

# 综合考虑服务水平和资源占用的平行机调度问题\*

冯鑫<sup>1</sup>, 陈旒珊<sup>2</sup>, 秦岚<sup>3</sup>

(1. 南京林业大学 经济管理学院, 南京 210037; 2. 华东交通大学 人文社会科学学院, 南昌 330013;

3. 浙江万里学院 物流与电子商务学院, 浙江 宁波 315100)

**摘要:**【目的】对综合考虑服务水平和资源占用的平行机调度问题进行研究,建立以最小化总延误和机器使用数量之和为优化目标的优化模型。【方法】针对该问题的特点,设计了一种改进的模拟退火算法进行求解。采用不同规模的数值实验测试算法的性能,并对问题参数展开灵敏度分析。【结果】改进的模拟退火算法具有良好的求解效率和求解质量。【结论】灵敏度分析表明,增加资源储备可以使企业更好地平衡服务水平和资源占用。模型能够对企业实现资源节约型制造提供指导和决策支持。

**关键词:** 平行机调度; 生产排序; 资源占用; 模拟退火算法

**中图分类号:** O223; F252

**文献标志码:** A

**文章编号:** 1672-6693(2023)02-0001-07

制造业作为中国国民经济的主体,自然成为实现节能降耗的主要关注对象。目前关于资源节约型制造业的研究多数集中于战略层面。为了形成一个从宏观政策层面到具体操作层面全覆盖的资源节约型制造业的构建体系,需要进一步丰富微观运营层面的研究<sup>[1]</sup>。任务调度是制造业微观运营层面的关键决策环节。制造环节中任务的合理调度能够提高系统工作效率,实现资源配置的优化<sup>[2]</sup>。本文以制造系统中的平行机调度问题为研究对象,设计综合考虑服务水平和资源占用的优化方案,以实现兼顾经济发展和资源节约的制造业发展模式。

平行机调度问题广泛存在于汽车装配、化工生产、电子产品制造等领域,因而受到了理论界的深度关注。多数平行机调度研究关注的优化目标仅侧重于服务水平,如任务的最大完工时间(makespan)<sup>[3]</sup>、总完工时间<sup>[4]</sup>、最大延误时间<sup>[5]</sup>、总延误时间<sup>[6-7]</sup>等。随着对资源节约型制造的重视,理论界逐渐开始关注资源占用的优化目标。在面向资源节约型的任务调度优化问题中,现有文献多用电力成本<sup>[8-10]</sup>、机器功耗<sup>[11-13]</sup>或机器使用数量<sup>[14-16]</sup>来刻画制造过程中的资源占用。本文采用机器使用数量刻画资源占用。在以机器使用数量刻画资源占用的平行机调度研究中,Chen等人<sup>[17]</sup>证明了最小化机器数的平行机调度问题是NP-难问题。因此,学者主要探讨了特殊情形下(所有任务的处理时间均为单位长度)的调度模型<sup>[18-19]</sup>。Wang等人<sup>[20]</sup>突破了单位长度处理时间的限制,对一般性问题设计了一种禁忌搜索启发式算法,以最小化机器数量和机器负载成本之和。Osorio-Valenzuela等人<sup>[21]</sup>研究了带有时间窗的平行机最小化使用机器数问题。

现有文献大多侧重于服务水平或资源占用一个优化目标,而同时考虑服务水平和资源占用这个优化目标对平行机调度问题的共同影响,并设计针对性的调度策略的相关研究较少。因此,本文分别用总延误时间和机器使用数量刻画服务水平和资源占用,建立以最小化总延误时间和机器使用数量的加权和为优化目标的混合整数规划模型,并设计基于模拟退火的启发式算法进行求解,为中国实现资源节约型制造业转型提供更为全面的决策参考。

## 1 问题描述

综合考虑生产效率和资源占用的平行机调度可以描述为:已知所有待处理任务  $J = \{1, 2, \dots, n\}$  在 0 时刻释放。其中:任务  $i$  的处理时间为  $p_i$ , 交货期为  $d_i$ 。制造车间有  $m$  台完全相同的平行机用于处理任务。一台机器在同一时刻只能处理 1 个任务。同样,1 个任务只能被 1 台机器处理,且处理过程不能中断。记任务  $i$  的完工时间为  $C_i$ 。则该任务的延迟时间为  $T_i = \max\{C_i - d_i, 0\}$ 。

