

# 基于 T-SSA 算法的流水车间订单调度问题研究

王 婷, 毋 涛

(西安工程大学 计算机科学学院, 陕西 西安 710048)

**摘 要:**针对目前服务制造型订单企业的生产调度优化问题,考虑企业资源和加工生产线的限制,以最小化最大订单完工时间为目标函数,建立流水车间订单调度模型,并模拟麻雀觅食过程提出了一种两段式麻雀搜索(two-vector sparrow search algorithm, T-SSA)算法。根据订单调度问题的特点,该文采用生产环节生产线两段式编码方式对个体进行编码;使用权重轮盘赌随机选择机制(考虑订单收益高低、交期紧急程度、订单权重)初始化麻雀种群,保证种群的多样性和质量;并设计麻雀搜索算法中的智能行为,包括发现者移动机制、跟随者跟随机制、警戒者侦察预警机制,防止算法陷入局部最优。最后,仿真类似算法进行类比分析,实验验证了 T-SSA 的有效性,且求解效率也显著提升;并将 T-SSA 对应用到上海某西装定制企业订单调度实例中,结果验证了 T-SSA 算法求解订单调度问题的可行性。

**关键词:**生产环节生产线两段式编码;麻雀搜索算法;流水车间订单调度;最小化最大订单完工时间

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2021)09-0182-07

doi:10.3969/j.issn.1673-629X.2021.09.031

## Research on Order Scheduling of Flow Shop Based on T-SSA

WANG Ting, WU Tao

(School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China)

**Abstract:** Aiming at the current production scheduling optimization problem of service manufacturing order companies, considering the constraints of enterprise resources and processing production lines, and minimizing the maximum order completion time as the objective function, a flow shop order scheduling model is established and a sparrow foraging process, two-vector sparrow search algorithm (T-SSA), is proposed. According to the characteristics of the order scheduling problem, we use a two-stage coding method in the production process to code individuals, with a weighted roulette random selection mechanism (considering the level of order revenue, delivery urgency, and order weight) to initialize the sparrow population to ensure population diversity and quality. Then we design the intelligent behavior in the sparrow search algorithm, including the movement mechanism of the finder, the follower mechanism and the alerter detection and early warning mechanism to prevent the algorithm from falling into the local optimum. Finally, analogous analysis is carried out by simulating similar algorithms, and the experiment verifies the effectiveness of T-SSA, and the solution efficiency is also significantly improved. The T-SSA pair is applied to the order scheduling example of a suit customization enterprise in Shanghai, and the result verifies the feasibility of solving the order scheduling problem by T-SSA.

**Key words:** two-stage coding of production processes production line; sparrow search algorithm; flow shop order scheduling; minimizing maximum order completion time

## 0 引 言

随着中国经济社会的发展,产品定制服务的出现,消费者个性化需求也日益凸显,有一部分制造型企业开始采用订单生产模式<sup>[1]</sup>(make-to-order, MTO),比如西装定制、船舶制造。

在 MTO 生产模式中,订单内不同的工件的交货期不同,而制造型企业通常都是配备固定的车间、机器和人员,订单调度决策往往受车间生产能力的制约。在这种情况下,这种决策往往会与实际生产有所偏差,

造成计划外的生产成本(如加班、外包)和拖期罚金,导致企业信誉受损,客户满意度降低。

为提高订单交付的准时率,企业需要在有限产能下对订单生产作业进行合理的计划安排和调度,依据订单产品的要求交期及时交付产品,履行合约。研究基于流水线车间的订单调度问题可以有效提高企业的生产效率和竞争力,对于处于 MTO 生产模式的企业制定出合理有效的订单调度决策,及时交付作业具有重要的研究意义。

收稿日期:2020-09-01

修回日期:2021-01-06

基金项目:陕西省科技成果转化与推广计划项目(2019CGXNG-018)

作者简介:王 婷(1994-),女,硕士研究生,通讯作者,研究方向为智能信息处理;毋 涛,副教授,研究方向为智能信息处理。

长期以来,混合流水车间调度问题<sup>[2]</sup> (hybrid flow shop scheduling problem, HFSP) 被证明是一个经典的 NP 难题。目前,求解 HFSP 的算法主要包括遍历式算法、构造型算法和智能化优化算法,其中遍历式算法可以求解出该类问题的精确解,但计算速度较慢;构造型算法在求解该类问题时运算速度较快,但算法结构复杂且通常无法求解出全局最优解<sup>[3]</sup>;优化算法有严格的理论基础,能在短时间内解决问题的最优或理想解。智能优化算法<sup>[4]</sup>主要有混合萤火虫算法 (hybrid firefly algorithms, HFA)<sup>[5]</sup>、最佳觅食算法 (optimal foraging algorithm, OFA)<sup>[6]</sup>、离散狼群算法 (discrete wolf pack algorithm, DWPA)<sup>[7]</sup>、离散布谷鸟算法 (cuckoo search, CS)<sup>[8]</sup>、改进蛙跳算法 (shuffled frog leaping algorithm, SFLA)<sup>[9]</sup>、果蝇优化算法 (fruit fly optimization algorithm, FOA)<sup>[10]</sup>、混合鱼群算法 (hybrid artificial fish swarm algorithm, HAFSA)<sup>[11]</sup>、人工蜂群算法 (artificial bee colony algorithm, ABC)<sup>[12]</sup> 等等,已经广泛应用于解决各种生产调度问题。

