

离散 NSGA-II 求解带有限缓冲区的多目标 批量流水线调度问题^①

韩玉艳¹ 李俊青^{1,2} 桑红燕¹ 包云¹

(1. 聊城大学 计算机学院, 山东 聊城 252059; 2. 山东师范大学 信息科学与工程学院, 山东 济南 250014)

摘要 针对带有限缓冲区的多目标批量流水线调度问题, 首先构建该问题的数学模型, 采用离散 NSGA-II 算法进化求解. 所提算法充分利用非支配解信息引导种群进化, 从而加快了种群收敛速度, 改进了算法的全局探搜和局部开发能力. 将所提算法用于求解太阳能电池组件生产调度的 13 个算例, 实验结果表明, 所提算法能够产生高质量的解, 有效求解该问题.

关键词 有限缓冲区; 多目标批量流水线; 离散; NSGA-II

中图分类号 TP181

文献标识码 A

0 引言

生产调度研究的问题是针对于某项可以分解的工作, 在满足一定技术和资源约束的前提下, 合理安排各部分工作占用的资源、加工时间以及加工次序, 使某一或几项性能指标达到最优. 一般流水线调度问题的工件是完整且不可分割的, 但在实际生产环境中, 企业对产品的完工时间和加工成本要求很高, 因此, 为了加速生产流程, 降低生产成本, 将工件分割成若干小批量, 每个小批量在机器上加工完毕之后, 可允许进入下一个机器上加工, 这就是所谓的批量流水线调度^[1,2]. 批量流水线调度(Lot-streaming Flow Shop, LSFS)问题是很多实际生产调度的简化模型之一. 由于 LSFS 能在很大程度上节省产品完工时间, 提高生产效率, 因此具有广泛的工程应用背景和重要的理论研究价值^[3].

太阳能电池组件生产过程是一个典型的批量流调度问题.《中国制造 2025》作为我国实施制造强国战略第一个十年的行动纲领, 明确提出了“创新驱动、质量为先、绿色发展、结构优化、人才为本”的基本方针, 努力构建高效、清洁、低碳、循环的绿色制造体系. 太阳能的合理利用是对可持续发展、节能减排、保护环境的积极响应和倡导, 而太阳能电池的生产则是充分利用太阳能的必经之路, 对于保障能源自主安全具有重要的战略意义.

近几十年来, 很多专家对单目标 LSFS 问题的求解进行了研究. 譬如, 遗传算法(Genetic Algorithm, GA)^[4]、和声搜索(Harmony Search, HS)^[5]算法、粒子群(Particle Swarm Optimization, PSO)^[6]算法、人工蜂群(Artificial bee colony, ABC)^[7,8]算法、分布估计算法(estimation of the distribution algorithm, EDA)^[9]等. 然而上述文献仅考虑了单目标情况, 而实际生产车间调度还需考虑完工时间、交货期时间、总流经时间等多个目标, 因此多目标调度具有更广泛的现实意义^[3]. 求解多目标调度问题, 一种方法是通过加权方式将多目标转化为单目标. Tseng 等^[10]采用离散的 PSO, 解决等量分批的多目标 LSFS 问题, 该算法优化带权重的提前完工时间和延迟时间. Marimuthu 等^[11]采用蚁群算法和模拟退火算法优化带启动

① 收稿日期: 2017-09-28

基金项目: 国家自然科学基金项目(61773246, 61503170, 61773192, 61603169); 山东省自然科学基金项目(ZR2017BF039, ZR2016FL13)资助

通讯作者: 李俊青, 男, 汉族, 博士, 教授, 研究方向: 智能优化与调度, E-mail: 1141231492@qq.com.

工完毕之后,如果下游机器正在被占用,此时,将机器 i 上完工的工件放入缓冲区中,这样机器 i 处于空闲状态,供下一个小批量使用.如果缓冲区已满,则已完工的工件被阻塞在当前机器上,直到下游机器或者缓冲区可用为止.这样上述 LSFS 问题就转化为有限缓冲区的 LSFS 问题.若优化的目标函数个数大于 1,则上述问题又被转化为有限缓冲区的多目标 LSFS 问题.根据上述问题描述以及约束条件,建立以下太阳能电池组件有限缓冲区多目标 LSFS 数学模型,其中优化的目标函数如

$$\min \begin{cases} f_1 = D_{\tau(e),m}, \\ f_2 = \sum_{j=1}^n \max(0, D_{\tau(l_{\pi(j)}),m} - d_j), \end{cases} \quad (1)$$