为了提高制造系统的生产效率,决策者希望尽可能降低所有任务的总延迟时间  $\sum_{i=1}^n T_i$ 。使用更多的机器用

\* 收稿日期:2021-12-09 修回日期:2022-09-19 网络出版时间:2023-04-20T16:29

资助项目:国家自然科学基金青年项目(No. 71701048)

第一作者简介:冯鑫,男,教授,博士,研究方向为供应链调度,E-mail:fengxin90@njfu.edu.cn

网络出版地址:https://kns.cnki.net/kcms/detail/50.1165.N.20230420.1421.020.html

于处理任务是提升生产效率、降低延误的有效方法。在合理的调度方案下,使用的机器数越多,总延误时间越小。然而,机器的使用会产生对应的资源占用成本,包括设备维修、日常维护、人员配备等。因此,通过增加机器数降低总延误时间是以资源占用成本为代价的。反之,如果过度控制机器使用数以减少资源占用,必然会影响到制造系统的生产效率。因此,本文关注的优化目标是总延误时间  $\sum_{i=1}^n T_i$  和机器使用数量  $M$  之间的权衡,即  $\alpha \sum_{i=1}^n T_i + (1-\alpha)M$ 。其中: $\alpha$  表示决策者关于总延误时间和机器使用数量的偏好程度,即决策偏好参数,取值范围为  $0 \leq \alpha \leq 1$ 。

## 2 数学模型

### 2.1 参数说明

本文将用到的符号及相关含义见表 1。

表 1 相关符号说明  
Tab. 1 Symbol description

模型参数	含义	决策变量	含义
$n$	待处理的任務总数	$C_{jk}$	第 $j$ 台机器第 $k$ 个任务的處理时间
$m$	可用机器数	$T_{jk}$	第 $j$ 台机器第 $k$ 个任务的延迟时间
$i$	任务编号	$x_{ijk}$	如果任务 $i$ 是机器 $j$ 的第 $k$ 个任务,则 $x_{ijk} = 1$ ; 否则, $x_{ijk} = 0$
$j$	机器编号	$y_j$	如果机器 $j$ 用于任务处理,则 $y_j = 1$ ; 否则, $y_j = 0$
$k$	机器上的任务位置编号		
$p_i$	任务 $i$ 的處理时间		
$d_i$	任务 $i$ 的交货期		
$\alpha$	决策偏好参数		

### 2.2 优化模型

对上述问题建立混合整数规划模型如下:

$$\min \alpha \sum_{j=1}^m \sum_{k=1}^n T_{jk} + (1-\alpha) \sum_{j=1}^m y_j, \quad (1)$$

$$\text{s. t. } \sum_{j=1}^m \sum_{k=1}^n x_{ijk} = 1, \forall i = 1, \dots, n, \quad (2)$$

$$\sum_{i=1}^n x_{ijk} \leq 1, \forall j = 1, \dots, m, k = 1, \dots, n, \quad (3)$$

$$x_{ijk} \leq x_{ij,k-1}, \forall i = 1, \dots, n, j = 1, \dots, m, k = 2, \dots, n, \quad (4)$$

$$C_{jk} \geq C_{j,k-1} + \sum_{i=1}^n x_{ijk} p_i, \forall j = 1, \dots, m, k = 1, \dots, n, \quad (5)$$

$$d_{jk} \geq \sum_{i=1}^n x_{ijk} d_i, \forall j = 1, \dots, m, k = 1, \dots, n, \quad (6)$$

$$T_{jk} \geq C_{jk} - d_{jk}, \forall j = 1, \dots, m, k = 1, \dots, n, \quad (7)$$

$$y_j = \sum_{i=1}^n x_{ij1}, \forall j = 1, \dots, m, \quad (8)$$

$$x_{ijk}, y_j \in \{0, 1\}, \forall i = 1, \dots, n, j = 1, \dots, m, k = 1, \dots, n, \quad (9)$$

$$C_{jk}, T_{jk} \geq 0, \forall i = 1, \dots, n, j = 1, \dots, m, k = 1, \dots, n. \quad (10)$$

式中:式(1)为最小化综合考虑总延迟和机器使用数的目标函数;约束条件(2)保证了每个任务只能在 1 台机器的 1 个位置上处理;约束(3)保证每台机器的每个位置最多处理 1 个任务;约束(4)保证机器上每个处理任务的位置保持连续;约束(5)、(6)和(7)分别计算每台机器对应位置上的完工时间和延误时间;约束(8)确定机器是否用于任务处理;约束(9)和(10)为决策变量的取值范围。

