1	0.068 2
(0.6, 0.6)	(10, 3)	43	0.461 5	0.199 3	45	0.265 3	0.225 3	47	0.166 7	0.100 6
	(10, 6)	45	0.295 5	0.146 5	46	0.205 1	0.142 8	46	0.118 5	0.048 8
	(15, 3)	42	0.473 7	0.254 2	42	0.272 7	0.175 4	44	0.266 7	0.143 9
	(15, 6)	42	0.535 7	0.171 4	43	0.333 3	0.163 8	44	0.200 0	0.106 6
	(20, 3)	42	0.346 2	0.154 1	43	0.208 3	0.099 6	46	0.122 8	0.051 1
	(20, 6)	42	0.363 6	0.141 7	43	0.275 4	0.130 5	45	0.147 5	0.056 4
时间/s	TSHA	<0.1			<0.1			<0.1		
	CPLEX	0.38			48.54			239.80		

由表 6 可以看出，随着工件数目的增大，CPLEX 所需要的时间呈指数级增长，而 TSHA 的计算时间

始终小于 0.1 s. 在解的优劣性方面，平均情况下，TSHA 获得最优解的数目为 44 个（总共 50 个），与最优解的平均偏差不超过 20%. 并且随着问题规模的增大，TSHA 的解愈加趋向于最优解，当工件数为 20 时，与最优解的平均偏差低至 5%. 此外，由表 6 可知，随着维护周期 T 的增大和维护时间 t 的减小，平均偏差逐渐减少，这是因为此种情况下维护活动次数减少。

4 结论

本文对以最小化最大拖期为优化目标的周期性维护和单机调度的集成问题进行了深入研究，建立了优化模型，并针对以往研究提出的启发式算法中由于对松弛时间的利用不够充分而导致求得的解不够完美的问题，设计了一种基于最小化松弛时间的两阶段启发式算法。数据实验结果表明，本文提出的算法可以在有限时间内获得近似最优解，而且较以往的算法获得的解更加优异。本文仅研究了单机系统下的集成优化模型，对于多台设备或者混联系统下的集成优化问题等，还需进一步的研究。

[参考文献] (References)

[1] MANE J K, GHADLE K P. Maintenance activity single-machines scheduling and due-date assignment simultaneously[J]. International Journal of Mathematical Engineering and Science, 2014, 4(1): 37-47.

[2] WAN L. Scheduling jobs and a variable maintenance on a single machine with common due-date assignment[J]. The Scientific World Journal, 2014, 2014(2): 748905.

[3] CHEN W J. Minimizing number of tardy jobs on a single machine subject to periodic maintenance[J]. Omega, 2009, 37(3): 591-599.

[4] LEE J Y, KIM Y D. Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance[J]. Computers & Operations Research, 2012, 39(39): 2196-2205.

[5] SBIHI M, VARNIER C. Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness[J]. Computers & Industrial Engineering, 2008, 55(4): 830-840.

[6] LIU B Y, CHEN W D. Single-machine scheduling with preventive periodic maintenance and resumable jobs in remanufacturing system[J]. Journal of Southeast University (English Edition), 2012, 28(3): 349-353. (in Chinese)

[7] 张思源，陆志强，崔维伟. 考虑设备周期性维护的流水车间生产调度优化算法[J]. 计算机集成制造系统，2014，20（6）：1379-1387.
ZHANG S Y, LU Z Q, CUI W W. Flow shop scheduling optimization algorithm with periodical maintenance[J]. Computer Integrated Manufacturing Systems, 2014, 20(6): 1379-1387. (in Chinese)

[8] LOW C, JI M, HSU C J, et al. Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance[J]. Applied Mathematical Modelling, 2010, 34(2): 334-342.

[9] 蒋志高. 考虑多阶段维护且加工时间可变的车间作业调度问题研究[D]. 上海：上海交通大学，2011.
JIANG Z G. Study on job shop scheduling problem with multi-phase maintenance and variable processing time[D]. Shanghai: Shanghai Jiao Tong University, 2011. (in Chinese)

[10] GRAHAM R L, LAWLER E L, LENSTRA J K, et al. Optimization and approximation in deterministic sequencing and scheduling: a survey[J]. Annals of Discrete Mathematics, 1979, 5(1): 287-326.

[11] LIAO C J, CHEN W J. Single-machine scheduling with periodic maintenance and nonresumable jobs[J]. Computers & Operations Research, 2003, 30(9): 1335-1347.