

# 混合并行机调度问题的多目标优化模型及算法

付亚平, 黄敏<sup>†</sup>, 王洪峰, 王兴伟

(东北大学 信息科学与工程学院 流程工业综合自动化国家重点实验室(东北大学), 辽宁 沈阳 110819)

**摘要:** 针对生产工序的合并造成一种串并联共存的生产布局, 研究了一种特殊的混合并行机调度问题, 并考虑以最小化总流水时间和最小化总延迟工件数量为目标的多目标调度问题, 建立了混合整数规划模型. 针对模型特点, 设计了一种改进的非支配排序遗传算法进行求解, 采用基于启发式方法的初始种群生成方式以提高种群的质量和多样性, 并引入一种局域搜索策略以改善求解算法所获得的非支配解的质量及分布性. 通过对大量数值算例进行仿真实验, 并与典型的多目标优化算法进行比较, 结果表明所提出的模型和算法在收敛性、分布性及极端点质量方面均具有优势, 能够较好的解决多目标混合并行机调度问题.

**关键词:** 混合并行机调度问题; 多目标优化; 非支配排序遗传算法; 局部搜索

**中图分类号:** TP13      **文献标识码:** A

## Multi-objective optimization model and algorithm for hybrid parallel machine scheduling problem

FU Ya-ping, HUANG Min<sup>†</sup>, WANG Hong-feng, WANG Xing-wei

(State Key Laboratory of Synthetical Automation for Process Industries (Northeastern University),

College of Information Science and Engineering, Northeastern University, Shenyang Liaoning 110819, China)

**Abstract:** A special hybrid machine scheduling problem is investigated since the combination of production processes can yield a special layout with the coexistence of serial and parallel production. In order to address this problem effectively, a mixed integer programming multiobjective model, which aims to minimize both the total flow time and the total number of tardy jobs, is built and an improved nondominated sorting genetic algorithm II (NSGA-II), where a heuristic method is proposed to generate and improve the quality and diversity of initial population, and a local search strategy is embedded to improve the quality and distribution of the achieved nondominated solutions, is designed in this paper. Compared with the classical multiobjective optimization algorithms, the experimental results on a number of stochastic examples with different scale show that the proposed model and algorithm have advantage in the convergence, spread and extreme value point, and can address effectively the investigated hybrid parallel machine scheduling problem.

**Key words:** hybrid parallel machine scheduling problem; multiobjective optimization; NSGA-II; local search

## 1 引言(Introduction)

随着现代化生产设备及复合型技术工人的出现, 很多制造型企业的生产线上会出现串并联共存的生产布局. 比如, 在玻璃加工过程中, 传统车间布局将成型和焊接操作设置为两个串联工序: 随着多功能玻璃机床的配备, 这两个工序可以合并为一个工序连续完成, 于是便会出现了一种特殊的串并联两种生产方式并存的布局(如图1所示). 两种操作均需热加工, 工件到达后依次经过加热成型后完成焊接操作, 由于工艺及技术要求, 成型设备需无空闲运行. 因串联方式存在工序不同步问题可能会造成工件在设备2(M2)处的

延迟开工, 使其不能满足焊接温度要求, 需要进行二次加热, 这无疑会大大增加额外的生成成本. 在制品库存是生产成本控制的重要组成部分, 因此, 如何降低在制品库存量和减少二次加热的工件数量是这类企业需要解决的关键问题. 研究这类生产调度的模型与方法, 具有重要的理论意义和应用价值.

并行机调度问题已经得到众多学者的广泛关注与研究<sup>[1]</sup>. 谭圆圆等<sup>[2]</sup>以钢铁生产为背景, 研究了以最小化总流水时间的并行加热炉调度问题. 汪恭书等<sup>[3]</sup>研究了工件具有最迟开始处理时间的并行机实时调度问题. Yang等<sup>[4]</sup>考虑了设备具有速率改变特性的非

收稿日期: 2013-11-24; 录用日期: 2014-09-01.

<sup>†</sup>通信作者. E-mail: mhuang@mail.neu.edu.cn; Tel.: +86 24-83691272.

基金项目: 国家杰出青年科学基金资助项目(71325002, 61225012); 国家自然科学基金资助项目(71071028, 71001018); 高等学校博士学科点专项科研基金优先发展领域资助课题(20120042130003); 高等学校博士学科点专项科研基金资助课题(20110042110024); 中央高校基本科研业务费专项资金资助项目(N110204003, N120104001, N130404017, N110204005); 流程工业综合自动化国家重点实验室基础科研业务费资助项目(2013ZCX11).

等效并行机调度问题. 李作成等<sup>[5]</sup>研究了准备时间依赖工件排序的非等效并行机调度问题. Hsu等<sup>[6]</sup>研究了设备具有维修活动的非等效并行机调度问题. 这些研究均考虑单设备并行的生产布局. Sundararaghavan等<sup>[7]</sup>提出了两条并行的两阶段流水线调度问题. 庞哈利等<sup>[8-9]</sup>研究了多条并行的多阶段流水线问题, 但其均未考虑生产工序合并问题. 从目前的研究情况来看, 尚无针对上述这种混合并行机调度问题(hybrid parallel machine scheduling problem, HPMSP)的直接相关文献. 随着现代化生产设备及新型生产布局的出现, 现有的调度模型和算法已不适用于解决该类问题. 另外, 实际生产环境中往往存在多个优化目标, 研究多目标调度问题更加符合实际, 且已成为调度问题研究的趋势<sup>[10-11]</sup>. 为此, 根据图1所示的生产布局, 本文针对这种特殊的混合并行生产线调度问题展开研究, 建立以最小化总流水时间和最小化设备2处总延迟工件数量目标的混合整数规划模型, 设计求解该多目标模型的优化算法, 并利用不同规模下HPMSP仿真算例来检验所提出模型和算法的有效性.