注意到  $m=1$  时,该问题退化为经典的单机最小化总延误排序问题。Du 等人<sup>[22]</sup>证明了单机最小化总延误排序问题是 NP-难问题。因此,本文的问题也是 NP-难的。

### 3 算法设计

考虑到问题的 NP-难特性使得大规模问题的求解难以在较短时间内完成,故本文设计了一种基于模拟退火(SA)方法的启发式算法来实现优化模型的目标。SA 算法是由 Kirkpatrick 等人<sup>[23]</sup>首先设计开发的。通过模拟热力学当中的退火过程,首先生成一个初始解作为当前解  $S$ ,并按照一定方式搜索相邻解  $S'$ ,再根据 Kirkpatrick 等人<sup>[23]</sup>设计的接受概率准则判断相邻解是否被接受为当前解,不断重复上述过程,直到满足终止准则,得到问题的解。接受概率准则为:

$$P_{\text{accept}} = \begin{cases} 1, & F(S') \leq F(S), \\ \exp(-(F(S') - F(S))/T), & F(S') > F(S). \end{cases} \quad (11)$$

式中:参数  $T=300/w$  为计算温度, $w$  表示算法的迭代次数。 $F(S)$  和  $F(S')$  分别代表当前解  $S$  和相邻解  $S'$  对应的目标函数值。

当接收概率  $P_{\text{accept}} > \text{rand}$  时,相邻解被接受为新的当前解,其中  $\text{rand} \in [0,1]$  为随机数。在算法计算的早期阶段,由于迭代次数较少,计算温度  $T$  较高,大多数相邻解都能被接受,从而能较好地跳出局部最优解。随着算法的进行,计算温度  $T$  不断降温,相邻解被接受的标准不断提高,从而保证算法的尽快收敛。

因为在求得最优解之前无法确定对应的使用机器数,本文设计的 SA 算法包含  $m$  个子算法,其中子算法  $j$  计算当机器使用数为  $j$  时对应的最优任务排序。SA 算法基本框架见图 1。

#### 3.1 初始解的生成

考虑到当机器数固定时,优化目标为最小化总延迟。因此在子算法  $j$  中,为了尽量最小化总延迟,在  $j$  台机器上分别使用最早交货期优先规则(earliest due date, EDD)方法和最小处理时间优先规则(shortest processing time, SPT)方法依次将任务分配到空闲机器上作为得到对应的解  $S_{\text{EDD}}(j)$  和  $S_{\text{SPT}}(j)$ 。子算法  $j$  的初始解为  $S(j) = \arg \min\{F(S_{\text{EDD}}(j)), F(S_{\text{SPT}}(j))\}$ 。

#### 3.2 更新规则

本文设计了基于任务互换的更新规则,即通过交换任意两个任务的处理位置来获得新的相邻解。由于问题中包含  $n$  个工件,那么每个解都存在  $C_n^2 = n(n-1)/2$  种可能的交换方案。为了提高 SA 算法求解效率,提出如下引理以减少交换方案的搜索域。

**引理 1** 如果存在两个任务  $i_1$  和  $i_2$  满足  $p_{i_1} \leq p_{i_2}$  且  $d_{i_1} - p_{i_1} \leq d_{i_2} - p_{i_2}$ ,那么在问题的最优解中,如果  $i_1$  和  $i_2$  被安排在同一台机器上连续处理,那么  $i_1$  一定先于  $i_2$  处理。

**证明** 假设在问题的最优解  $S^*$  中,存在满足上述条件的工件  $i_1$  和  $i_2$ ,且  $i_1$  在  $i_2$  之后处理,见图 2a。任务  $i_1$  和  $i_2$  的延误分别为  $T_{i_1} = \max\{t + p_{i_1} + p_{i_2} - d_{i_1}, 0\}$  和  $T_{i_2} = \max\{t + p_{i_2} - d_{i_2}, 0\}$ 。

通过交换  $i_1$  和  $i_2$  的顺序,构造一个新的排序  $S'$ ,如图 2b 所示。在新的排序中,除  $i_1$  和  $i_2$  之外的任务的开始处理时间和完工时间均保持不变,因而延误时间不变。任务  $i_1$  和  $i_2$  的延误时间分别变为  $T'_{i_1} = \max\{t + p_{i_1} - d_{i_1}, 0\}$  和  $T'_{i_2} = \max\{t + p_{i_1} + p_{i_2} - d_{i_2}, 0\}$ 。分如下情况对比排序  $S^*$  和  $S'$ 。

