第 41 卷 第 5 期
2019 年 5 月

系统工程与电子技术
Systems Engineering and Electronics

Vol. 41 No. 5
May 2019

文章编号：1001-506X(2019)05-1079-08

网址：www. sys－ele. com

求解混合流水车间调度的改进贪婪遗传算法

宋存利
（大连交通大学软件学院，辽宁 大连 116028）

摘 要：针对最小化最大完工时间的带有不相关并行机的混合流水车间调度问题，提出了改进贪婪遗传算法。首先，该算法染色体编码采用基于工件加工顺序的编码，解码提出了两种设备分配方案，并考虑到不同阶段加工设备配置不同对算法的影响，采用了正序解码和逆序解码加再调度并用的解码策略。其次，提出贪婪交叉算子和贪婪变异算子，这些算子不仅承担改进种群，增加种群多样性的功能，同时还具有较强的局部搜索能力。最后通过正交实验确定算法的参数设置，与已有算法对已知案例的求解结果进行了比较，说明了该算法的有效性。同时实验表明了正序和逆序解码策略的必要性以及正序或逆序解码的时机。

关键词：混合流水车间调度；贪婪遗传算法；正序和逆序解码；最小化最大完工时间

中图分类号：TP 36　　**文献标志码**：A　　**DOI**：10. 3969/j. issn. 1001－506X. 2019. 05. 21

Improved greedy genetic algorithm for solving the hybrid flow－shop scheduling problem

SONG Cunli
(*College of Software*, *Dalian Jiaotong University*, *Dalian 116028*, *China*)

Abstract: The greedy genetic algorithm is proposed to solve the hybrid flow shop scheduling problem with unrelated parallel machines for minimizing makespan. Firstly, the chromosome coding is based on the job processing sequence and two kinds of machine allocation schemes are considered. Considering the influence of different stages' equipment configuration on the result of scheduling, the forward or reverse decoding strategy with the corresponding rescheduling method are used. Moreover, the greedy crossover and mutation operators are proposed. They not only improve the quality and diversity of the population, but also have a strong local search ability. Finally, the orthogonal experiment is done to determine the parameters of the algorithm, the computation results on the known cases show that the effectiveness of the algorithm compared with the known algorithms. At the same time, the experiments also show the necessity of the forward and reverse decoding strategy and the timing of using them.

Keywords: hybrid flow－shop scheduling problem; greedy genetic algorithm; forward and reverse decoding; makespan

0 引 言

混合流水车间调度问题(hybrid flow－shop scheduling problem，HFSP)是在传统流水车间调度问题基础上，存在多个阶段的加工设备配置了多台的情况。因此该调度问题就由工件排序和设备分配两个工作组成，是传统流水车间调度与并行机调度的综合。相比流水车间调度问题，其复杂度更高，求解空间更大。同时该问题具有很强的工业背景，广泛存在于机械加工、冶炼、食品加工、化工等行业，因此，对此问题进行研究具有重要的学术价值和意义。传统HFSP按并行机类型可分为3类[1]：①相同并行机HFSP，即每一阶段上同一工件在任一台并行设备上的加工时间相同；②均匀并行机HFSP，即每一阶段上同一工件在任一台并行设备上的加工时间与该台设备的加工速度成反比；③不相关并行机HFSP，即每一阶段上同一工件在任两台并行设备上的加工时间互不相关，而取决于工件与设备的匹配程度。

1988年JND等[2]就两阶段的HFSP问题进行了分析，指出该问题是NP难题。针对此，国内外学者提出了很多解决该问题的方法，其中Azizoglu等[3]提出了求解HFSP

收稿日期：2018－05－10；修回日期：2018－11－23；网络优先出版日期：2019－03－18。
网络优先出版地址：http://kns. cnki. net/kcms/detail/11. 2422. TN. 20190318. 0931. 004. html
基金项目：辽宁省自然科学基金(201602130，20170540141，20170540125)；辽宁省教育厅项目(JDL2017017)资助课题