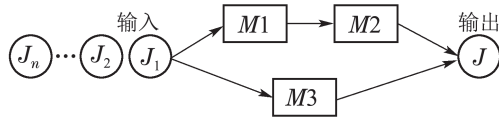


图 1 HPMSP的生产布局

Fig. 1 Production unit of HPMSP

## 2 问题描述与建模(Problem description and model)

HPSMP问题可描述如下: 工件有两种加工生产线, 生产线1分两个工序完成, 且设备1需无空闲运行, 工件在设备2处具有最大等待时间要求, 若等待时间过长造成延迟开工, 将产生额外的二次加热时间: 生产线2由一个工序完成, 不存在上述问题. 工件在同一生产线不同设备上的排序与工件进入该生产线的顺序相同, 每台设备在同一时间只能加工一个工件, 每个工件在同一时间只能在一台设备上加工, 加工过程不允许中断.

问题参数:

$J$ : 表示工件集,  $J = 1, 2, \dots, n$ ;

$J'$ : 表示扩展工件集,  $J' = 0, 1, 2, \dots, n$ ;

$L$ : 表示生产线集,  $1, 2, \dots, k$ ;

$M_l$ : 表示生产线 $l$ 上的机器集,  $1, 2, \dots, m_l$ ;

$j$ : 表示工件编号;  $j \in J$ ;

$i$ : 表示扩展工件编号, 其中 $i = 0$ 表示虚拟工件,  $i \in J'$ ;

$l$ : 表示生产线编号,  $l \in L$ ;

$m$ : 表示机器编号,  $m \in M_l$ ;

$r_j$ : 表示工件 $j$ 的到达时间;

$p_{jlm}$ : 表示工件 $j$ 在生产线 $l$ 上机器 $m$ 处的加工时间;

$w_j^{\max}$ : 表示工件 $j$ 的最大允许等待时间;

$p'$ : 表示工件进行二次加热所需的时间;

$M$ : 表示无穷大的正整数;

$U_{jlm}$ : 表示工件 $j$ 在生产线 $l$ 上机器 $m$ 处因等待时间过长导致二次加热的单位惩罚, 这里对于 $\forall j \in J, l \in L$ , 当 $m = 1$ 时,  $U_{j11} \equiv 0$ , 这表示工件在生产线上第1台机器处不会出现二次加热现象. 当 $m > 1$ 时, 若 $r_{jlm} - C_{(jl(m-1))} > w_j^{\max}$ 成立, 则 $U_{jlm} = 1$ ; 否则 $U_{jlm} = 0$ .

决策变量:

$r_{jlm}$ : 表示工件 $j$ 在生产线 $l$ 上机器 $m$ 处的开始加工时间.

$C_j$ : 表示工件 $j$ 的完工时间.

$C_{jlm}$ : 表示工件 $j$ 在生产线 $l$ 上机器 $m$ 处的完工时间.

$$x_{jl} = \begin{cases} 1, & \text{工件 } j \text{ 安排在生产线 } l \text{ 上加工}, \\ 0, & \text{否则}, \end{cases}$$

$$y_{ijl} = \begin{cases} 1, & \text{工件 } j \text{ 紧跟工件 } i \text{ 进入生产线 } l, \\ 0, & \text{否则}, \end{cases}$$

于是, 以最小化总流水时间和最小化总延迟工件数量为目标函数的HPSMP问题的多目标混合整数规划模型为

$$\min Z1 = \sum_{j=1}^n (C_j - r_j),$$

$$\min Z2 = \sum_{j=1}^n \sum_{l=1}^k \sum_{m=1}^{m_l} U_{jlm},$$

s.t.

$$\sum_{l=1}^k x_{jl} = 1, j = 1, 2, \dots, n, \quad (3)$$

$$\sum_{l=1}^k \sum_{i=0}^n y_{ijl} = 1, j = 1, 2, \dots, n, \quad (4)$$

$$r_{jlm} = \max \left\{ \sum_{i=0}^n (y_{ijl} \cdot C_{ilm}), r_j \right\}, j = 1, 2, \dots, n, \\ l = 1, 2, \dots, k, m = 1, 2, \dots, m_l, \quad (5)$$

$$C_{jlm} = r_{jlm} + p_{jlm} + U_{jlm} \cdot p', j = 1, 2, \dots, n, \\ l = 1, 2, \dots, k, m = 1, 2, \dots, m_l, \quad (6)$$

$$C_{jlm} - C_{ilm} + M \cdot (1 - y_{ijl}) \geq 0, j = 1, 2, \dots, n, \\ l = 1, 2, \dots, k, m = 1, 2, \dots, m_l, \quad (7)$$

$$C_{0lm} = 0, l = 1, 2, \dots, k, m = 1, 2, \dots, m_l, \quad (8)$$

$$C_j \geq x_{jl} \cdot C_{jlm}, j = 1, 2, \dots, n, l = 1, 2, \dots, k, \\ m = 1, 2, \dots, m_l, \quad (9)$$

$$x_{jl}=1 \text{ 或 } 0, j=1, 2, \dots, n, l=1, 2, \dots, k, \quad (10)$$

$$y_{ijl}=1 \text{ 或 } 0, i=1, 2, \dots, n, j=1, 2, \dots, n, \\ l=1, 2, \dots, k, \quad (11)$$