情形 1,当  $t + p_{i_2} - d_{i_2} \geq 0$  时,由  $d_{i_1} - p_{i_1} \leq d_{i_2} - p_{i_2}$  可知,  $t + p_{i_1} - d_{i_1} \geq 0$ 。此时,  $F(S') - F(S^*) = T'_{i_1} + T'_{i_2} -$

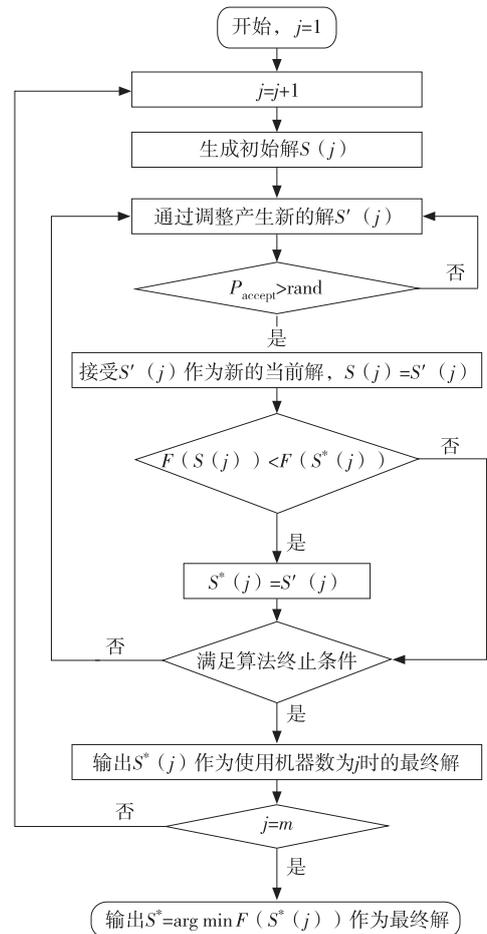


图 1 算法流程

Fig. 1 Algorithm flow chart

$$(T_{i_1} + T_{i_2}) = p_{i_1} - p_{i_2} \leq 0.$$

情形 2, 当  $t + p_{i_2} - d_{i_2} < 0$  时, 进一步分类讨论如下:

1) 如果  $t + p_{i_1} - d_{i_1} \geq 0$ . 此时,  $F(S') - F(S^*) = T'_{i_1} + T'_{i_2} - (T_{i_1} + T_{i_2}) = \max\{t + p_{i_1} + p_{i_2} - d_{i_2}, 0\} - p_{i_2}$ . 因为  $t + p_{i_2} - d_{i_2} < 0$ ,  $\max\{t + p_{i_1} + p_{i_2} - d_{i_2}, 0\} \leq p_{i_1}$ , 那么  $F(S') - F(S^*) = T'_{i_1} + T'_{i_2} - (T_{i_1} + T_{i_2}) \leq p_{i_1} - p_{i_2} \leq 0$ .

2) 如果  $t + p_{i_1} - d_{i_1} < 0$ . 此时, 有:

$$F(S') - F(S^*) = T'_{i_1} + T'_{i_2} - (T_{i_1} + T_{i_2}) = \max\{t + p_{i_1} + p_{i_2} - d_{i_2}, 0\} - \max\{t + p_{i_1} + p_{i_2} - d_{i_1}, 0\},$$

因为  $p_{i_1} \leq p_{i_2}$  且  $d_{i_1} - p_{i_1} \leq d_{i_2} - p_{i_2}$ ,  $t + p_{i_1} + p_{i_2} - d_{i_2} \leq t + p_{i_1} + p_{i_2} - d_{i_1}$ , 则有  $F(S') - F(S^*) = T'_{i_1} + T'_{i_2} - (T_{i_1} + T_{i_2}) \leq 0$ .

综上所述, 通过交换  $i_1$  和  $i_2$  的顺序得到新的排序  $S'$  对应的总延误小于原排序, 和原排序  $S^*$  是最优排序矛盾. 证毕

根据引理 1, 如果当前解的排序方案确定时, 可以判断每台使用机器上的工件之间合理的优先序, 即如果两个连续处理的任务  $i_1$  和  $i_2$  满足  $p_{i_1} \leq p_{i_2}$  且  $d_{i_1} - p_{i_1} \leq d_{i_2} - p_{i_2}$ , 那么  $i_1$  优先于  $i_2$ , 记为  $i_1 > i_2$ . 这种优先序具有传递性, 即如果  $i_1 > i_2$  且  $i_2 > i_3$ , 那么  $i_1 > i_3$ . 这样, 在通过交换工件处理顺序得到新的相邻解时, 排除  $i_1$  和  $i_2$ ,  $i_1$  和  $i_3$  之间的互换, 从而加快算法求解效率.