麻雀搜索算法<sup>[13]</sup> (sparrow search algorithm, SSA) 是一种新型的群智能优化算法,主要是受麻雀的觅食行为和反捕食行为的启发。针对 SSA 的研究表明,该算法在不同的搜索空间中具有良好的性能。在 SSA 的基础上,文中提出了基于生产环节生产线两段式麻雀搜索算法 (two-vector sparrow search algorithm, T-SSA),并将 T-SSA 实际应用于流水车间订单调度问题中,验证了该算法的有效性。

## 1 问题描述与建模

### 1.1 问题描述

一次生产任务中有  $n$  个相对独立的工件,在加工过程中需要按照相同的方向在  $s$  个生产环节上进行加工。工件来自  $H$  订单,每个订单包含  $h$  个工件,  $Q_h$  ( $1 \leq h \leq H$ ) 表示订单的工件集合,即  $Q_1 \cup Q_2 \cup \dots \cup Q_n = \{1, 2, \dots, n\}$ ,每个工件的工艺路线是相同的,所有的工件在交付给客户之前必须经过每个加工节点;订单  $h$  的权重  $\omega_h$  根据产品价值、客户等级和实际环境中订单对应的利润水平来确定。假设一个制造企业的生产系统由  $m$  条生产线组成,每个生产环节至少有一种并联生产线,每一种并联生产线都有特殊的工艺约束,每一个工件在各生产环节中都有一整套合适的生产线;记工件  $i$  属于订单  $h$ ,其交货日期为  $D_i^h$ ,工件收入为  $Q_i^h$ ,完成时间为  $C_i^h$ 。

### 1.2 模型的假设条件

(1) 在初始时间 (设置为 0), 可以处理任何订单任务。

(2) 同一生产线的同一时间只能加工 1 件产品。

(3) 不同的订单任务之间没有先后约束。

(4) 在生产过程中,订单前一个生产环节被处理完成后,会立即进入下一个生产环节进行处理;如果下一个生产环节生产线正在工作,则订单进行等待。

(5) 每两个处理环节之间有足够的缓存空间,允许产品在处理环节之间停留和等待。

(6) 订单调度过程为非抢占式调度。

(7) 订单任务中各生产环节的交货时间和利润值均为已知量 (交货时间从初始时间 0 开始计算)。

根据上述解释和假设,研究的问题是:确定订单的加工顺序和各生产环节的生产线分配,从而使加权订单完成时间最小化。

### 1.3 参数和变量说明

$i$ : 表示待加工工件编号。

$j$ : 表示待加工的生产环节序号。

$k$ : 表示每个生产环节的生产线编号。

$M_j$ : 第  $j$  个生产环节并行生产线的集合。

$N_{M_j}$ : 第  $j$  个生产环节并行生产线的数量。

$MP_{ij}$ : 工件  $i$  在第  $j$  个生产环节可用的生产线集合,  $MP_{ij} \subseteq M_j$ 。

$t_{ijk}$ : 工件  $i$  的第  $j$  个生产环节在生产线  $k$  上的加工时间。

$st_{ijk}$ : 工件  $i$  的第  $j$  个生产环节在生产线  $k$  上的开始加工时间。

$et_{ijk}$ : 工件  $i$  的第  $j$  个生产环节在生产线  $k$  上的结束加工时间。

$x_{ijk}$ : 当工件  $i$  的第  $j$  个生产环节在生产线  $k$  上的加工时  $x_{ijk} = 1$ ; 否则  $x_{ijk} = 0$ 。

$r_{jk}$ : 第  $j$  个生产环节在生产线  $k$  上加工的第  $r$  个任务的工件编号。

$T_i$ : 工件  $i$  的各生产环节加工时间总和。

### 1.4 模型构建

根据问题描述,采用最小最大订单完成时间为目标函数,可转化为求解订单内工件集的最小化完成时间,即一个工件完成时间的最小值,模型构建具体如下:

$$F_i^h = \min(\max(C_i^h)) \quad (1)$$

模型描述如下:

$$\begin{aligned} et_{ijk} &\leq st_{i(j+1)k}, i = 1, 2, \dots, n; j = 1, 2, \dots, s; \\ k &\in MP_{ij}; k^* \in MP_{i(j+1)} \end{aligned} \quad (2)$$

$$\begin{aligned} et_{ijk} &= st_{ijk} + t_{ijk}; i = 1, 2, \dots, n; j = 1, 2, \dots, s; \\ k &\in MP_{ij} \end{aligned} \quad (3)$$

$$\sum_k x_{ijk} \leq 1; i = 1, 2, \dots, n; j = 1, 2, \dots, s; k \in MP_{ij} \quad (4)$$

$$et_{ijk} \leq st_{(r+1)jk}; j = 1, 2, \dots, s; k \in M_j; r_{jk} \in \{i\}_{i=1}^k,$$

$$r = 1, 2, \dots, N_{jk} - 1 \quad (5)$$

若  $r_{jk} = i$ , 则:

$$st_{ijk} = \begin{cases} 0, & j = 1, r = 1 \\ et_{(r-1)jk}, & j = 1, r > 1 \\ et_{i(j-1)k}, & r = 1, j > 1 \\ \max(et_{(r-1)jk}, et_{i(j-1)k}), & r > 1, j > 1 \end{cases}$$