问题的分支定界方法来求解最小化总流水时间；Kis 等[4]提出了下界确定方法和分支定界求解过程，并证明该分支定界方法比 Azizoglu 等提出的方法要快。以上算法属于精确算法，该类算法的最大特点就是在问题规模较小时，能够在合理的时间内给出问题的最优解，但当问题规模变大，则这类算法的求解时间往往不能容忍。启发式算法以某种经验或启发式规则指导算法，往往能在较短的时间内求出问题的近优解而受到推崇。例如屈国强[5]提出了基于瓶颈指向的启发式算法，充分利用各设备负荷一般不相等的特点，瓶颈阶段前加工时间较短而之后加工时间相对较长的工件优先开始加工的启发式知识指导算法在较短的时间内求出了较好的近优解，并与 Nawaz 等[6]提出的 NEH 算法在各种情况下进行了比较，两者各有所长。Koulamas 等[7]提出首先使用 Johnson 算法获得工件排序，然后使相邻工件在下一阶段互相交换得到工件的非排列排序。Ruiz 等[8]对求解 HFSP 问题的方法进行了总结回顾，指出启发式方法相对精确算法来说，它以一定的启发式规则为指导，算法可在相对较短的时间内寻得近优解。但启发式算法的缺点是启发式规则的选取对算法寻优能力有很大影响。元启发式算法是启发式算法的改进，是随机算法与局部搜索算法相结合的产物，它的随机机制使得算法有更高的概率收敛到最优解，因此许多学者采用元启发式算法求解 HFSP。例如：Alaykyran 等[9]提出求解该问题的蚂蚁算法；Low 等[10]提出了求解不相关 HFSP 问题的模拟退火算法；Shiau 等[11]提出了两阶段编码的粒子群算法来求具有带权完工时间的 HFSP 算法；Komaki 等[12]提出了人工免疫算法求解带有批处理机的两阶段混合流水车间调度问题。王凌等[13]提出了求解不相关并行解的人工蜂群算法。王圣尧等[1]、张凤超[14]、刘芳等[15]分别提出了解决不相关并行解的 EDA 算法。Xu 等[16]提出了解决该问题的 DE 算法。采用遗传算法的文献有：苏志雄[17]提出了求解相同并行机的遗传算法，虽然他使用了正逆序的解码，但是采用的正逆序混合机制，不利于算法收敛，同时也没有给出正逆序解码的时机。Figielska[18]等将遗传算法与模拟退火算法相混合来求解带有交货期约束的 HFSP 问题。崔建双等[19]提出了求解不相关并行机的遗传算法。Dai 等[20]提出了求解具有相同并行机的混合流水车间自我导向遗传算法。文献[17－20]遗传算法的特点是将遗传算法与局部搜索算法混合，来提高遗传算法的局部搜索能力，从而提高算法性能。

考虑到遗传算法具有内在的隐并行性和鲁棒性，是求解组合优化问题的常用算法。其概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，从而保证算法向最优解收敛。针对遗传算法强的全局搜索能力，较弱的局部搜索能力，充分挖掘遗传算法自身的局部寻优能力，提出一种贪婪遗传算法(greedy genetic algorithm，GGA)，该算法的贪婪表现在两个方面。①突破传统交叉操作一对双亲只产生两个孩子的惯例，选定进行交叉操作的一对双亲可采用多种交叉操作，产生多个孩子，在多个孩子和父亲中，只有适应值最优的两个个体才能进入下一代，从而确保父代较优基因的遗传，同时增强了交叉算子局部搜索的能力；②进行变异操作的个体可进行多种变异操作，当适应值最优的变异体的适应值比当前染色体较优时，该变异体替换当前染色体进入下一代。这样变异操作不仅承担增加种群多样性的责任，同时具有了较强的局部寻优特性。鉴于设备布局对调度结果的影响，提出了两种设备分配方案和正向或逆向的解码策略，从而增加找到最优解的概率。最后实验证明了所提算法的有效性。

1 问题描述及建模

不相关并行机 HFSP 可描述为 N 个工件在具有 S 个加工阶段的流水线上进行加工，各阶段至少有一台设备且至少有一个阶段存在并行设备，同一阶段上各设备加工同一工件的加工时间不同。在每一阶段各工件均要完成一道工序，但可在相应阶段的任意一台设备上加工。已知工件各道工序在各设备上的处理时间，要求确定所有工件的加工排序 π 以及每一阶段上设备的分配情况，使得加工任务的最大完工时间最小。为了描述问题方便，用到的数学符号定义如下：

N：待加工工件总数；

M：总的加工设备数；

S：加工阶段数，也称工件的加工工序总数；

i：工件编号，值为 $i=1,2,\cdots,N$；

j：工序编号，与加工阶段对应，$j=1,2,\cdots,S$；

m_j：第 j 阶段的加工设备总数；

k：每个阶段的设备编号，$k=1,2,\cdots,m_j$；

$t_{i,j,k}$：第 i 个工件的第 j 道工序在第 k 台设备上的加工时间；

$x_{i,j,k}$：第 i 个工件的第 j 道工序的设备分配情况，若分配到了第 k 台设备上，则其值为 1，否则为 0；

$p_{i,j}$：第 i 个工件的第 j 道工序的加工时间；

$b_{i,j}$：第 i 个工件的第 j 道工序在相应设备上的开始加工时间；

$e_{i,j}$：工件 J_i 的第 j 道工序在相应设备上的加工完成时间；

c_i：第 i 个工件的加工完成时间；

C：所有工件的最大完成时间；

w_j：第 j 阶段的设备平均负荷。

在 HFSP 问题中，通常假设工件的某道工序在某台设备上的加工一旦开始便不可中断，直到该工序加工完成；一台设备同一时刻只能加工一个工件；一个工件同一时刻只能在一台设备上加工；工件可在每阶段的任意一台设备上加工；不同阶段间有无限缓冲。问题目标是求解该问题的最小化最大完工时间，即

$$C=\min\{\max\{c_1,c_2,\cdots,c_n\}\} \tag{1}$$

约束条件为

$$0\leqslant b_{i,j}<e_{i,j} \tag{2}$$

$$e_{i,j-1} \leqslant b_{i,j} \tag{3}$$

$$c_i = e_{i,s} \tag{4}$$

$$\sum_{k=1}^{m_j} x_{i,j,k} = 1 \tag{5}$$

$$p_{i,j} = \sum_{k=1}^{m_j} x_{i,j,k} \cdot t_{i,j,k} \tag{6}$$

$$e_{i,j} = b_{i,j} + p_{i,j} \tag{7}$$

$$w_j = \left(\sum_{i=1}^{n} \sum_{k=1}^{m_j} t_{i,j,k} / m_j \right) / m_j \tag{8}$$