### 3.3 算法流程

SA 算法具体步骤如下.

步 0,  $j = 1$ ;

步 1, 根据 3.1 节方法生成初始排序  $S(j)$ ,  $S^*(j) = \emptyset$ ,  $F(S^*(j)) = +\infty$ ,  $N = 0$ ,  $i = 0$ ;

步 2,  $i = i + 1$ ;

    如果  $i = n$ , 那么  $i = 1$ ;

$h = i$ ;

步 3,  $h = h + 1$ ,  $N = N + 1$ . 如果  $h > n$ , 那么令  $i = 0$ , 返回步 2;

    如果  $i$  和  $h$  在一台机器上处理, 且  $i > h$ , 互换后  $i$  晚于  $h$  处理, 那么重复步 3;

    否则, 交换  $i$  和  $h$  的位置, 得到新的排序  $S'(j)$ , 进入步 4;

    如果  $N > n(n-1)/2$ , 转到步 5;

步 4, 如果  $P_{\text{accept}} > \xi$ , 那么

$S(j) = S'(j)$ ,  $N = 1$ ;

        如果  $F(S(j)) < F(S^*(j))$ ,

        那么  $S^*(j) = S(j)$ ;

    返回步 3;

步 5,  $j = j + 1$ ;

    如果  $j > m$ , 算法结束, 输出最终解  $S^* = \arg \min_j \{F(S(j))\}$ ;

    否则, 返回步 1;

## 4 算例实验

本文通过随机生成的算例测试 SA 算法的计算性能, 并和商业优化求解软件 ILOG CPLEX 12.7.1 以及 Biskup 等人<sup>[7]</sup>设计的 BHG-MDD 算法进行对比, 验证 SA 算法的优越性. 其中 BHG-MDD 算法是用于有效求解不考虑资源占用的平行及调度问题, 优化目标为最小化总延迟. 针对所研究的综合考虑服务水平和资源占用的平行机调度问题, 本文设置 BHG-MDD 算法依次对机器使用数为 1, 2,  $\dots$ ,  $m$  时平行机调度问题求解, 并从中选取最优解作为 BHG-MDD 算法的最终解.

此外, 本文还对问题相关参数展开了灵敏度分析, 以挖掘问题参数变化对综合考虑服务水平和资源占用的平行机调度绩效的影响作用.

### 4.1 算例参数设置

本文的算例实验包含  $m = 5$  台可用机器和  $n = 10, 20, 30, 40$  个任务的小规模算例用于算法性能测试, 以及

$m=5, 10, 20$  台可用机器和  $n=50, 100, 150, 200$  个任务的大规模算例用于参数灵敏度分析。任务  $i$  的处理时间  $p_i$  在  $[1, 10]$  的区间内随机产生, 任务  $i$  的交货期  $d_i = p_i + \Delta_i$ ,  $\Delta_i$  在  $[1, 20]$  的区间内随机生成。为了降低随机性, 本文对每组实验都重复 10 次, 共产生了 40 组小规模算例和 120 组大规模算例。

#### 4.2 算法性能测试

SA 算法的性能主要从求解效率和求解质量两个标准展开评价。40 组小规模算例的实验结果见表 2。其中表 2 第 1 列记录任务数量  $n$ ; 第 2~3 列、第 4~5 列和第 7~8 列分别记录 CPLEX、SA 算法和 BHG-MDD 算法输出解对应的计算时间和目标函数值; 第 6 列为 SA 算法和最优解之间的偏差 ( $G_1$ )。第 9 列为 BHG-MDD 算法和最优解之间的偏差 ( $G_2$ )。

表 2 算法性能测试

Tab. 2 The computational performance testing

$n$	CPLEX		SA 算法			BHG-MDD 算法		
	计算时间	目标函数值	计算时间	目标函数值	$G_1/\%$	计算时间	目标函数值	$G_2/\%$
10	9.21	2.78	11.23	2.79	0.21	6.64	2.80	0.59
20	129.25	12.37	22.78	12.53	1.29	13.57	12.67	2.44
30	542.68	20.98	33.36	21.37	3.47	23.36	22.62	7.82
40	1 891.08	55.21	47.21	56.72	5.83	34.21	62.13	12.53
平均值	643.06	22.84	28.65	23.35	2.70	19.45	23.35	5.85