其中: 式(1)表示最小化总流水时间的目标函数, 式(2)表示最小化总延迟工件数量的目标函数; 式(3)表示每一个工件只能在一条生产线加工; 式(4)表示每一工件都有且仅有一个紧前工件; 式(5)–(6)表示工件在各台设备上的开始时间和完成时间; 式(7)保证同一条生产线不同设备上的工件排序与工件进入生产线的顺序相同; 式(8)表示虚拟工件的完成时间; 式(9)表示工件完工时间; 式(10)–(11)为决策变量取值约束。

### 3 算法设计(Algorithm design)

根据文献[12]的结论, 工件具有到达时间的单机, 并行机及流水线调度问题均为NP-Hard问题, 显然HPMSP问题可以被认为是一些问题的结合, 因此也是NP-Hard问题。考虑到这里的模型是一个多目标优化模型, 于是利用一种经典的多目标遗传算法NSGA-II来求解HPMSP问题。NSGA-II是Deb等<sup>[13]</sup>于2002年提出的一种多目标优化算法, 因其具有较快的收敛速度以及良好的Pareto解集分布性而得到了广泛的关注<sup>[14]</sup>。由于NSGA-II最初被应用于实优化问题, 而本文研究的HPMSP问题是一个混合整数规划问题, 接下来将从染色体编码, 初始种群生成以及遗传算子(选择、交叉和变异)等方面给出求解算法的具体设计。此外, 根据问题的特点, 本文还考虑将一种局域搜索策略嵌入到求解算法中, 以进一步改善NSGA-II算法的性能。

#### 3.1 染色体编码(Chromosome presentation)

HPMSP问题是由单机和两阶段流水线组成的并行的生产调度问题。因此, 本文采用带分割符的顺序编码方法。以5个工件的HPMSP问题为例, 图2所示的染色体表示工件5, 3以方式1加工, 其顺序为5→3, 工件1, 4, 2以方式2加工, 顺序为1→4→2, “0”为分隔符, 用于区分两种方式加工的工件集。

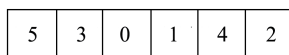


图2 染色体的表示方法

Fig. 2 Chromosome representation

#### 3.2 初始种群生成(Generating initial population)

为了生成高质量的初始种群, 本文首先利用ERT-ECT(the earliest release time and earliest completion time rule)启发式方法(如图3所示)产生一个好的初始个体, 然后通过对其执行基因移位变异的方式产生50%的优良初始个体, 剩余的个体则利用完全随机产生。

#### 3.3 选择(Selection)

采用锦标赛选择机制从种群中选取优良个体进行遗传操作。该方法首先从种群中随机选择两个个体, 将非支配个体置于配偶池中; 若两个个体为非支配关系, 则它们均被置于配偶池中。

**Procedure** Algorithm of ERT-ECT heuristic method:

**begin**

将集合  $J$  中工件按到达时间升序排序;

令  $t_1 = t_2 = t_3 = 0; S_1 = S_2 = \emptyset;$

**while** ( $J = \emptyset$ )

选择  $J$  中第一个工件;

$t'_1 = \max\{r_i, t_1\} + p_{j1}; t'_2 = \max\{t_1, t_2\} + p_{j2};$

$t'_3 = \max\{r_i, t_3\} + p_{j3};$

**if** ( $t_2 \leq t'_1 + w_j^{\max}$ ) **then**

$S_1 = S_1 \cup \{j\}; t_1 = t'_1; t_2 = t'_2;$

**else if** ( $t'_2 + p' \leq t'_3$ ) **then**

$S_1 = S_1 \cup \{j\}; t_1 = t'_1; t_2 = t'_1 + p';$

**else**

$S_2 = S_2 \cup \{j\}; t_3 = t'_3;$

**endif**

$J = J - j;$

**endwhile**

合并  $S_1$  和  $S_2$  生成一个初始个体;

**end**

符号:

$J$ : 工件集;

$S_1$ : 生产线1上工件加工顺序;

$S_2$ : 生产线2上工件加工顺序;

$t_m$ : 设备  $m$  的最早可用时间,  $m = 1, 2, 3;$

$t'_m$ :  $t_m$  的临时值,  $m = 1, 2, 3;$

$p_{jm}$ : 工件  $j$  在设备  $m$  上的处理时间,  $m = 1, 2, 3.$

图3 ERT-ECT启发式方法伪码图

Fig. 3 Pseudo-code of ERT-ECT heuristic method

#### 3.4 交叉和变异(Crossover and mutation)

根据第2.1节中的染色体编码方法, 本文将分隔符视为一个虚拟工件编号来设计交叉和变异操作。交叉操作采用如下步骤: 1) 随机选择父代个体parent1分隔符左侧的基因直接遗传给子代个体offspring1; 2) 从另一个父代个体parent2中删除已选择的基因, 并将剩余基因顺序添加到offspring1的剩余位置。依此方法, 产生另一个子代个体offspring2, 如图4所示。