约束说明：式(2)说明工件的第 j 道工序的开工时间必须小于其相应工序的完工时间，且所有工件的工序的开工时间大于等于 0；式(3)说明工件的前一道工序必须完工之后方能加工后一道工序；式(4)说明工件的完工时间等于此工件的最后一道工序的完工时间；式(5)说明每个工件在每个阶段只能安排到一台设备上进行加工；式(6)表示工件的每道工序加工时间的确定方法；式(7)表示工件 J_i 的第 j 道工序完工时间计算公式。式(8)表示加工阶段平均加工负荷的公式，是一种利用所有工件在设备上的平均加工工时的总和除以设备数量的粗略计算方法。

2　贪婪遗传算法

2.1　编码

目前大多数求解 HFSP 的算法都采用的是基于工件排列的编码方法，即用 N 个工件排成顺序队列，工件在队列中的位置标明工件第一道工序的加工顺序。鉴于后续阶段工件的加工受上一阶段影响较大，也采用此方法，后续阶段则根据前一阶段工件加工完成时间采用非降序排列的方式进行编码。举例说明，例如 5 个工件的编码为(4，2，5，1，3)，此编码标明在第一加工阶段，工件 4 先加工，紧接着工件 2 加工、其次是 5、再是 1、最后是工件 3 加工。若它们在第一阶段加工完成到达第二阶段的顺序依次为 2、5、4、1、3，则在第二阶段的编码为(2，5，4，1，3)，后续阶段的编码同理。

2.2　解码方案

基于工件排列的编码只是确定了工件在每个阶段的加工顺序，但没有确定将哪台设备分配给相应工件进行加工。因此在解码过程中，重点是设备的分配，分配方案不同，解码结果就不同。因此提出两种解码方案，并在实验阶段对两种方案进行实验比较。

设备分配算法 1 采用与文献[1]相同的算法，即先完成先分配(first end first allocation，FF)优先分配方案，也就是在相应的加工阶段，哪一台设备能优先完成相关工件的加工就将哪台设备分配给相关工件。此算法的特点不一定能够保证将加工时间最短的设备分配给相关工件，但能将最早完成该工件加工的设备分配给该工件。

设备分配算法 2 采用轮盘赌(random allocation，RA)策略的随机分配算法，即首先根据式(9)计算工件分配到每台设备的概率，加工时间越短的设备分配到的概率越高。此算法能够以较高概率将加工时间较短的设备分配给相关工件，但不一定将完成时间最早的设备分配给相关工件。

$$r_{i,j,k} = (1/t_{i,j,k}) / \sum_{k=1}^{m_j} (1/t_{i,j,k}) \tag{9}$$

2.3　正逆序解码

定义 1　正序解码。就是按照工件的正常工序顺序对染色体进行解码的过程。

定义 2　逆序解码。就是按照工件的正常工序顺序的相反顺序对染色体进行解码的过程。

定义 3　再调度。就是在逆序解码结果的基础上，首先保持此调度方案设备指派方案不变；再次将工件在设备上的加工顺序进行逆序，即假如原来逆序解码时在设备 m_k 上的工件加工顺序若为 1，2，3，则再调度时在设备 m_k 上的加工工件仍为 1，2，3，但再调度后的加工顺序为 3，2，1。然后采用工件的正常工序顺序安排生产，每个工件的对应工序的开工时间、完工时间及其加工时间的对应关系为

$$b_{i,s-j+1} = \overleftarrow{C} - \overleftarrow{e}_{i,j} \tag{10}$$

$$e_{i,s-j+1} = \overleftarrow{C} - \overleftarrow{b}_{i,j} \tag{11}$$

$$p_{i,s-j+1} = \overleftarrow{p}_{i,j} \tag{12}$$

式中，$\overleftarrow{C}$ 为逆序调度方案的完工时间；$\overleftarrow{b}_{i,j}$ 为第 i 个工件的第 j 道工序(即逆序中的第 j 道工序)的开工时间；$\overleftarrow{e}_{i,j}$ 为完工时间；$\overleftarrow{p}_{i,j}$ 为加工时间；则再调度时，第 i 个工件的第 $s-j+1$ 道工序与该工件的逆序工序的第 j 道工序对应。

定理 1　按照定义 3 的再调度方法得到的调度方案是一个合理的调度方案，且任务的完工时间与逆序调度方案的完工时间相同。

证明　在逆序调度方案再调度之后，对任意的第 i 个工件，其第 j 道工序的开始加工时间根据式(10)可计算为 $b_{i,j} = \overleftarrow{C} - \overleftarrow{e}_{i,s-j+1}$，其第 j 道工序的完工时间根据式(11)为 $e_{i,j} = \overleftarrow{C} - \overleftarrow{b}_{i,s-j+1}$。而根据式(7)，逆序调度时，$\overleftarrow{e}_{i,s-j+1} = \overleftarrow{b}_{i,s=j+1} + \overleftarrow{p}_{i,s-j+1}$，所以能得出结论

$$e_{i,j} - b_{i,j} = (\overleftarrow{C} - \overleftarrow{b}_{i,s-j+1}) - (\overleftarrow{C} - \overleftarrow{e}_{i,s-j+1}) = \overleftarrow{e}_{i,s-j+1} - \overleftarrow{b}_{i,s-j+1} = \overleftarrow{p}_{i,s-j+1} = p_{i,j}$$