在求解质量方面, 观察表 2 第 6 列可知, SA 算法输出的解和最优解之间的差距不超过 5.83%。对所有  $n \leq 40$  的算例, SA 算法输出的解和最优解之间的平均偏差为 2.70%。这表明 SA 算法输出的解非常接近最优解, 具有较好的求解质量。作为对比, BHG-MDD 算法和最优解之间的最大和平均偏差分别为 12.53% 和 5.85%。这说明 SA 算法相较于 BHG-MDD 算法, 输出的解更接近最优, 具有明显的求解质量优越性。

在求解效率方面, 观察表 2 第 2 列可知, CPLEX 的求解时间随着问题规模的增大而迅速上升。当  $n=40$  时, CPLEX 计算时间激增至 1 891.08 s。当  $n=50$  时, CPLEX 无法在 2 h 内输出最优解。作为对比, SA 算法的计算时间随着问题规模的增大而平稳增加。所有问题均可在 50 s 内输出最终结果, 平均计算时间仅为 28.65 s。这表明本文设计的 SA 算法具有很好的计算效率, 在应对较大规模算例时明显优于商业优化软件。同时, 和 BHG-MDD 算法的平均计算时间 19.45 s 相比, SA 算法的平均求解时间相差在 10 s 以内, 近似接近 BHG-MDD 算法。

综合考虑求解质量和求解效率, SA 算法相比 CPLEX 而言, 能够较好地解决较大规模的问题。同时, SA 算法和 BHG-MDD 算法相比, 能够通过牺牲微小求解效率来实现求解质量的大幅提升, 因而具有算法优越性。

#### 4.3 灵敏度分析

为了分析制造系统的资源储备对本文所研究问题的影响作用, 本文通过灵敏度分析挖掘可用机器数  $m$  对问题的影响作用。因为当  $n \geq 50$  时, CPLEX 不能在 2 h 内输出最优解, 因此采用 SA 算法完成灵敏度分析。

图 2a 和图 2b 分别记录了当可用机器数  $m$  和待处理任务数  $n$  变化时, 问题最终解的计算时间和目标函数值的变化。图 2a 表明, SA 算法的计算时间随着可用机器数  $m$  的增加而增加。因此, 制造企业的资源储备数量增加了问题的求解难度。观察图 2b 可知, 随着可用机器数量  $m$  的增加, 总延误和机器使用成本的目标函数值逐渐降低, 这表明储备更多资源对制造企业提高服务水平和降低资源使用成本具有重要作用。

综上所述, 灵敏度分析结果表明, 增加资源储备是一种可以有效提升服务水平的同时实现向资源节约型制造业转型的有效途径。

## 5 结束语

本文研究了综合考虑服务水平和资源占用的平行机调度优化问题。用总延误时间刻画服务水平, 机器使用数刻画资源占用, 构建了对应的整数规划模型。根据模型特点和问题的性质, 设计了具有针对性的 SA 算法用于模型求解。通过仿真实验对 SA 算法性能进行验证, 结果表明 SA 算法相较 CPLEX, 不仅具有更好的求解效率, 求得的解也能够很好地贴近最优解。同时, 与现有的智能算法相比, SA 算法同样具有优越性。灵敏度分析表

明,增加资源储备能够帮助制造企业在提升服务水平同时,实现向资源节约型制造业转型,降低企业的转型运营成本。因此,在中国制造业为实现“碳达峰、碳中和”的转型目标时,保证资源储备和供给能力是重要的支撑手段。