其中:

$$k \in MP_{ij}, k^- \in MP_{i(j-1)}, r = 1, 2, \dots, N_{jk} \quad (6)$$

$$C_i^h = et_{isk}; i = 1, 2, \dots, n; k^s \in MP_{is} \quad (7)$$

$$x_{ijk} \in \{0, 1\}, x_{ijk^-} = 1, \forall i, j, k, r \quad (8)$$

$$P_{ij} = \sum_{k=1}^{M_j} x_{ijk} \times t_{ijk} \quad (9)$$

公式(1)是最小化工件完成时间的公式和目标函数;公式(2)表示在前一个生产环节完成后,才可以在下一个生产环节中进行加工;公式(3)表示每个工件的加工完成时间,其中每个生产环节的加工结束完成时间取决于该生产环节的开始时间和加工时长;公式(4)表示工件的每个生产环节只能选择该生产环节的一台生产线进行加工;公式(5)表示一条生产线同时只能加工一个工件;公式(6)表示当工件  $i$  是第  $j$  个生产环节在生产线  $k$  上处理的第  $r$  个任务时,生产环节开始时间取决于该工件前一个生产环节的结束时间与本生产线的前一任务结束时间;公式(7)表示每个工件的完成时间等于工件在最后一个生产环节中的完成时间;公式(8)为变量取值范围;公式(9)表示每个工件每个生产环节加工时间的计算关系,即工件的总加工时间。

## 2 麻雀搜索算法(SSA)

麻雀搜索算法是一种基于麻雀觅食行为和反捕食行为的新群体智能优化算法,其仿生原理如下:

在麻雀觅食的过程中,将整个麻雀种群分为发现者和加入者两种角色,叠加了侦查预警机制。发现者在种群中负责寻找食物并为整个麻雀种群提供觅食区域和方向,而加入者则是利用发现者来获取食物。为了获得食物,麻雀通常可以采用发现者和加入者这两种行为策略进行觅食。种群中的个体会监视群体中其他个体的行为,并且该种群中的攻击者会与高摄取量的同伴争夺食物资源,以提高自己的捕食率。此外,当麻雀种群受到捕食者的攻击时会做出反捕食行为。

SSA 算法的具体流程见图 1。

Step1: 初始化麻雀种群。

Step2: 设置发现者规模,将种群分为发现者和跟随者。

Step3: (发现者位置更新)当预警值小于安全值的时候,即觅食环境没有捕食者时,发现者位置随机更

新;若预警值大于安全值,种群中的一些麻雀已经发现了捕食者,并向种群中其他麻雀发出了警报,此时所有麻雀都需要迅速飞到其他安全的地方进行觅食。更新方式如公式(10):

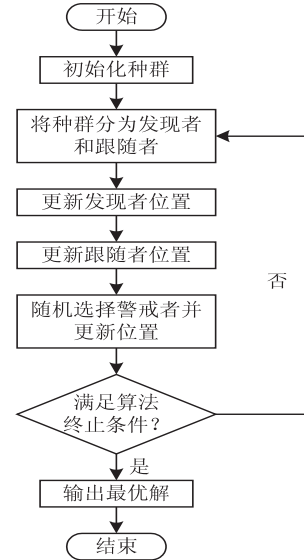


图 1 麻雀搜索算法流程

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \times \exp\left(\frac{-i}{\alpha \times \text{iter}_{\max}}\right), & R_2 < ST \\ X_{i,j}^t + Q \times L, & R_2 \geq ST \end{cases} \quad (10)$$

上式表示种群中第  $t$  代中的第  $i$  个个体的第  $j$  维位置信息,  $\alpha$  和  $Q$  是服从不同分布的随机数。 $\text{iter}_{\max}$ 、 $R_2$ 、 $ST$  分别表示最大迭代次数、预警值和安全值。

Step4: (跟随者位置更新)若当前跟随者处于十分饥饿的状态,则需要飞往其他地方觅食,以获得更多的能量,进行位置更新;否则,跟随者是围绕最好的发现者周围进行觅食,期间也有可能发生食物争夺,使自己成为生产者。更新方式如公式(11):

$$X_{i,j}^{t+1} = \begin{cases} Q \times \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{l^2}\right), & i > n/2 \\ X_p^{t+1} + |X_{i,j}^t - X_{i,j}^{t+1}| \times A^+ \times L, & i \leq n/2 \end{cases} \quad (11)$$

其中,  $X_p$  是目前发现者的最优位置,  $X_{\text{worst}}$  是当前全局最差的位置,  $n$  是种群大小。

Step5: (随机选择警戒者并更新位置)当任意麻雀意识到危险靠近时,它们会放弃当前的食物,即无论该麻雀是发现者还是跟随者,都将放弃当前的食物而移动到一个新的位置,处于种群外围的麻雀向安全区域靠拢,处于种群中心的麻雀则随机行走靠近别的麻雀。更新规则如公式(12):