满足式(7)且与式(12)一致。根据式(10)，有 $b_{i,1} = \overleftarrow{C} - \overleftarrow{e}_{i,s}$；根据式(1)，有 $\overleftarrow{e}_{i,s} \leqslant \overleftarrow{C}$，因此得出 $b_{i,1} \geqslant 0$，满足条件(2)；根据式(11)，有 $e_{i,j-1} = \overleftarrow{C} - \overleftarrow{b}_{i,s-j+2}$，而 $b_{i,j} = \overleftarrow{C} - \overleftarrow{e}_{i,s-j+1}$；根据式(3)知 $\overleftarrow{e}_{i,s-j+1} \leqslant \overleftarrow{b}_{i,s-j+2}$，所以 $e_{i,j-1} \leqslant b_{i,j}$，即再调度的结果满足式(3)；同理可证明再调度的结果也满足式(4)～式(5)。因此本调度是一个合理的调度，且其 $C = \overleftarrow{C}$。　　证毕

2.4　交叉操作

贪婪交叉策略的思想受自然界中一个家庭的父母可生多个孩子，而优秀的孩子往往是家庭的支柱，影响着整个家庭的发展这一特点启发。对进行交叉操作的两个父亲，选择多种不同交叉操作产生多个孩子；从孩子和父亲中选择适应值最好的两个个体进入下一代。从而提高次操作的全局和局部搜索能力。采用 3 种交叉操作，具体描述如下：

2.4.1 位置交叉算子

随机产生 3～5 个 1 到 n 之间不相等的随机整数，将父亲 $P1$ 中的相应位置的基因写入到孩子 $C2$ 上对应位置，将父亲 $P2$ 中去掉和孩子 $C2$ 相同的基因，然后将父亲 $P2$ 上剩余的基因按照顺序写入到孩子 $C2$ 的空白位置。同理生成孩子 $C1$。举例如下：设父亲 $P1(6,5,8,2,4,1,7,3)$，$P2(3,8,2,5,1,6,47)$，假如随机产生的 3 个不相同的随机数为 2，5，7。具体交叉操作过程如下：

步骤 1 将父亲 $P1$ 基因位置 2，5 和 7 上的基因写入到孩子 $C2$ 的对应基因位上中，同理父亲 $P2$ 写入到孩子 $C1$ 中，X 代表空白，如图 1 所示。

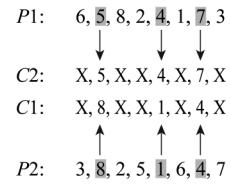

图 1 位置交叉算子第 1 步

Fig. 1 1st step of position based crossover

步骤 2 将父亲 $P1$ 中不在孩子 $C1$ 中的基因按照顺序写入到孩子 $C1$ 的空白基因位上，产生孩子染色体，同理生成孩子 $C2$，如图 2 所示。

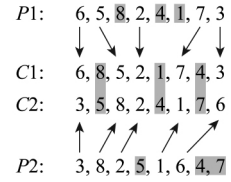

图 2 交叉操作第 2 步

Fig. 2 2nd step of position based crossover

2.4.2 转接交换交叉算子

随机产生几个不连续的整数，在父亲 $P1$ 中找到这些位置的基因，将父亲 $P2$ 中不包含的父亲 $P1$ 中选定的基因中的其余基因写入到孩子 $C1$ 的对应基因位置，然后将父亲 $P1$ 中的选定位置的基因按照顺序插入到 $C1$ 的空白位置。同理产生孩子 $C2$。具体过程如下：

步骤 1 假如随机生成的 3 个整数为 2、5 和 7，则将父亲 $P2$ 中不在这些位置的基因写入到 $C1$ 中，如图 3 所示。

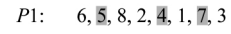

图 3 转接交换交叉第 1 步

Fig. 3 1st step of subtour exchange crossover

步骤 2 先在父代 $P2$ 中找到父代 $P1$ 被选中基因的位置，再用父代 $P2$ 中其余的基因生成子代 $C1$，并保证位置对应，如图 4 所示。

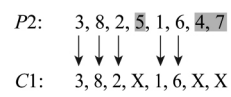

图 4 转接交换交叉第 2 步

Fig. 4 2nd step of subtour exchange crossover

步骤 3 将父代 $P1$ 中被选择的基因按顺序放入子代 $C1$ 的剩余位置中，如图 5 所示。

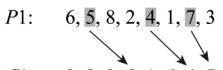

图 5 转接交换交叉第 3 步

Fig. 5 3rd step of subtour exchange crossover

同理，产生孩子 $C2$ 染色体。

2.4.3 顺序交叉算子

随机产生 2 个不等的 1 到 n 之间的整数 $h1$ 和 $h2$，假如较小的数存放在 $h1$ 中，将父亲 $P1$ 区间 $h1$ 到 $h2$ 上的基因写入到孩子 $C2$ 的对应区间上，父亲 $P2$ 对应区间的基因写入到孩子 1 中，将父亲 $P2$ 中去掉孩子 $C2$ 上有的基因，然后按顺序写入到孩子 $C2$ 的空白的位置，孩子 $C2$ 形成。同理对孩子进行如此操作，形成孩子 $C1$。以图 3 中两个父染色体为例，生成随机数 4 和 6，则具体交叉操作流程如下：