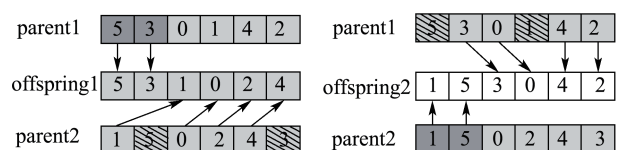


图4 交叉方法示意图

Fig. 4 Graph of crossover

变异操作采用随机选择基因换位, 基因移位或基因段逆序的方法。

### 3.5 局部搜索(Local search)

为了获得更好的Pareto解集, 通过结合HPMSP问题的特点, 本文在求解算法中嵌入了一种局部搜索方法, 对交叉和变异产生的新个体, 将以概率 $\beta$ 执行局部搜索(如图5所示). 局部搜索通过修复策略和改进策略两个步骤进行, 其中修复(repair)策略是将设备2处的延迟工件转移到设备3处任意位置加工, 以提高非支配解的分布性; 改进策略则采用随机选择基因换位(swap), 基因移位(insert)或基因段逆序(reverse)方法, 以改进算法的局部开发能力.

**Procedure** Algorithm of local search method:

**begin**

**if** (rand(0, 1) <  $\beta$ ) **then**

  //修复策略

  对个体 $x$ 执行修复获得 $x'$ ;  $P = P \cup \{x'\}$ ;

  //改进策略

  令iter=0;

**while** (iter < 5)

    对个体 $x'$ 随机选择基因换位, 移位或逆序操作获得 $x''$ ;

    iter = iter + 1;

**if** ( $x''$ 支配 $x'$ ) **then**

$P = P \cup \{x''\}$ ; **break**;

**endif**

**if** ( $x''$ 和 $x'$ 为非支配关系) **then**

$P = P \cup \{x''\}$ ;  $x' = x''$ ;

**endif**

**endwhile**

**endif**

**end**

符号:

$x$ : 执行局部搜索的个体;

$P$ : 当前种群.

图 5 局部搜索方法伪码图

Fig. 5 Pseudo-code of local search method

### 3.6 提出的改进NSGA-II算法 (The proposed improved NSGA-II)

根据上述算法设计, 本文提出了一种求解HPMSP问题的改进NSGA-II算法(improved NSGA-II, INSGA-II), 其伪码如图6所示. 在完成算法参数和种群的初始化后, 算法将进入如下迭代过程: 首先从当前种群 $P$ 选择出一定数量的个体构成配偶池MP, 并对配偶池MP中的个体执行交叉和变异操作; 然后对每个新产生个体以一定概率 $\beta$ 执行局域搜索: 最后在剔除当前种群 $P$ 和配偶池MP中相同个体后, 采用个体分层和拥挤距离的方法<sup>[13]</sup>产生下一代种群. 若种群中个体数量不足, 则利用随机个体补充.

## 4 仿真实验(Simulation experiment)

本文利用随机生成的不同规模( $n = 20, 40, 60, 80,$

100)HPMSP问题进行仿真实验, 每种规模下生成算例, 所有算例均采用如下相同的参数设置: 工件在设备1, 2, 3上的处理时间分别服从 $U[40, 60]$ ;  $U[60, 80]$ 和 $U[80, 120]$ ; 工件的到达时间服从 $U[1, 20n]$ ; 工件在设备2处的最大等待时间服从 $U[0, (\sum |p_{j1} - p_{j2}|)/n]$ ; 工件二次加热所需时间为20. 为了更好地检验所提出的INSGA-II算法在求解HPMSP问题的有效性, 仿真实验中将以NSGA-II, MNSGA-II<sup>[15]</sup>和AMOSA<sup>[16]</sup>为比较对象, 利用4种算法分别对每个算例运行50次, 将每次获得的非支配解集作为该算法每次运行的求解结果, 最终将取的平均值(mean), 最好值(best), 最差值(worst)及标准偏差(std)作为算法评估的依据.

**Procedure** Algorithm of the proposed improved NSGA-II:

**begin**

  初始化算法参数;

  初始化种群 $P(0)$ (如第2.2节);

$t = 0$ ;

**repeat**

    采用赌轮选择方法从当前种群 $P(t)$ 中选择个体组成配偶池MP( $t$ )(如第2.3节);

    对MP( $t$ )中的个体执行交叉和变异操作(如第2.4节);

    对MP( $t$ )中的个体, 以概率 $\beta$ 执行局部搜索操作

    (如第2.5节);

    删除MP( $t$ )和 $P(t)$ 中的相同个体;

    合并MP( $t$ )和 $P(t)$ , 并生成下一代种群 $P(t+1)$ ;

$t = t + 1$ ;