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t \times \beta \times |X_{i,j}^t - X_{\text{best}}^t|, & f_i > f_g \\ X_{i,j}^t + K \times \left[ \frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon} \right], & f_i = f_g \end{cases} \quad (12)$$

其中,  $X_{best}$  是当前的全局最优位置,  $f_g$  和  $f_w$  分别是当前全局最佳和最差的适应度值,  $f_i$  是当前麻雀的适应度值,  $\varepsilon$ 、 $\beta$ 、 $K$  是不同随机常数。

Step6: 判断是否达到最大迭代次数, 若达到, 则输出最优适应度值, 即所求问题的最优解, 否则转到 Step2。

### 3 基于 T-SSA 算法的模型求解

针对流水车间订单调度问题采用基于生产环节和生产线的两段式编码方式来表示麻雀种群中麻雀的位置。应用其设计的初始化种群方式初始化种群, 对麻雀的智能行为进行重新设计; 则基于两段式麻雀搜索算法的流水车间订单调度问题的具体操作如下:

#### 3.1 编码方式

根据流水车间订单调度问题的特点, 编码方式不仅要能表示出各个工件在每个生产环节中的加工顺序, 还要能表示出各工件在每个生产环节中所选择的加工生产线, 因此在 T-SSA 中, 麻雀个体采用生产环节生产线两段式的编码方式, 其中两段的长度相等, 分别对应生产环节的分配方案和生产线的排序方案。当待加工的工件总数为  $n$ , 工件  $n_i$  的加工生产环节为  $m_i$  时, 则个体表示长度为  $2 \sum_{i=1}^k n_i m_i$  的整数串, 即所有工件总生产环节的 2 倍。编码的第一段是生产环节编码, 第二段是生产线编码。假设有三个工件, 每个工件有三个生产环节, 第一个生产环节共有两台并行生产线 ( $M_1, M_2$ ), 第二个生产环节有两台并行生产线 ( $M_3, M_4$ ), 第三个生产环节也是两台并行生产线 ( $M_5, M_6$ )。假设工件的两段式编码示意图如图 2 所示, 图中 1 到 9 位为生产环节编码, 10 到 18 位为生产线编码。

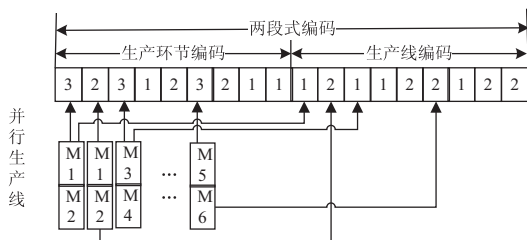


图 2 两段式编码示意图

生产环节编码的编码长度为总生产环节数, 基因位上的数字表示工件号, 相同工件号出现的次数对应工件的第几个生产环节。图 2 中表示的工件加工顺序为: 工件 3 → 工件 2 → 工件 3 → 工件 1 → 工件 2 → 工件 3 → 工件 2 → 工件 1 → 工件 1, 其中每个工件出现三次, 即每个工件有 3 个生产环节, 比如基因位 1 上的数字 3, 表示工件 3 的第一个生产环节, 基因位 3 上的数字 3, 表示工件 3 的第二个生产环节, 诸如此类。

生产线编码的编码长度与生产环节编码长度一致, 也是总生产环节数, 基因位上的数字代表前半段对应的生产环节可以选择的生产线集合中的第几条生产线。图 2 中表示选择的生产线顺序依次为: 生产线  $M_1$  → 生产线  $M_2$  → 生产线  $M_3$  → 生产线  $M_1$  → 生产线  $M_4$  → 生产线  $M_6$  → 生产线  $M_5$  → 生产线  $M_4$  → 生产线  $M_6$ , 基因位 10 位上的数字 1 表示工件 3 的第一个生产环节选择生产线  $M_1$  加工, 基因位 12 上的数字 1 表示工件 3 的第二个生产环节上选择生产线  $M_3$  加工, 以此类推。

[3, 2, 3, 1, 2, 3, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2] 整个整数段表示一只麻雀个体。

#### 3.2 种群初始化

在麻雀搜索算法中, 多样性好的初始解集能够有效提高运算效率, 扩大搜索范围, 避免局部收敛的情况发生。考虑到两段式编码的特点, 先采用随机初始化方式产生足够多的麻雀个体, 确保种群的多样性, 然后使用轮盘赌选择机制选取所需数量的初始个体, 确保种群的质量。采用权重法计算个体被选取的概率, 优先选择收益高、交期紧、订单权重大的工件个体, 工件选择概率越大, 个体越容易被选用。

$$Z_i^h = \frac{Q_i^h}{\sum_{h=1}^H \sum_{i=1}^{h_s} Q_i^h} + \frac{D_i^h}{\sum_{h=1}^H \sum_{i=1}^{h_s} D_i^h} + \omega_h \quad (13)$$

$$P_i^h = \frac{Z_i^h}{\sum_{i=1}^n Z_i^h} \quad (14)$$

其中, 公式 (13) 表示的是每个工件个体被选择的权重, 符号  $Z_i^h$  表示工件编号  $i$  在订单号  $h$  上的权重值;  $Q_i^h$  表示订单  $h$  中工件编号  $i$  的收益;  $D_i^h$  表示订单  $h$  中工件编号  $i$  的交货期。公式 (14) 表示每个工件被选择的概率, 符号  $P_i^h$  表示订单  $h$  中工件  $i$  的权重选择概率。