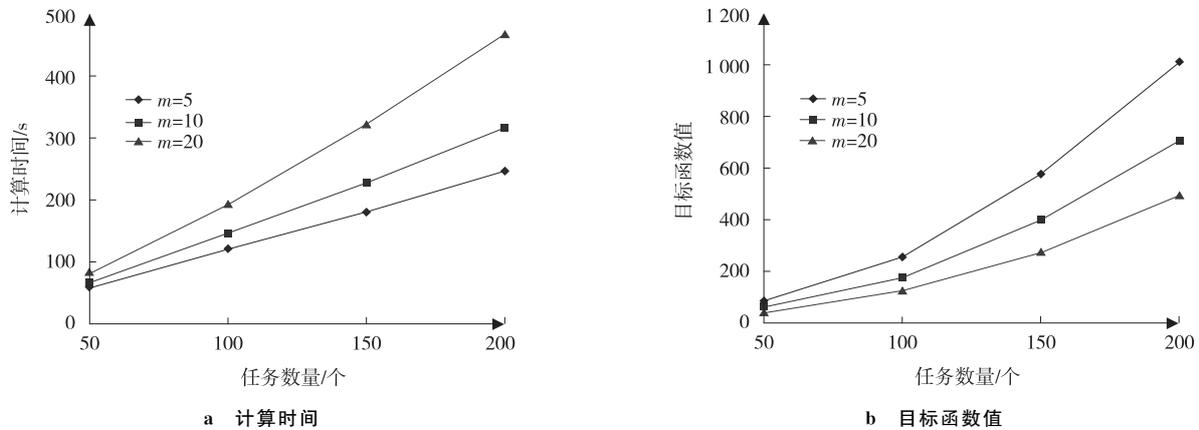


图 2 可用机器数  $m$  对问题的影响作用

Fig. 2 Impact of the number of available machines  $m$

考虑到模型的复杂性,当  $n \geq 50$  时,CPLEX 不能在 2 h 内输出最优解,无法在更大规模的算例下进一步评价 SA 算法的计算性能。因此需要根据模型的结构挖掘问题的性质和寻找有效不等式,从而设计更为高效的精确算法来获得最优解。此外,在不确定环境下综合考虑服务水平和资源占用的平行机调度随机优化问题也是下一步研究的重点。

#### 参考文献:

- [1] GIRET A, TRENTESAUX D, PRABHU V. Sustainability in manufacturing operations scheduling: a state of the art review[J]. *Journal of Manufacturing System*, 2015, 37: 126-140.
- [2] 张玉忠. 工件可拒绝排序问题综述[J]. *运筹学学报*, 2020, 24(2): 111-130.  
ZHANG Y Z. A survey on job scheduling with rejection[J]. *Operations Research Transactions*, 2020, 24(2): 111-130.
- [3] 郑斐峰, 隋杨, 徐寅峰. 考虑最小化最大完工时间间隔的三台平行机调度[J]. *系统工程理论与实践*, 2021, 41(4): 1025-1036.  
ZHENG F F, SUI Y, XU Y F. Three parallel machines scheduling with minimizing the maximum inter-completion time[J]. *Systems Engineering-Theory & Practice*, 2021, 41(4): 1025-1036.
- [4] 孙玉姣, 胡祥培, 曾庆成. 集装箱码头装卸作业顺序多目标优化调度模型[J]. *系统工程理论与实践*, 2020, 40(1): 195-209.  
SUN Y J, HU X P, ZENG Q C. A multi-objective optimization model for operation scheduling in container terminals[J]. *Systems Engineering-Theory & Practice*, 2020, 40(1): 195-209.
- [5] 金苗苗, 吴蒙洁, 罗文昌. 具有不可用区间且工件可拒绝下的单机重新排序问题的近似方案[J]. *运筹与管理*, 2021, 30(8): 87-92.  
JIN M M, WU M J, LUO W C. Approximation scheme for single machine rescheduling problem with an unavailable interval and job rejection[J]. *Operations Research and Management Science*, 2021, 30(8): 87-92.
- [6] 李敬花, 曹旺, 赵定刚, 等. 基于改进遗传算法的托盘拣选延误时间优化[J]. *计算机集成制造系统*, 2020, 26(2): 340-355.  
LI J H, CAO W, ZHAO D G, et al. Tardiness minimization of picking pallets based on improved genetic algorithm[J]. *Computer Integrated Manufacturing Systems*, 2020, 26(2): 340-355.
- [7] BISKUP D, HERRMANN J, GUPTA J N D. Scheduling identical parallel machines to minimize total tardiness[J]. *International Journal of Production Economics*, 2008, 115(1): 134-142.
- [8] CHE A D, ZHANG S B H, WU X Q. Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs [J]. *Journal of Cleaner Production*, 2017, 156: 688-697.
- [9] 李小林, 张松, 陈华平. 考虑分时电价的多目标批调度问题蚁群算法求解[J]. *中国管理科学*, 2014, 22(12): 56-64.  
LI X L, ZHANG S, CHEN H P. Solving multi-objective batch scheduling under TOU price using ant colony optimization[J]. *Chinese Journal of Management Science*, 2014, 22(12): 56-64.
- [10] 王君. 分时电价和时变行程时间下的供应链调度[J]. *中国机械工程*, 2018, 29(22): 2750-2757.  
WANG J. Supply chain scheduling under time-of-use electricity tariffs and time-dependent travel times[J]. *China Mechanical Engineering*, 2018, 29(22): 2750-2757.