其中

$$D_{\tau(1),1} = p_{\pi(1),1}, \quad (2)$$

$$D_{\tau(1),i} = D_{\tau(1),i-1} + p_{\pi(1),i}, i = 2, 3, \dots, m. \quad (3)$$

$$D_{\tau(e),1} = D_{\tau(e-1),1} + p_{\pi(j),1}, e = 2, 3, \dots, B_1 + 1, j = 1, 2, \dots, n. \quad (4)$$

$$D_{\tau(e),1} = \max(D_{\tau(e-1),1} + p_{\pi(j),1}, D_{\tau(e-B_1-1),2}), e = B_1 + 2, B_1 + 3, \dots, \sum_{k=1}^j l_{\pi(k)}, j = 1, 2, \dots, n. \quad (5)$$

$$D_{\tau(e),i} = \max(D_{\tau(e-1),i}, D_{\tau(e),i-1}) + p_{\pi(j),i}, e = 2, 3, \dots, B_i + 1, j = 1, 2, \dots, n, i = 2, 3, \dots, m - 1. \quad (6)$$

$$D_{\tau(e),i} = \max(\max(D_{\tau(e-1),i}, D_{\tau(e),i-1}) + p_{\pi(j),i}, D_{\tau(e-B_i-1),i+1}), e = B_i + 2, B_i + 3, \dots, \sum_{k=1}^j l_{\pi(k)}, j = 1, 2, \dots, n, i = 2, 3, \dots, m - 1. \quad (7)$$

$$D_{\tau(e),m} = \max(D_{\tau(e-1),m}, D_{\tau(e),m-1}) + p_{\pi(j),m}, e = 2, 3, \dots, \sum_{k=1}^j l_{\pi(k)}, j = 1, 2, \dots, n. \quad (8)$$

f_1 和 f_2 分别是最大完工时间和推迟完工时间, π 为一个包含 n 个不同型号电池组件(工件)序列,即 $\pi = \{\pi(1), \pi(2), \dots, \pi(j), \dots, \pi(n)\}$, 每个工件 $\pi(j)$ 被分割为若干小批次,即 $\pi(j) = \{\pi(j)^1, \pi(j)^2, \dots, \pi(j)^{l_{\pi(j)}}\}$, $l_{\pi(j)}$ 为工件 $\pi(j)$ 包含的小批量总数, τ 为所有工件的小批量集合, $\tau = \{\tau(1), \tau(2), \dots, \tau(l_{\pi(1)}), \dots, \tau(\sum_{k=1}^{j-1} l_{\pi(k)} + 1), \tau(\sum_{k=1}^{j-1} l_{\pi(k)} + 2), \dots, \tau(\sum_{k=1}^{j-1} l_{\pi(k)} + l_{\pi(j)}), \dots, \tau(\eta)\}$, 其中 $\eta = \sum_{j=1}^n l_{\pi(j)}$; B_i 为机器 i 与 $i+1$ 之间缓冲区个数; $p_{\pi(j),i}$ 为工件 $\pi(j)$ 在机器 i 上的加工时间; $D_{\tau(e),i}$ 为小批量 $\tau(e)$ 离开机器 i 上的时间; d_j 为工件 $\pi(j)$ 的交货期.

2 离散 NSGA-II

在求解多目标调度问题时,已有的 NSGA-II 在个体比较、排序、选择等策略上进行了改进,而针对交叉和变异算子并没有给出进一步的研究.传统的交叉和变异算子的随机性和无方向性,不能保证新生成的解具有较高的质量,从而降低了算法的收敛速度.鉴于此,本文利用种群个体之间的差异和非支配解优良基因信息,提出了离散 NSGA-II 算法(如图 2 所示).