#### 3.3 适应度函数

以最小化最大订单完成时间为目标函数求解流水车间订单调度问题, 主要思想是订单内的每个工件的完成时间最优, 可以将该问题转化为求解订单内工件集的最小化最大完成时间为适应度函数, 工件最大完成时间越小, 表示个体的适应度值越好, 即  $\text{fitness} = \min \{C_{\max}(P)\}$ 。

#### 3.4 智能行为设计

文中以最小化最大订单完工时间为目标函数来确定最优订单调度方案, 即订单中工件的生产环节排序和生产线选择方案。结合 HFSP 的特性, 对两段式麻雀搜索算法中的具体定义如下:

(1) 最优最差麻雀选择机制。

记录第  $i$  只麻雀个体的适应度函数为  $\text{Fit}_i$ , 根据适应度值对麻雀个体进行排序, 那么取全局最差适应



度值为  $worseF = \max(Fit_i), i = 1, 2, \dots, n$ ,  $worseF$  对应的全局最差位置为  $worseX$ ; 最优麻雀个体对应的适应度值为  $bestF1 = \min(Fit_i), i = 1, 2, \dots, n$ ,  $bestF1$  对应的全局最优位置为  $bestX1$ 。

### (2) 发现者移动机制。

根据设置麻雀发现者个体的比例  $PR$ , 从种群中随机取出发现者的个体数量。将随机预警值  $R_2$  与发现者警戒阈值  $ST$  进行比较, 若  $R_2 < ST$ , 说明附近没有捕食者出现, 发现者个体在附近小范围移动; 若  $R_2 \geq ST$ , 说明有捕食者出现威胁到了种群的安全, 需要到其他地方觅食, 所有发现者个体位置进行大幅移动。淘汰边界个体, 计算新发现者群体的新的适应度值, 寻找最优适应度值  $bestX2$ , 对应的全局最优位置为  $bestF2$ 。

### (3) 跟随者跟随机制。

对去除发现者种群个体为跟随者个体, 种群数量为  $pop$ , 跟随者个体的编号为  $i$ 。若  $i > \frac{pop}{2}$ , 表示该麻雀处于十分饥饿的状态, 能量很低, 适应度值比较差, 需要到其他地方去觅食, 进行位置更新; 若  $i \leq \frac{pop}{2}$ , 代表围绕最好的发现者个体  $bestF2$  进行觅食, 期间有可能发生食物争夺, 导致跟随者成为发现者。淘汰边界个体, 并计算跟随者的适应度值。

### (4) 侦察预警机制(警戒者)。

对意识到危险的个体称为警戒者, 并不代表出现了真正的捕食者。随机指定个体  $j$ , 取该个体的适应度值为  $Fit_j$ , 若  $Fit_j > bestX1$ , 表示此时的麻雀正处于种群的边缘, 极易受到捕食者的攻击, 位置跳跃性比较大, 警戒者位置更新; 若  $Fit_j = bestX1$ , 表明处于种群中间的麻雀意识到了危险, 需要靠近其他的麻雀以此尽量减少它们被捕食的风险。

## 3.5 T-SSA 算法迭代过程

根据智能行为设计的 4 大机制, 可将使用 T-SSA 算法求解流水车间订单调度问题的步骤概括如下:

Step1: 初始化算法相关参数。设置种群的规模大小为  $pop$ , 最大迭代次数为  $iter_{max}$ , 发现者比例为  $PR$ , 侦查者比例为  $SD$ , 预警值为  $R_2$ , 发现者警戒阈值为  $ST$ 。

Step2: 根据初始化方案进行种群初始化, 确定麻雀个体的位置编码。

Step3: 依据设定的  $PR$ , 将种群分为发现者和跟随者。计算个体的适应度值  $Fit_i$ , 进行排序, 并取出种群最优位置  $bestX1$  和最差位置  $worseF$  对应的适应度值  $bestF1$ 、 $worseF$ 。

Step4: 将预警值  $R_2$  与警戒阈值进行比较, 更新所

有发现者位置, 淘汰掉边缘个体, 迭代并计算发现者的适应度值。取出发现者的最优位置  $bestX2$  对应的适应度值  $bestF2$ 。

Step5: 依据跟随者的编号, 判断跟随者的状态, 淘汰边缘个体, 跟随者进行随机位置更换, 或者抢夺最优发现者  $bestX2$  成为发现者。迭代, 计算所有跟随者的适应度值。

Step6: 判断随机警戒者所处位置是否在种群中心, 若处在中心, 向最差位置  $worseF$  个体移动位置, 淘汰边界个体, 更新警戒者的适应度值。

Step7: 求解最新全局个体适应度值, 替换最优位置  $bestX1$  及其响应的适应度值  $bestF1$ 。

Step8: 判断是否达到最大迭代次数, 若达到, 则输出最优适应度值, 即所求问题的最优解, 否则转到 Step4。

## 4 算法验证与分析

实验仿真环境为操作系统 Windows8、处理器 Intel (R) Core (TM) i5-4210U CPU @ 1.70 GHz 2.40 GHz、内存 12G, 采用 Matlab R2018a。