- [11] WU X Q, CHE A D. A memetic differential evolution algorithm for energy-efficient parallel machine scheduling[J]. *Omega-International Journal of Management Science*, 2019, 82:155-165.
- [12] 许显杨, 陈璐. 考虑设备可靠性与能耗的平行机调度[J]. *上海交通大学学报*, 2020, 54(3):247-255.  
XU X Y, CHEN L. Parallel machine scheduling problem considering machine reliability and energy consumption[J]. *Journal of Shanghai Jiaotong University*, 2020, 54(3):247-255.
- [13] 耿凯峰, 叶春明. 考虑多时间因素的绿色可重入混合流水车间调度问题[J]. *计算机集成制造系统*, 2023, 29(1):75-90.  
GENG K F, YE C M. Green re-entrant hybrid flow shop scheduling problem considering multiple time factors[J]. *Computer Integrated Manufacturing System*, 2023, 29(1):75-90.
- [14] 赵福强, 刘桂庆. 考虑作业释放时间和机器数量变化的同型机调度问题[J]. *合肥工业大学学报(自然科学版)*, 2017, 40(9):1283-1288.  
ZHAO F Q, LIU G Q. Identical parallel machine scheduling problem with release dates and the change of the number of machines[J]. *Journal of Hefei University of Technology (Natural Science)*, 2017, 40(9):1283-1288.
- [15] KONG M, PEI J, LIU X B, et al. Green manufacturing: Order acceptance and scheduling subject to the budgets of energy consumption and machine launch[J]. *Journal of Cleaner Production*, 2020, 248:119300.
- [16] 李中胜, 杨玉中. 基于最小机器数的柔性作业车间调度研究[J]. *计算机工程与应用*, 2022, 58(2):281-288.  
LI Z S, YANG Y Z. Research on flexible job shop scheduling based on minimum number of machines [J]. *Computer Engineering and Applications*, 2022, 58(2):281-288.
- [17] CHEN Z L, LI C L. Scheduling with subcontracting options[J]. *IIE Transactions*, 2020, 40(12):1171-1184.
- [18] KRAVCHENKO S A, WERNER F. Minimizing the number of machines for scheduling jobs with equal processing times[J]. *European Journal of Operational Research*, 2009, 199(2):595-600.
- [19] ALIDAEE B, LI H. Parallel machine selection and job scheduling to minimize sum of machine holding cost, total machine time costs, and total tardiness costs[J]. *IEEE Transactions on Automation Science and Engineering*, 2014, 11(1):294-301.
- [20] WANG H, ALIDAEE B. Unrelated parallel machine selection and job scheduling with the objective of minimizing total workload and machine fixed costs[J]. *IEEE Transactions on Automation Science and Engineering*, 2018(4):1-9.
- [21] OSORIO-VALENZUELA L, PEREIRA L, QUEZADA F, et al. Minimizing the number of machines with limited workload capacity for scheduling jobs with interval constraints[J]. *Applied Mathematical Modelling*, 2019, 74:512-527.
- [22] DU J, LEUNG J Y T. Minimizing total tardiness on one processor is NP-hard[J]. *Mathematics of Operations Research*, 1990, 15(4):483-495.
- [23] KIRKPATRICK S, GELATT C D, VECCHI M. Optimization by simulated annealing[J]. *Science*, 1983, 220(4598):671-680.

## Operations Research and Cybernetics

### Parallel Machine Scheduling Problem Concerned with Service Level and Resource Consumption

FENG Xin<sup>1</sup>, CHEN Nishan<sup>2</sup>, QIN Lan<sup>3</sup>

(1. College of Economics and Management, Nanjing Forestry University, Nanjing 210037;

2. School of Humanities and Social Sciences, East China Jiaotong University, Nanchang 330013;

3. Logistics and E-Commerce College, Zhejiang Wanli University, Ningbo Zhejiang 315100, China)

**Abstract:** [Purposes] Aiming at realizing the transformation of resource-saving manufacturing industry, a parallel machine scheduling model is established to minimize the sum of total delay and machine using cost. [Methods] According to the characteristics of the problem, an improved simulated annealing algorithm is developed to solve the model. The computation performance of the developed algorithm is tested by different scale instances. The sensitivity analysis of the problem parameters is carried out as well. [Findings] The developed algorithm can achieve a good computational efficiency and solution quality. [Conclusions] Sensitivity analysis shows that increasing resource reserves can effectively realize the transformation of resource-saving manufacturing industry while improving the customer service level of manufacturing enterprises. The established model can provide decision support for achieving resource-conservation manufacturing system.

**Keywords:** parallel machine scheduling; job scheduling; resource consumption; simulated annealing algorithm

(责任编辑 黄颖)