2.1 初始化种群

在初始化种群中,若存在若干个非支配解,则可加快进化速度,使得优化解更趋向于 Pareto 最优解.此外,由于中间缓冲区的限制,使得 LS-

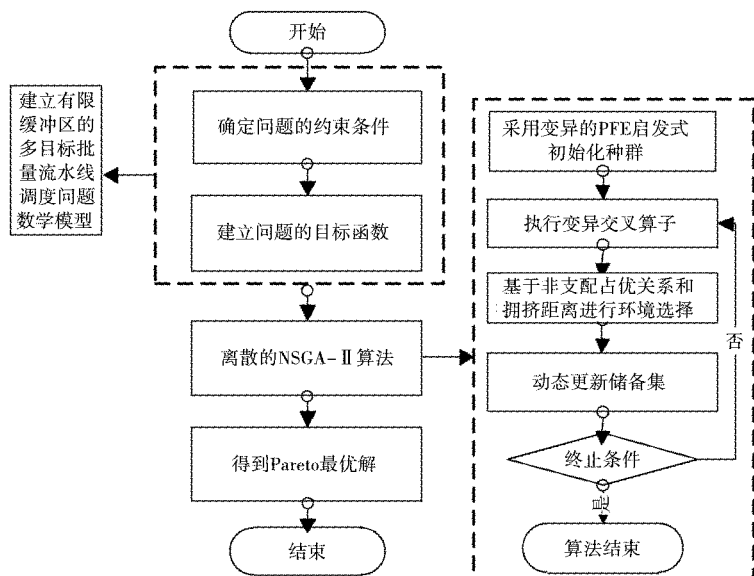


图 2 所提算法流程图

FS 问题的若干小批量阻塞在机器上,导致未调度的小批量延迟开工.基于该问题的约束特性,McCormick 等人提出了曲线拟合方法(Profile fitting, PF)^[14],其目的是试图找到一个序列,使得工件在机器上的阻塞时间与机器空闲时间之和最小.本文针对问题的多目标和多约束特性,基于 NEH^[15]启发式方法,提出了变异的 PFE(variable of PF and NEH, vPFE)启发式,该方法基于非支配占优关系,邻域搜索若干个较好初始解.为了保证种群多样性,剩余解随机产生.具体步骤如下:

令 π 为包含 k 个已调度的工件序列, U 为含有 μ 个待调度工件的集合.首先,采用 PF 启发式产生一个初始解;然后,结合 NEH 启发式对上述初始解邻域搜索,基于占优关系保留 β 个较好的解;最后,采用随机函数,生成种群中剩余解,具体步骤如下

步骤 1:执行 PF 启发式

步骤 1.1:选择总加工时间最小的工件作为 π 的第一个工件,即 $\pi = \{\pi(1)\}$;

步骤 1.2:令 $U = U - \{\pi(1)\}$, $k = 1$;

步骤 1.3:计算 π 中最后一个工件的离开时间 $D_{l(e),i}$, $i = 1, 2, \dots, m$, $e = \sum_{k'=1}^k l_{\pi(k')}$;

步骤 1.4:从待排序序列 U 中取出工件 j , ($j \notin \pi$), 放入 π 的第 $k+1$ 个位置.计算工件 j 的离开时间 $D'_{l(e'),i}$.然后根据公式(9),计算工件 j 所引起的机器空闲时间和阻塞时间之和,

$$t_{j,[k]} = \sum_{i=1}^m (D'_{w(e'),i} - D_{w(e),i} - \sum_{k'=1}^{l_{\pi(j)}} p_{\pi(j),i}), i = 1, 2, \dots, m, e' = \sum_{k'=1}^{k+1} l_{\pi(k')} \quad (9)$$

步骤 1.5:重复步骤 1.4,直到 U 中每一个工件都考虑完毕.选择 $t_{j,[k]}$ 最小的工件作为 π 的第 $k+1$ 个工件 $\pi(k+1)$;

步骤 1.6:令 $U = U - \{\pi(k+1)\}$, $k = k+1$, 执行步骤 1.4,直到 $U = \varphi$. 输出 PF 产生的初始序列 $\pi = \{\pi(1), \pi(2), \dots, \pi(n)\}$.

步骤 2:执行 vNEH 启发式

步骤 2.1:从 π 中取出前两个工件 $\pi(1), \pi(2)$ 进行排列组合,得到两个部分排序 $\{\pi(1), \pi(2)\}$ 与 $\{\pi(2), \pi(1)\}$, 评价这两个排序的目标函数 f_1 , 选择 f_1 最小作为当前排序 π^* , 令 $k = 2$;

步骤 2.2:取出 π 的第 $k+1$ 个工件 $\pi(k+1)$, 将其分别插入到 π^* 的 $k+1$ 个不同位置,共产生 $k+1$ 个排序,选择 f_1 最小的排序赋值给 π^* ;

步骤 2.3:令 $k = k+1$, 若 $k < n$, 则转步骤 2.2; 否则 $k = n$ 执行步骤 2.4;

步骤 2.4:取出 $\pi(n)$, 将其分别插入到 π^* 的 n 个不同位置,共产生 n 个完整排序,并利用 Pareto 占优关系进行比较,选择 β 个 Pareto 最优解,执行步骤 3.