### 4.1 算法仿真验证

采用以下两种方式验证算法的有效性:

#### (1) 实例验证一。

引用文献[14]的数据进行算法测试, 参数与 SA-PSO-GA 相同, 取麻雀个体数量为 100, 迭代次数为 150, 发现者警戒阈值为 0.8, 发现者比例为 20%。T-SSA 算法的甘特图如图 3 所示。图上的任务名称表示某个工件的第几个生产环节, 比如 401 代表工件一的第一个生产环节。

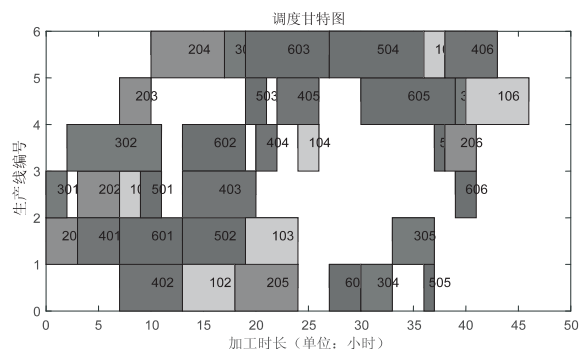


图 3 TSSA 算法测试甘特图

由图可知, 两段式麻雀搜索算法甘特图调度结果显示 46 小时可完成 6 个工件的生产, 与文献[13]中优化的 SA-PSO-GA 调度效果相同, 均优于传统的 PSO-GA 算法。将 T-SSA 算法的迭代曲线与原文献比较, PSO-GA 算法和 SA-PSO-GA 更易陷入局部最优, 输出的解并非实际最优解。从运行时间上来看, 由于 T-SSA 达到最优的迭代次数较少, 运算时间为 14-17S,

在效率上明显高于原文中提到的算法。

(2)实例验证二。

设置 9 种不同规模的调度算例,随机生成订单数量、订单权重、订单收益,订单中的工件数量合计分别为 20、50、100,生产环节数为 2、4、8,生产线条数分别合计为 5、10、20,工件交期根据 EDD 规则<sup>[15]</sup>与经验设置。数据结果与 4.2 中的数据格式类似。对生成的订单数据进行处理,得到算例规模  $n \times s(m)$ ,  $n$  表示工件数,  $s$  表示生产环节数,  $m$  表示生产线总数。

使用相同的算法参数和编码方式对人工蜂群算法(ABC)<sup>[12]</sup>、两段式麻雀搜索算法(T-SSA)进行 9 次实例验证比较。仿真结果如表 1 所示,在不同规模的算例中,T-SSA 求解的最小值、平均值基本优于 ABC 算法,体现出了明显的搜索能力。

表 1 实例运行结果统计(最小值、平均值单位:小时)

| 算例编号 | 规模<br>$n \times s(m)$ | ABC   |         | T-SSA |         |
|------|-----------------------|-------|---------|-------|---------|
|      |                       | 最小值   | 平均值     | 最小值   | 平均值     |
| 1    | 20×2(5)               | 52    | 53.1    | 43    | 46.2    |
| 2    | 20×4(10)              | 285   | 293.5   | 258   | 276.3   |
| 3    | 20×8(20)              | 643   | 649.3   | 635   | 644.6   |
| 4    | 50×2(5)               | 126   | 131.4   | 115   | 122.5   |
| 5    | 50×4(10)              | 527   | 536.7   | 503   | 512.4   |
| 6    | 50×8(20)              | 1 159 | 1 167.3 | 1 156 | 1 160.1 |
| 7    | 100×2(5)              | 317   | 328.4   | 296   | 302.2   |
| 8    | 100×4(10)             | 1 043 | 1 069.2 | 1 033 | 1 052.3 |
| 9    | 100×8(20)             | 2 291 | 2 313.4 | 2 285 | 2 230.2 |

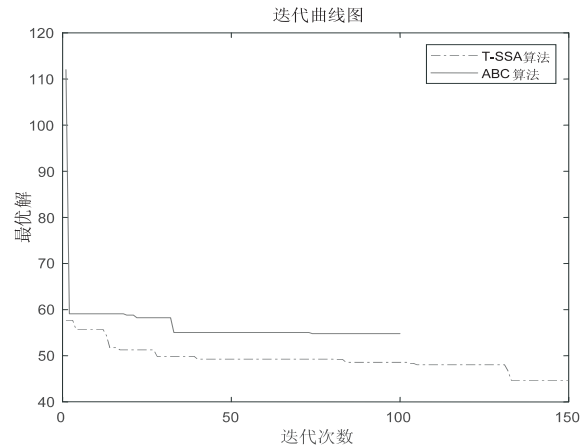


图 4 订单调度迭代图

其中某一次的迭代曲线如图 4 所示,实验结果基本类似,ABC 的最优解 T-SSA 均可在相近迭代次数到达,且最优解的消耗时长接近。结合表 1 和图 5 可以看出,T-SSA 的最优解质量更好,表明该算法的搜索能力更强,更进一步证明了该算法的有效性和优越性。