**until** a stop condition is met

**end**

图 6 提出的改进NSGA-II算法伪码图

Fig. 6 Pseudo-code of the proposed Improved NSGA-II

### 4.1 算法参数设置(Algorithm parameter setting)

INSGA-II, NSGA-II和MNSGA-II的参数设置为: 种群大小均设为60; 变异率均设为0.1; MNSGA-II的其他参数与原文相同. 为保证4种算法具有相同的评价次数, INSGA-II的迭代代数设为400次, 而NSGA-II和MNSGA-II的迭代代数设为600次, 这主要是考虑到INSGA-II中个体以概率 $\beta = 0.1$ 执行的局域搜索也会消耗额外的评价次数. AMOSA的参数设置为: 初始温度设为10000°, 停止温度设为100°, 冷却系数设为0.95, 每一状态下迭代400次. 所有算法均采用C++语言编写, 在Intel 3.06 GHz 12 GB内存的计算机环境下运行. 实验分别采用符号A, B, C及D表示算法INSGA-II, NSGA-II, MNSGA-II及AMOSA.

### 4.2 实验结果分析(Experiment result analysis)

首先, 验证4种算法在非支配解集收敛性及获得非支配解数量方面的性能. Zitzler<sup>[17]</sup>提出采用 $C$ 指标来评价两种算法所获得的非支配解集收敛性的优劣, 计

算方法如式(12), 其中 $X$ 和 $Y$ 分别表示两种算法获得的非支配解集. 若 $C(X, Y) = 1$ 说明对于 $Y$ 中的任一个非支配解,  $X$ 中总存在支配它的解;  $C(X, Y) = 0$ 表示对于 $Y$ 中任一个解,  $X$ 中都不存在支配它的解.

$$C(X, Y) = \frac{|y \in Y | \exists x \in X : x \succ y|}{|Y|}. \quad (12)$$

表1和表2给出了INSGA-II, NSGA-II, MNSGA-II及AMOSA的实验结果, 包括 $C$ 指标及非劣解的个数的对比. 从 $C$ 指标上分析, INSGA-II的求解结果优于NSGA-II, MNSGA-II和AMOSA, 获得的非支配解集均能支配NSGA-II, MNSGA-II和AMOSA获得的支配解集, 且INSGA-II算法具有较好的稳定性. 从表2中实验结果可以看出, INSGA-II算法获得的非劣解的个数要多于其它3种算法, 且随着问题规模的增大, INSGA-II获得的非支配解的数量优势愈明显.

表1 4种算法 $C$ 指标实验结果对比

Table 1 Comparison of experimental results on four algorithms via  $C$  index

$n$	算法	平均值±标准差	最优值	最差值
20	$C(A, B)$	1.0±0.0	1.0	1.0
	$C(A, C)$	1.0±0.0	1.0	1.0
	$C(A, D)$	1.0±0.0	1.0	1.0
	$C(B, A)$	0.0±0.0	0.0	0.0
	$C(C, A)$	0.0±0.0	0.0	0.0
	$C(D, A)$	0.0±0.0	0.0	0.0
40	$C(A, B)$	1.0±0.0	1.0	1.0
	$C(A, C)$	1.0±0.0	1.0	1.0
	$C(A, D)$	1.0±0.0	1.0	1.0
	$C(B, A)$	0.0±0.0	0.0	0.0
	$C(C, A)$	0.0±0.0	0.0	0.0
	$C(D, A)$	0.0±0.0	0.0	0.0
60	$C(A, B)$	1.0±0.0	1.0	1.0
	$C(A, C)$	1.0±0.0	1.0	1.0
	$C(A, D)$	1.0±0.0	1.0	1.0
	$C(B, A)$	0.0±0.0	0.0	0.0
	$C(C, A)$	0.0±0.0	0.0	0.0
	$C(D, A)$	0.0±0.0	0.0	0.0
80	$C(A, B)$	1.0±0.0	1.0	1.0
	$C(A, C)$	1.0±0.0	1.0	1.0
	$C(A, D)$	1.0±0.0	1.0	1.0
	$C(B, A)$	0.0±0.0	0.0	0.0
	$C(C, A)$	0.0±0.0	0.0	0.0
	$C(D, A)$	0.0±0.0	0.0	0.0
100	$C(A, B)$	1.0±0.0	1.0	1.0
	$C(A, C)$	1.0±0.0	1.0	1.0
	$C(A, D)$	1.0±0.0	1.0	1.0
	$C(B, A)$	0.0±0.0	0.0	0.0
	$C(C, A)$	0.0±0.0	0.0	0.0
	$C(D, A)$	0.0±0.0	0.0	0.0

表2 4种算法获得非支配解数量实验结果对比

Table 2 Comparison of experimental results on four algorithms via number of nondominated solutions

$n$	算法	平均值±标准差	最优值	最差值
20	A	13.54±0.54	14	12
	B	12.64±0.66	13	10
	C	12.64±0.91	14	10
	D	12.58±0.78	14	12
40	A	25.26±0.66	26	22
	B	18.32±1.05	20	15
	C	19.22±2.79	22	11
	D	20.38±2.30	24	11
60	A	35.46±0.73	37	34
	B	26.26±3.53	31	12
	C	21.74±1.11	24	19
	D	25.1±2.42	29	20
80	A	47.82±1.32	50	44
	B	29.6±3.64	33	15
	C	28.28±3.69	33	14
	D	28.18±5.74	37	19
100	A	55.02±2.02	61	52
	B	29.7±2.42	33	20
	C	32.58±5.58	39	15
	D	37.74±6.73	43	18

非支配解集的分布性是评价多目标算法性能的一个重要指标, 表3给出了4种算法获得的非支配解集分布性方面的实验结果.