步骤 1 将父亲 $P1$ 基因位置 4 到 6 的基因写入到孩子 $C2$ 的对应基因位中，同理将父亲 $P2$ 写入到孩子 $C1$ 中，X 代表空白，如图 6 所示。

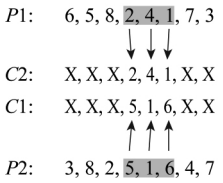

图 6 顺序交叉第 1 步

Fig. 6 1st step of order crossover

步骤 2 将父亲 $P1$ 中不在孩子 $C1$ 中的基因按照顺序写入到孩子 $C1$ 的空白基因位上，产生孩子染色体，同理生成孩子 $C2$，如图 7 所示。

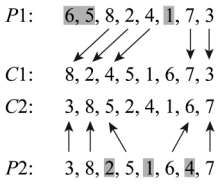

图 7 顺序交叉第 2 步

Fig. 7 2nd step of order crossover

2.5 变异操作

贪婪变异操作思想为：对一个选定变异的染色体，采用 3 种变异操作进行变异产生 3 个变异染色体，分别计算 3 个变异染色体的适应值，选出适应值最好的变异染色体，将其与原染色体的适应值进行比较，若其适应值优于原染色体，则替换原有染色体进入下一代，否则原染色体保持不变。鉴于变异操作不仅能增加种群多样性，而且能提高算法局部搜索能力，因此可设置较高的变异概率。同时为了确保

其增加种群多样性的性质不变，算法增加了一个最低替代率替代概率(probability of substitution，PS)，即在一次变异操作中，最少应确保 PS 个染色体被其变异染色体替换。

具体的变异操作如下：

(1)变异操作 1

随机产生 1 到 n 之间的两个随机整数，交换此位置的两个基因，则此变异操作完成。

(2)变异操作 2

随机产生 1 到 n 之间的两个随机整数，对两随机数之间的基因进行逆序，则形成变异后的染色体。

(3)变异操作 3

随机产生 1 到 n 之间的两个随机整数，对两随机数之间的基因进行随机排序，则形成变异后的染色体。

2.6　GGA 算法框架

GGA 仍标准遗传算法流程，初始种群采用随机生成，同时采用了最优保留策略。

3　仿真实验及分析

实验算法采用 VC＋＋6.0 编写，计算机配置为：操作系统采用 WIN7.0，内存 2 GB，主频 2.66 GHz。

3.1　基于正序解码的 GGA 的参数校验

考虑到正序逆序解码受到问题数据的影响较大，而算法参数的配置对它们的影响一致，因此采用 GGA－FF 算法来做实验。实验采用 $L_{16}(3^4)$ 正交实验法确定参数，每个参数考察 4 个水平，种群规模 Psize 考察 30、40、50 和 60；交叉概率(probability of crossover，PC)考察 0.6、0.75、0.85 和 1；变异概率(probability of mutation，PM)考察 0.06、0.5、0.85 和 1；PS 固定设为 0.085。鉴于小规模问题对算法参数要求不高，因此实验数据以 20 个工件 5 个阶段的较大规模问题为准。工件的加工时间在 5～20 个单位时间区间内，每阶段并行设备数在 2～3 台之间，具体实验数据如表 1 所示。

表 1　20 个工件的加工时间表

Table 1　Process time table of 20 jobs

阶段	设备	工件																		
		J1	J2	J3	J4	J5	J6	J7	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20
S1	M1	10	13	17	7	15	16	19	9	16	17	5	15	6	9	11	18	17	7	13
	M2	8	8	13	6	19	8	10	16	19	18	16	18	15	10	18	19	17	9	10
	M3	14	7	5	6	6	14	15	19	19	16	14	11	19	18	8	17	13	18	6
S2	M4	14	19	7	7	16	8	13	9	13	10	6	14	12	11	15	16	13	15	15
	M5	11	16	15	6	15	12	18	15	9	9	19	5	9	6	5	12	14	7	13
	M6	10	10	7	11	15	9	6	15	10	13	17	16	9	13	11	14	17	14	18
S3	M7	12	9	7	8	9	19	7	12	13	12	7	6	7	10	5	11	17	14	12
	M8	16	6	8	15	8	7	16	12	16	14	6	14	17	16	11	15	8	11	9
S4	M9	15	13	18	12	16	16	10	15	13	9	8	16	13	18	16	12	10	13	18
	M10	14	7	14	14	19	6	16	9	14	15	16	11	5	14	18	10	10	8	19
	M11	12	17	11	8	17	16	19	10	12	19	15	16	9	14	14	11	11	10	10
S5	M12	11	10	8	14	6	14	18	19	11	11	7	16	18	10	17	12	14	12	16
	M13	11	6	10	13	17	15	13	19	11	16	11	16	18	9	10	8	19	6	7
	M14	10	13	10	14	13	18	7	13	12	16	12	12	7	11	10	12	12	12	10