4.2 订单调度实例应用验证

结合上海某西装定制企业的流水车间生产过程为

实例作为研究对象,验证订单调度模型和两段式麻雀算法的有效性。该生产企业的西装定制生产过程可描述为有限并行生产线混合流水车间,所有西装定制产品从加工到结束,主要分为五个环节:裁剪有 2 条生产线( $M_1, M_2$ );缝制有 3 条生产线( $M_3, M_4, M_5$ )、手缝有 3 条生产线( $M_6, M_7, M_8$ )、整烫有 2 条生产线( $M_9, M_{10}$ ),检验有 2 条生产线( $M_{11}, M_{12}$ )。此处忽略不同工件的规格、生产工艺在每个加工环节的局部差异。本次选取该企业一次实际生产任务为实验对象,共有 8 个客户订单,合计 20 个待加工工件,订单权重、交货期、工件收益已知。假设所有订单在 0 时刻接受,且所有生产线环境运作正常,物料库存充足。模拟订单信息数据如表 2 所示,表 3 为订单对应的西装企业生产工时数据。

表 2 订单信息

| 订单编号 | 订单权重  | 工件编号 | 交货期<br>(D/天) | 工件收益<br>(Q/万元) |
|------|-------|------|--------------|----------------|
| 1    | 0.060 | 1    | 60           | 7              |
|      |       | 2    | 90           | 11             |
|      |       | 3    | 150          | 6              |
|      |       | 4    | 60           | 5              |
| 2    | 0.200 | 5    | 90           | 18             |
|      |       | 6    | 120          | 6              |
| 3    | 0.150 | 7    | 60           | 6              |
|      |       | 8    | 120          | 14             |
|      |       | 9    | 60           | 18             |
| 4    | 0.250 | 10   | 120          | 9              |
|      |       | 11   | 150          | 7              |
| 5    | 0.080 | 12   | 120          | 11             |
|      |       | 13   | 90           | 9              |
| 6    | 0.100 | 14   | 60           | 7              |
|      |       | 15   | 90           | 9              |
| 7    | 0.125 | 16   | 150          | 11             |
|      |       | 17   | 60           | 11             |
| 8    | 0.035 | 18   | 120          | 2              |
|      |       | 19   | 200          | 7              |
|      |       | 20   | 240          | 6              |

使用 T-SSA 算法进行实例调度,设置种群数量分别为 30、60、100,最大迭代次数为 500,发现者警戒阈值为 0.8,发现者比例为 20%,重复多次计算以上订单实例。

表 3 上海某西装定制企业生产工时(单位:小时)

| 工<br>件 | 裁剪 |    | 缝制 |    |    | 手缝 |    |    | 整烫 |     | 检验  |     |
|--------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
|        | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 | M12 |
| 1      | 20 | 19 | 29 | 32 | 30 | 22 | 21 | 23 | 15 | 16  | 15  | 14  |
| 2      | 28 | 27 | 35 | 38 | 36 | 30 | 29 | 32 | 21 | 20  | 18  | 19  |
| 3      | 25 | 23 | 31 | 33 | 34 | 25 | 24 | 26 | 19 | 22  | 17  | 17  |

续表 3

| 工<br>件 | 裁剪 |    | 缝制 |    |    | 手缝 |    |    | 整烫 |     | 检验  |     |
|--------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
|        | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 | M12 |
| 4      | 18 | 19 | 27 | 29 | 28 | 19 | 21 | 22 | 15 | 14  | 13  | 14  |
| 5      | 22 | 20 | 27 | 28 | 26 | 21 | 23 | 24 | 18 | 19  | 16  | 17  |
| 6      | 36 | 34 | 46 | 45 | 43 | 38 | 39 | 41 | 2  | 30  | 29  | 31  |
| 7      | 21 | 24 | 31 | 35 | 34 | 24 | 26 | 27 | 17 | 18  | 16  | 14  |
| 8      | 22 | 22 | 30 | 33 | 32 | 23 | 24 | 25 | 19 | 21  | 17  | 19  |
| 9      | 31 | 33 | 43 | 42 | 41 | 39 | 36 | 38 | 25 | 27  | 26  | 27  |
| 10     | 21 | 23 | 27 | 29 | 28 | 23 | 23 | 22 | 15 | 15  | 14  | 18  |
| 11     | 20 | 19 | 27 | 27 | 26 | 21 | 24 | 23 | 16 | 17  | 17  | 15  |
| 12     | 22 | 20 | 29 | 30 | 28 | 23 | 21 | 22 | 17 | 18  | 15  | 16  |
| 13     | 20 | 18 | 32 | 34 | 33 | 23 | 25 | 26 | 19 | 16  | 17  | 17  |
| 14     | 21 | 22 | 29 | 28 | 28 | 23 | 24 | 24 | 15 | 17  | 14  | 16  |
| 15     | 20 | 19 | 29 | 31 | 30 | 22 | 23 | 24 | 14 | 16  | 15  | 17  |
| 16     | 32 | 31 | 45 | 43 | 44 | 37 | 35 | 36 | 22 | 23  | 23  | 26  |
| 17     | 24 | 23 | 35 | 37 | 36 | 26 | 29 | 25 | 20 | 19  | 19  | 21  |
| 18     | 29 | 30 | 40 | 38 | 39 | 32 | 34 | 33 | 22 | 21  | 20  | 22  |
| 19     | 21 | 22 | 37 | 35 | 36 | 28 | 25 | 27 | 19 | 17  | 18  | 16  |
| 20     | 19 | 20 | 29 | 32 | 30 | 23 | 23 | 21 | 16 | 17  | 14  | 15  |