分布性指标 $\Delta$ 的计算公式如式(13)所示. 考虑到两个目标的量纲不同, 这里将非支配解集中各目标值进行规范化处理, 规范化方法如式(14). 其中,  $f_{\max}$ 和 $f_{\min}$ 分别表示对应目标在非支配解集中的最大值和最小值,  $f_i$ 和 $f'_i$ 分别表示原目标值和规范化后目标值.

$$\Delta = \sum_{i=1}^{|PF|} \frac{d_i - \bar{d}}{|PF|}, \quad (13)$$

$$f'_i = \frac{f_i - f_{\min}}{f_{\max} - f_{\min}}. \quad (14)$$

从表3中的实验结果可以看出, INSGA-II获得的非支配解集具有更好的分布性, 且具有较好的稳定性.

此外, 图7-10给出了4种算法分别针对不同规模的算例, 选择50次运行所获得的最好非支配解集的分布情况.

显然, 从图中可以直观的看出INSGA-II所获得的非支配解集无论从质量上还是分布性上都要远远优于其他3种算法.

最后, 验证4种算法在极端点质量方面的性能, 实验结果表明4种算法均能获得的最小总延迟工件数量为0的调度方案, 这里仅列举总流水时间的极端点质



量方面的实验结果, 如表4所示.

表 3 4种算法的分布性对比

Table 3 Comparison of experimental results on four algorithms via spread index

$n$	算法	平均值 $\pm$ 标准差	最优值	最差值
20	A	$2.68\text{E}-02\pm 0.00$	$4.19\text{E}-02$	$1.86\text{E}-02$
	B	$4.91\text{E}-02\pm 0.01$	$9.89\text{E}-02$	$2.32\text{E}-02$
	C	$6.87\text{E}-02\pm 0.02$	$1.33\text{E}-01$	$3.30\text{E}-02$
	D	$4.94\text{E}-02\pm 0.01$	$1.00\text{E}-01$	$2.61\text{E}-02$
40	A	$1.39\text{E}-02\pm 0.00$	$2.19\text{E}-02$	$1.15\text{E}-02$
	B	$4.62\text{E}-02\pm 0.01$	$1.21\text{E}-01$	$3.10\text{E}-02$
	C	$4.59\text{E}-02\pm 0.01$	$6.39\text{E}-02$	$2.88\text{E}-02$
	D	$4.52\text{E}-02\pm 0.01$	$8.14\text{E}-02$	$2.65\text{E}-02$
60	A	$1.00\text{E}-02\pm 0.00$	$1.80\text{E}-02$	$7.77\text{E}-03$
	B	$4.44\text{E}-02\pm 0.01$	$7.62\text{E}-02$	$2.43\text{E}-02$
	C	$4.26\text{E}-02\pm 0.01$	$6.86\text{E}-02$	$2.36\text{E}-02$
	D	$4.08\text{E}-02\pm 0.00$	$7.44\text{E}-02$	$1.33\text{E}-02$
80	A	$7.89\text{E}-03\pm 0.00$	$1.18\text{E}-02$	$6.14\text{E}-03$
	B	$4.11\text{E}-02\pm 0.01$	$6.68\text{E}-02$	$2.64\text{E}-02$
	C	$3.99\text{E}-02\pm 0.01$	$5.91\text{E}-02$	$2.48\text{E}-02$
	D	$3.77\text{E}-02\pm 0.01$	$6.38\text{E}-02$	$1.83\text{E}-02$
100	A	$7.12\text{E}-03\pm 0.00$	$1.02\text{E}-02$	$5.31\text{E}-03$
	B	$4.26\text{E}-02\pm 0.01$	$7.25\text{E}-02$	$2.31\text{E}-02$
	C	$4.04\text{E}-02\pm 0.01$	$6.36\text{E}-02$	$2.79\text{E}-02$
	D	$3.85\text{E}-02\pm 0.01$	$7.11\text{E}-02$	$1.48\text{E}-02$