算法最大迭代 1 000 次，每种参数组合算法独立运行 10 次，取 10 次的平均值，实验结果数据如表 2 所示。

表 2　正交实验结果

Table 2　Orthogonal experiment results

参数组合序号	Psize	PM	PC	平均值
1	1	1	4	126.2
2	2	1	3	124.7
3	3	1	2	124.8
4	4	1	1	125.6
5	1	2	3	124.2
6	2	2	2	125
7	3	2	1	124.2
8	4	2	4	123.3
9	1	3	2	123.3
10	2	3	1	123.3
11	3	3	4	122.9
12	4	3	3	122.9
13	1	4	1	124.8
14	2	4	4	122.8
15	3	4	3	122.3
16	4	4	2	121.9

表 2 的正交实验表明，参数组合采用[4，4，2]是较为稳定的参数组合。即参数 Psize＝60，PM＝1，PC＝0.75。GGA 各参数的极差如表 3 所示。

表 3　GGA 各参数的极差

Table 3　Range of parameters of GGA

水平	Psize	PM	PC
1	124.7	125.1	122.9
2	123.6	122.8	122.1
3	121.9	122.7	122
4	122.1	122.1	122.7
极差	2.8	3.0	0.9
等级	2	1	3

表 3 分析表明变异操作对算法性能影响最大，其次是种群规模，相对而言，交叉操作的影响最小。

3.2　两种设备分配方案的校验

为了验证哪种设备分配方案较好，实验仍以表 1 的实验案例为准，参数采用第 3.1 节实验得出的参数最佳组合 Psize＝60，PM＝1，PC＝0.75，每一种方案运行 10 次，统计 10 次的最优值和平均值，结果如表 4 所示。

表 4 两种设备分配方案实验结果

Table 4 Experiment results with two machine allocation strategies

设备分配策略	次数										最优值	平均值
	1	2	3	4	5	6	7	8	9	10		
GGA－FF	117	121	125	123	121	122	124	120	123	123	117	121.9
GGA－RA	125	125	129	125	127	128	125	127	126	125	125	126.2

从表 4 的统计结果可看出，FF 的设备分配策略比 RA 的随机设备分配策略从最优值和平均值方面均要好。因此可得出结论，设备分配方案策略 FF 更好。

3.3 已知算例测试

首先利用文献[1]、文献[14]和文献[15]中提出的 4 个 HFSP 实例，分别称为 $L1$、$L2$、$L3$ 和 $L4$，并与相应文献的结果进行对比，本文算法参数仍采用第 3.1 节验证参数，Psize＝60，PM＝1，PC＝0.75，算法最大迭代代数 1 000，算法运行 10 次。相应结果如表 5 所示。

表 5 已知案例测试结果

Table 5 Experiment results on known test case

次数	EDA[1]		EDA[14]	EDA－I[15]			GA[19]		ABC[13]		GGA－RA				GGA－FF			
	$L1$	$L2$	$L3$	$L2$	$L3$	$L4$	$L1$	$L2$	$L1$	$L2$	$L1$	$L2$	$L3$	$L4$	$L1$	$L2$	$L3$	$L4$
1	23	297	14	297	13	21	30	347	23	297	23	297	13.5	22	23	297	13.5	21
2	24	297	14	297	13	22	27	—	23	297	23	297	14	22	23	297	13.5	21
3	23	297	15	297	13	21	26	—	23	297	24	297	13.5	22	23	297	13.5	21
4	23	297	14	297	13.5	21	27	—	24	297	23	298	14	21	23	297	13.5	21
5	23	298	14	297	13	21	20	—	23	297	23	297	14	22	23	297	13.5	21
6	23	297	14.5	297	13	21	27	—	23	297	24	298	13.5	22	23	297	13.5	21
7	24	297	14	298	13	21	26	—	23	297	24	298	14	21	23	297	13.5	21
8	24	298	14.5	297	13.5	21	27	—	24	297	23	297	14	22	23	297	13.5	21
9	23	298	14	297	13	22	26	—	23	297	23	298	14	22	23	297	13.5	21
10	24	298	14	297	13	21	28	—	23	297	24	298	13.5	22	23	297	13.5	21

从表 5 可看出，案例 $L1$ 的最优解为 23，且 GGA－FF 是 100％寻优，文献[1]、文献[13]和 GGA－RA 的最佳寻优结果也为 23，但寻优概率均不及 GGA－FF 好，文献[19]最差，没有找到最优解。案例 $L2$ 的最优结果为 297，GGA－FF 和文献[13]都是 100％找到，其他文献均不理想。针对案例 $L3$，GGA－FF 和 GGA－RA 找到的最优解均为 13.5，比文献[14]的寻优结果要好，而文献[15]的最优解为 13，但可证明 13.5 为全局最优解，不可能达到 13，除非数据有误。研究发现文献[15]错把案例 $L1$ 的实验数据错误使用，也就是提出的案例 $L4$，经实验比对，GGA－FF 和 GGA－RA 均寻得最优完工时间为 21 的较好解，且 GGA－FF 为 100％寻得，比文献[15]的 80％要好。综上，GGA－FF 算法稳定性较好，并且寻优概率相比已知文献要高，而 GGA－RA 在 4 个案例上的 10 次运行均找到了和 GGA－FF 一样的优化解，但相比 GGA－FF 其寻优概率较低，究其原因，和其概率化的设备分配方案不无关系。图 8 为 $L3$ 案例的优化解的甘特图，其工件加工数据可参考文献[14]。