各种群规模随机取 20 次运算结果统计如表 4 所示,最优迭代次数均在 150 代以内,运算时间较短。实际结果表明,种群增大可以有效减少迭代次数,得到目标值。

表 4 实例运行结果统计

| 种群<br>规模 | 迭代次数 |     | 完成时间(小时) |     | 平均值 | 方差       |
|----------|------|-----|----------|-----|-----|----------|
|          | 最优   | 最差  | 最优       | 最差  |     |          |
| 30       | 137  | 280 | 405      | 438 | 421 | 16.507 6 |
| 60       | 84   | 235 | 399      | 430 | 414 | 15.517 7 |
| 100      | 60   | 257 | 396      | 421 | 408 | 12.510 1 |

目前该企业还是通过传统的 Excel 和人为经验制作订单生产排程,人工负荷调整工作量巨大,采用这种方式进行以上实例调度编排企业应用总共耗时 450 分钟,使用 T-SSA 算法运行实例,最差结果也是少于 450 分钟的,验证了两段式麻雀搜索算法结果流水车间订单调度问题的实用性和有效性。

## 5 结束语

针对流水车间订单调度问题,建立单目标数学模型,采用两段式双层编码的麻雀搜索算法进行问题求解,对发现者、跟随者、警戒者三种角色采用不同的机制更新策略。算法很好地挖掘了全局最优潜在区域的能力,从而有效地避免了局部最优问题。将 T-SSA 算法与文献[10,12]中的算法进行模拟比较,验证了算法的有效性和优越性。采用企业订单调度实例测试 T-SSA,更进一步验证了 T-SSA 的有效性和实用性。不足之处是,T-SSA 在求解大规模应用时,耗时稍微

长一些,可对算法进行进一步的研究,将其扩展使用在更复杂的调度问题中,例如多目标调度问题。

## 参考文献:

- [1] KANDA S, TAKAHASHI K, MORIKAWA K. A flexible service rule for the dynamic make-to-stock/make-to-order hybrid production system[J]. Procedia Manufacturing, 2015, 2:46-50.
- [2] GUPTA J N D. Two-stage, hybrid flowshop scheduling problem[J]. Journal of the Operational Research Society, 1988, 39(4):359-364.
- [3] HAMADAMEEN A O, HASSAN N. Pareto optimal solution for multiobjective stochastic linear programming problems with partial uncertainty[J]. International Journal of Mathematics in Operational Research, 2018, 12(2):139-166.
- [4] SHI Yuhui. Developmental swarm intelligence: developmental learning perspective of swarm intelligence algorithms[J]. International Journal of Swarm Intelligence Research, 2014, 5(1):36-54.
- [5] 裴小兵, 张睿, 于秀燕. 混合萤火虫算法求解多目标置换流水车间调度问题[J]. 信息与控制, 2020, 49(4):478-488.
- [6] 杜士卿, 朱光宇, 徐文婕. 最佳觅食算法求解多目标混合流水车间调度问题[J]. 福州大学学报:自然科学版, 2020, 48(3):325-332.
- [7] 谢锐强, 张惠珍. 求解置换流水车间调度的离散狼群算法[J]. 控制工程, 2020, 27(2):288-296.
- [8] 罗函明, 罗天洪, 吴晓东, 等. 求解混合流水车间调度问题的离散布谷鸟算法[J]. 计算机工程与应用, 2020, 56(22):264-271.
- [9] 王朝, 潘顺发, 方博, 等. 基于改进蛙跳算法的离散制造车间调度模型研究[J]. 数字制造科学, 2020, 18(2):107-112.
- [10] 杜利珍, 王震, 柯善富, 等. 混合流水车间调度问题的果蝇优化算法求解[J]. 中国机械工程, 2019, 30(12):1480-1485.
- [11] 蔡芸, 邓勇, 张波, 等. 带多处理器混合流水车间调度问题的混合鱼群算法[J]. 机械设计与制造, 2017(7):22-25.
- [12] 王凌, 周刚, 许烨, 等. 求解不相关并行机混合流水线调度问题的人工蜂群算法[J]. 控制理论与应用, 2012, 29(12):1551-1557.
- [13] XUE Jiankai, SHEN Bo. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1):22-34.
- [14] 景会成, 王颖. 模拟退火算法优化 PSO-GA 算法解决柔性流水车间调度问题[J]. 小型微型计算机系统, 2020, 41(5):996-999.
- [15] POPPLEWELL K. "Manufacturing systems analysis: with application to production scheduling" by MICHEL BAUDIN[J]. International Journal of Computer Integrated Manufacturing, 1990, 3(6):380.