表 4 4种算法的极端点质量对比

Table 4 Comparison of experimental results on four algorithms in extreme points

$n$	算法	平均值 $\pm$ 标准差	最优值	最差值
20	A	$4.91\text{E}+03\pm 2.69\text{E}+02$	$5.83\text{E}+03$	$4.71\text{E}+03$
	B	$5.99\text{E}+03\pm 3.63\text{E}+02$	$7.22\text{E}+03$	$5.79\text{E}+03$
	C	$6.68\text{E}+03\pm 7.60\text{E}+02$	$7.16\text{E}+03$	$6.62\text{E}+03$
	D	$6.66\text{E}+03\pm 3.10\text{E}+02$	$7.75\text{E}+03$	$6.33\text{E}+03$
40	A	$1.58\text{E}+04\pm 7.81\text{E}+02$	$2.04\text{E}+04$	$1.55\text{E}+04$
	B	$2.58\text{E}+04\pm 6.69\text{E}+02$	$2.81\text{E}+04$	$2.56\text{E}+04$
	C	$2.63\text{E}+04\pm 1.09\text{E}+03$	$2.99\text{E}+04$	$2.56\text{E}+04$
	D	$2.64\text{E}+04\pm 1.31\text{E}+03$	$2.72\text{E}+04$	$2.40\text{E}+04$
60	A	$3.67\text{E}+04\pm 9.33\text{E}+02$	$4.27\text{E}+04$	$3.61\text{E}+04$
	B	$6.17\text{E}+04\pm 1.04\text{E}+03$	$6.54\text{E}+04$	$6.08\text{E}+04$
	C	$6.09\text{E}+04\pm 1.00\text{E}+03$	$6.22\text{E}+04$	$5.97\text{E}+04$
	D	$6.27\text{E}+04\pm 2.13\text{E}+03$	$7.15\text{E}+04$	$6.07\text{E}+04$
80	A	$6.86\text{E}+04\pm 3.01\text{E}+03$	$8.13\text{E}+04$	$6.67\text{E}+04$
	B	$1.17\text{E}+05\pm 3.31\text{E}+03$	$1.31\text{E}+05$	$1.15\text{E}+05$
	C	$1.12\text{E}+05\pm 3.65\text{E}+03$	$1.27\text{E}+05$	$1.09\text{E}+05$
	D	$1.16\text{E}+05\pm 5.85\text{E}+03$	$1.43\text{E}+05$	$1.13\text{E}+05$
100	A	$1.06\text{E}+05\pm 3.94\text{E}+03$	$1.27\text{E}+05$	$1.05\text{E}+05$
	B	$1.87\text{E}+05\pm 6.30\text{E}+03$	$2.19\text{E}+05$	$1.83\text{E}+05$
	C	$1.84\text{E}+05\pm 4.41\text{E}+03$	$1.98\text{E}+05$	$1.79\text{E}+05$
	D	$1.83\text{E}+05\pm 1.26\text{E}+04$	$2.08\text{E}+05$	$1.71\text{E}+05$

从表4中可以看出, INSGA-II获得的最小总流水时间优于NSGA-II, MNSGA-II和AMOSA. 由此可以看出, INSGA-II算法在单目标的优化效果上也具有优势.

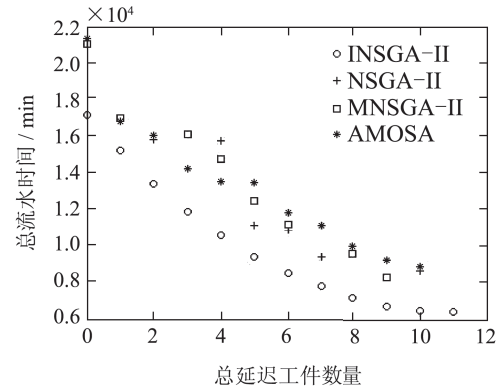


图 7 4种算法求解问题规模为20的非支配解集对比

Fig. 7 Distribution of nondominated solutions obtained by four algorithms in solving problem size 20

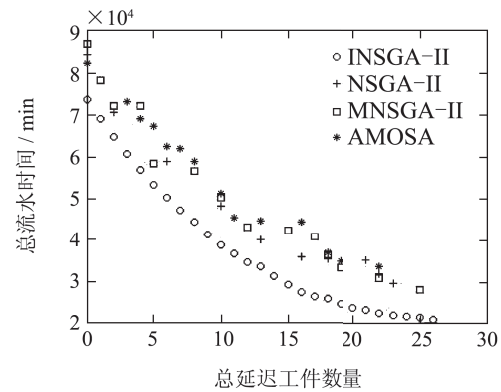


图 8 4种算法求解问题规模为40的非支配解集对比

Fig. 8 Distribution of nondominated solutions obtained by four algorithms in solving problem size 40

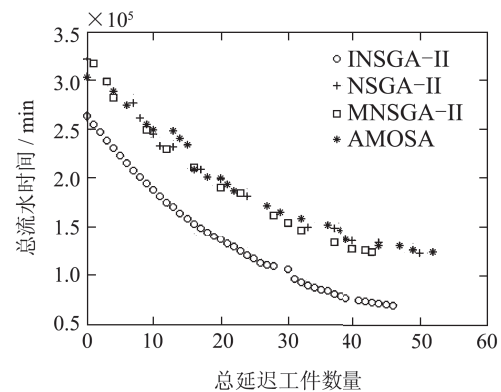


图 9 4种算法求解问题规模为80的非支配解集对比

Fig. 9 Distribution of nondominated solutions obtained by four algorithms in solving problem size 80

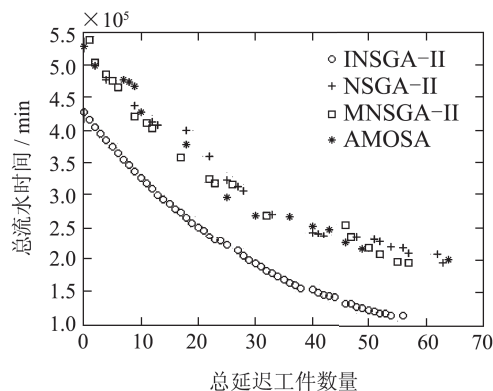


图 10 4种算法求解问题规模为100的非支配解集对比

Fig. 10 Distribution of nondominated solutions obtained by four algorithms in solving problem size 100

## 5 结论(Conclusion)

针对实际生产环境存在串并联并存的生产单元, 本文提出了一种特殊的混合并行机调度问题, 建立了以最小化总流水时间和最小化总延迟工件数量为目标的多目标混合整数规划模型, 并设计一种改进的NSGA-II算法进行求解. 通过对不同规模下HPMSP问题进行大量的仿真实验, 实验结果表明本文所提出的算法获得的非支配解集在收敛性, 分布性及极端点质量方面均具有较好的效果.