3.4 较大规模测试案例实验

实验的目的是验证正序逆序解码策略的必要性。实验数据以 20 个工件 5 个阶段、30 个工件 5 个阶段的较大规模问题为准，工件的加工时间在 5～20 个单位时间之间，每阶段并行设备数在2～3台之间，具体实验数据采用随机算法生成。案例命名采用 J20c5a＊，J 代表工件，20 代表工件数，c 代表阶段，5 代表阶段数，a＊代表本规模的第几个问题。例如 J30c5a2 代表工件数为 30，阶段数为 5 个的第 2 个案例。针对每个案例每个算法均运行 10 次，每次算法最大迭代 1 000 次，统计其最优值和平均值。由于第 3.2 节、第 3.3 节实验已经证明了基于 FF 的设备分配策略比 RA 要好，因此，实验采用 FF 验证正序及逆序解码的必要性。算法分别为 GGA－FF 和 GGA－FF’，其中 GGA－FF 为采用正序解码的 GGA，GGA－FF’则为采用逆序解码的 GGA 算法，算法参数采用 Psize＝60，PM＝1，PC＝0.75，具体实验结果如表 6 所示。最大瓶颈阶段为 w_j 最大的阶段。

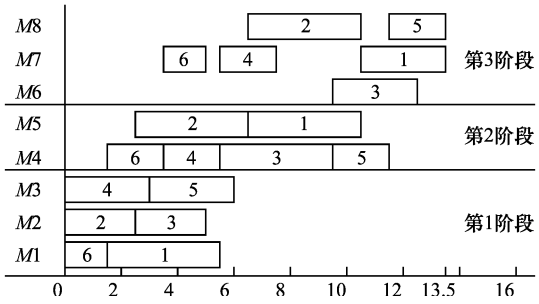

图 8 案例 $L3$ 的甘特图

Fig. 8 Gantt chart on test case $L3$

表 6　较大规模 HFSP 的实验结果

Table 6　Experiment results on large scale HFSP

案例	设备配备情况	瓶颈阶段	GGA－FF		GGA－FF'	
			最优值	平均值	最优值	平均值
J20c5a1	[3,3,2,2,2]	5	117	121.9	121	123
J20c5a2	[3,3,2,3,3]	3	119	119.9	118	120
J20c5a3	[3,2,3,3,2]	5	119	121.2	119	120.1
J20c5a4	[2,2,3,3,2]	1	138	142.8	141	143.1
J20c5a5	[3,2,2,3,2]	5	141	141.3	140	141
J30c5a1	[3,2,2,3,3]	2	180	184.9	182	185.6
J30c5a2	[3,3,3,3,2]	5	169	169.9	167	169.2
J30c5a3	[2,2,2,2,3]	3	185	188.8	188	191.7
J30c5a4	[3,3,2,2,2]	4	176	179	175	177.3
J30c5a5	[2,3,2,2,2]	3	188	191	190	191.9

从表 6 的实验结果可看出，在 10 个案例中，有 4 个案例即 J20c5c2、J20c5a4、J30c5a2 和 J30c5a4 采用逆序解码时求得的 Cmax 和 AVG 均比采用正序解码的结果要好。有 1 个案例即 J20c5a3 两种解码算法均找到了同样的最优解 Cmax，但逆序解码的 AVG 要比正序好，说明逆序解码时求得此案例的最好解的概率更高。剩余 5 个案例正序解码均比逆序要好。通过此实验得出结论逆序解码是正序解码的必要补充，究其原因，这和设备的配置有一定关系。在表 6 的 10 个案例中，不难发现逆序解码较好的 4 个案例中，瓶颈阶段多在靠后的阶段上（也就是阶段 4 或阶段 5 上），而正序解码较好的 5 个案例中，其中 2 个案例瓶颈阶段靠前，在阶段 1 或阶段 2 上，2 个案例的瓶颈阶段在阶段 3 上，只有案例 J20c5a1 的瓶颈阶段在 5 上，从而可得出结论在瓶颈阶段在较前面的加工阶段时，采用正序解码较好，而瓶颈阶段在加工流程的较后面阶段时，采用逆序解码效果更好一些。

4　结　论

研究了最小化最大完工时间的带有不相关并行机的 HFSP，针对该问题的并行设备的配置对调度结果有很大的影响，提出一种正序或逆序的解码策略，同时提出了 GGA，该算法的交叉算子和变异算子不仅承担传统遗传算法相关特性，而且交叉和变异均采用贪婪策略，即只有优秀的孩子染色体和变异染色体才能进行到下一代种群中，否则将会被淘汰掉，这大大加强了算法的寻优能力和收敛速度。通过正交实验验证了变异算子对算法的影响最大，当变异概率设置为 1 时，算法具有较好的性能。这说明了贪婪策略对变异算法的影响较大，且增加了算法的局部寻优的能力。同时，针对工件排序的染色体编码，采用两种设备分配策略，实验验证了两种策略的优劣。最后，对较大规模 HFSP，实验验证了正序和逆序策略的正确性。未来针对该问题的研究方向是基于工件加工顺序的设备调度规则的进一步研究。

参考文献：

[1] 王圣尧，王凌，许烨，等. 求解混合流水车间调度问题的分布估计算法[J]. 自动化学报，2012，38(3)：437－443.
WANG S Y，WANG L，XU Y，et al. An estimation of distribution algorithm for solving hybrid flow－shop scheduling problem[J]. Acta Automatica Sinica，2012，38(3)：437－443.