## 参考文献

- [1] PINEDO M. *Scheduling: Theory, Algorithms, and Systems* [M]. Berlin: Springer, 2012.
- [2] 谭园园, 宋健海, 刘士新. 加热炉优化调度模型及算法研究 [J]. 控制理论与应用, 2011, 28(11): 1549 – 1557.  
(TAN Yuanyuan, SONG Jianhai, LIU Shixin. Model and algorithm for scheduling reheating furnace [J]. *Control Theory & Applications*, 2011, 28(11): 1549 – 1557.)
- [3] 汪恭书, 唐立新. 并行机实时调度问题的LR&CG算法 [J]. 控制与决策, 2013, 28(6): 829 – 836.  
(WANG Gongshu, TANG Lixin. LR & CG algorithm for parallel machine real-time scheduling problem [J]. *Control and Decision*, 2013, 28(6): 829 – 836.)
- [4] YANG D L, YANG S J. Unrelated parallel-machine scheduling problems with multiple rate-modifying activities [J]. *Information Sciences*, 2013, 235(12): 280 – 286.
- [5] 李作成, 钱斌, 胡蓉, 等. 求解一类异构并行机调度问题的分布估计算法 [J]. 计算机集成制造系统, 2013, 19(9): 2202 – 2212.  
(LI Zuocheng, QIAN Bin, HU Rong, et al. Estimation of distribution algorithm for a kind of heterogeneous parallel machine scheduling problem [J]. *Computer Integrated Manufacturing Systems*, 2013, 19(9): 2202 – 2212.)
- [6] HSU C J, CHENG T C E, YANG D L. Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time [J]. *Information Sciences*, 2011, 181(20): 4799 – 4803.
- [7] SUNDARARAGHAVAN P S, KUNNATHUR A S, VISWANATHAN I. Minimizing makespan in parallel flowshops [J]. *Journal of the Operational Research Society*, 1997, 48(8): 834 – 842.
- [8] 庞哈利, 万珊珊. 并行流程车间调度问题及其概率学习进化算法 [J]. 控制理论与应用, 2005, 22(1): 149 – 152.  
(PANG Hali, WAN Shanshan. Parallel flow shop scheduling problem using probability learning based evolutionary algorithm [J]. *Control Theory & Applications*, 2005, 22(1): 149 – 152.)
- [9] JIANG Y, WAN S. Parallel flow shop scheduling problem using quantum algorithm [C] // *Proceedings of the International Conference on Applied Informatics and Communication, ICAIC 2011*. Xi'an: Springer, 2011: 269 – 274.
- [10] LEI D. Multiobjective production scheduling: a survey [J]. *The International Journal of Advanced Manufacturing Technology*, 2009, 43(9/10): 926 – 938.
- [11] GEN M, LIN L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey [J]. *Journal of Intelligent Manufacturing*, 2014, 25(5): 849 – 866.
- [12] LENSTRA J K. *Sequencing by Enumerative Methods* [M]. Amsterdam: Mathematisch Centrum, 1976.
- [13] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [14] ZHOU A, QU B Y, LI H, et al. Multiobjective evolutionary algorithms: a survey of the state of the art [J]. *Swarm and Evolutionary Computation*, 2011, 1(1): 32 – 49.
- [15] BANDYOPADHYAY S, BHATTACHARYA R. Solving multiobjective parallel machine scheduling problem by a modified NSGA-II [J]. *Applied Mathematical Modeling*, 2013, 37(10): 6718 – 6729.
- [16] BANDYOPADHYAY S, SAHA S, MAULIK U, et al. A simulated annealing-based multiobjective optimization algorithm: AMOSA [J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(3): 269 – 283.
- [17] ZITZLER E. *Evolutionary algorithms for multiobjective optimization: methods and applications* [D]. Switzerland: Swiss Federal Institute of Technology, 1999.

## 作者简介:

付亚平 (1985–), 男, 博士研究生, 研究方向为多目标优化算法、生产计划与调度, E-mail: fuyaping0432@163.com;

黄敏 (1968–), 女, 博士, 教授, 博士生导师, 研究方向为物流与供应链管理、生产计划、调度与存储控制、风险管理和软计算等, E-mail: mhuang@mail.neu.edu.cn;

王洪峰 (1979–), 男, 博士, 副教授, 硕士生导师, 研究方向为进化计算、生产及供应链管理, E-mail: hfwang@mail.neu.edu.cn;

王兴伟 (1968–), 男, 博士, 教授, 博士生导师, 研究方向为分布式优化与决策、计算机网络、分布式多媒体系统等, E-mail: wangxw@mail.neu.edu.cn.