[2] JND G. Two－stage，hybrid flowshop scheduling problem[J]. Journal of the Operational Research Society，1988，39(4)：359－364.

[3] AZIZOGLU M，ÇAKMAK E，KONDAKCI S. A flexible flow-shop problem with total flow time minimization[J]. European Journal of Operational Research，2001，132(3)：528－538.

[4] KIS L，PESCH E. A review of exact solution methods for the non－preemptive multi processor flowshop problem[J]. European Journal of Operational Research，2005，164(3)：592－608.

[5] 屈国强. 瓶颈指向的启发式算法求解混合流水车间调度问题[J]. 信息与控制，2012，41(4)：514－521.
QU G Q. Bottleneck focused heuristic algorithm for hybrid flow shop scheduling problem[J]. Information and Control，2012，41(4)：514－521.

[6] NAWAZ M，ENSCORE E E JR，HAM I. A heuristic algorithm for the m－machine，n－job flow－shop sequencing problem[J]. Omega，1983，11(1)：91－95.

[7] KOULAMAS C. A new constructive heuristic for the flow shop scheduling problem[J]. European Journal of Operational Research. 1998，105(1)：66－71.

[8] RUIZ R，VÁZQUEZ－RODRÍGUEZ J A. The hybrid flow shop scheduling problem[J]. European Journal of Operational Research，2010，205(1)：1－18.

[9] ALAYKYRAN K，ENGIN O，DÖYEN A. Using ant colony optimization to solve hybrid flow shop scheduling problems[J]. The International Journal of Advanced Manufacturing Technology，2007，35(5/6)：541－550.

[10] LOW C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines[J]. Computers and Operations Research，2005，32(8)：2013－2025.

[11] SHIAU D F，HUANG Y M. A hybrid two－phase encoding particle swarm optimization for total weighted completion time minimization in proportionate flexible flow shop scheduling[J]. The International Journal of Advanced Manufacturing Technology，2012，58(1)：339－357.

[12] KOMAKI M，EHSAN T，VAHID K. Minimizing makespan in the two－stage assembly hybrid flow shop scheduling problem using artificial immune systems[J]. International Journal of Production Research，2016，54(4)：963－983.

[13] 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.
WANG L，ZHOU G，XU Y，et al. An artificial bee colony algorithm for solving hybrid flow－shop scheduling problem with unrelated parallel machines[J]. Control Theory & Applications，2012，29(12)：1551－1557.

[14] 张凤超. 改进的分布估计算法求解混合流水车间调度问题研究[J].

软件导刊，2014，13(8)：23－36.
ZHANG F C. An improved estimation of distribution algorithm to solve the hybrid flow shop scheduling problem[J]. Software Guide，2014，13(8)：23－36.

[15] **王芳，唐秋华，饶运清. 求解柔性流水车间调度问题的高效分布估算算法[J]. 自动化学报**，2017，43(2)：280－293
WANG F，TANG Q H，RAO Y Q. Efficient estimation of distribution for flexible hybrid flow shop schedling[J]. Acta Automatica Sinica，2017，43(2)：280－293.

[16] XU Y，WANG L. Differential evolution algorithm for hybrid flowshop scheduling problem[J]. Journal of Systems Engineering and Electronics，2011，22(5)：794－798.

[17] **苏志雄，伊俊敏. 基于正逆序策略的混合流水车间遗传调度算法[J]. 计算机集成制造系统**，2016，22(4)：1059－1069.
SU Z X，YI J M. Genetic algorithm with forward backward scheduling approach for hybrid flow shop problems[J]. Computer Integrated Manufacturing Systems，2016，22(4)：1059－1069.

[18] FIGIELSKA E. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flow shop with additional resources[J]. Computers and Industrial Engineering，2009，56(1)：142－151.

[19] **崔建双，李铁克，张文新. 混合流水车间调度模型及其遗传算法[J]. 北京科技大学学报**，2005，27(5)：623－626.
CUI J S，LI T K，ZHANG W X. Hybrid flowshop scheduling model and its genetic algorithm[J]. Journal of University of Science and Technology Beijing，2005，27(5)：623－626.

[20] DAI W，XIA K. Approach to hybrid flow－shop scheduling problem based on self－guided genetic algorithm[J]. Journal of Advanced Computational Intelligence & Intelligent Informatics，2015，19(3)：365－371.

作者简介：

宋存利(1975－)，**女，副教授，博士，主要研究方向为生产调度、智能优化算法**。
E－mail：scunli@163.com