步骤 3:随机产生剩余工件

步骤 3.1:令 $t = PS - \beta$;

步骤 3.2:采用随机函数 $\pi_i = \text{random_shuffle}(\pi_{t-1}.begin(), \pi_{t-1}.end())$, 产生工件序列,其中,随机函数中的第一个参数是指向序列 π_{t-1} 首元素的迭代器,第二个参数则指向序列 π_{t-1} 最后一个元素的下一个位置;

步骤 3.3:如果产生的解 π_i 在种群中存在,则舍去,执行步骤 3.2, 否则,对其评价后,执行步骤 3.4;

步骤 3.4:令 $t = t+1$, 若 $t \leq PS$, 执行步骤 3.2, 否则终止算法. 输出初始种群.

2.2 子代生成

交叉和变异是 NSGA-II 的关键部分,本文利用种群中不同个体间的差异性和外部储备集中精英个体有效信息,产生子代种群中每个子个体,具体步骤如下

步骤 1:变异过程

步骤 1.1:从当前种群中,随机选择 3 个不同的解, π_a, π_b, π_c ;

步骤 1.2:令 $j = 1$;

步骤 1.3:工件序列 π 中每一分量通过式(10)获取

$$\pi(j) = (\pi_b(j) - \pi_c(j)) * flag \circ \pi_a(j), \quad (10)$$

其中, a, b, c 是 1 到 PS 之间的随机数, 代表种群中个体标号, $\pi(j)$ 是工件序列 π 的第 j 个工件; $flag$ 是布尔变量, 取值为 0 或 1, 运算符“ $*$ ”是乘号, 运算符“ \circ ”是取余运算符, 即, $(\pi_b(j) - \pi_c(j)) * flag \circ \pi_a(j) = ((\pi_b(j) - \pi_c(j)) * flag + \pi_a(j) + n) \% n + 1$;

步骤 1.4: $j = j + 1$; 如果 $j \leq n$, 执行步骤 1.3, 否则, 输出新解 π , 执行步骤 2.

步骤 2: 交叉过程

步骤 2.1: 将上述变异得到的 π 中重复的工件删除, 得到新的工件序列 π_p , π_p 中工件个数记为 k .

步骤 2.2: 从非支配解集中随机选择一个解 π' , 利用非支配解的有价值信息, 引导种群进化;

步骤 2.3: 从 π' 中删除与 π_p 相同的工件, 得到部分工件序列 π_{new} , 这样保留了非支配解部分有价值信息, 对其进行邻域搜索, 引导种群快速收敛;

步骤 2.4: 令 $t = 1$;

步骤 2.5: 从 π_p 中取出第 t 个工件, 即 $\pi_p(t)$, 将其插入到 π_{new} 中不同的位置, 得到 $n - k + t$ 个不同工件序列, 选择 f_1 最小的工件序列, 重新赋值给 π_{new} ;

步骤 2.6: $t = t + 1$; 如果 $t \leq k$; 执行步骤 2.5, 否则, 执行步骤 2.7;

步骤 2.7: 得到完整的工件序列 π_{new} , 对其进行评价后, 放入子代种群中.

2.3 环境选择与外部储备集更新

通过变异交叉操作得到 PS 大小的子代种群 O . 合并当前父代种群 P 与子代种群 O , 生成具有 $PS + PS$ 个体的种群 PO , 采用 Pareto 占优关系对种群 PO 中个体进行等级划分, 结合拥挤距离选择 PS 个体, 更新父代种群. 关于个体等级划分和拥挤距离方法, 可参考文献[16], 本文不再赘述. 每一次迭代过程中都产生一定数量的有限非支配解, 为了避免每次迭代生成的非支配解丢失, 本文设置一个有限的外部储备集来保留每代搜索到的 Pareto 解, 并动态更新储备集.

3 实验分析

所有算法均采用具有相同库函数的 Visual C++ 6.0 编码, 操作系统为 Microsoft Windows 7X64, PC 环境为 Intel (R) Core(TM) 3 GHz 和 2 G 内存. 此外, 在所有算法执行过程中, 没有并行程序运行, 并将可能占用系统资源的后台程序关闭. 本文考虑了 13 不同型号的太阳能电池组件, 分别为: QX6060-4/80, QX6060-4/120, QX6060-4/70, QX5555-4/40, QX6060-8/100, QX6060-8/80, QX6060-8/60, QX6060-5/100, QX6060-5/80, QX5555-6/50, Q5555-8/50, QX7070-8/50, QX8060-8/100. 每一型号在 11 个生产工艺阶段的加工时间分别在下列取值范围内随机产生: [30 s, 40 s], [30 s, 40 s], [30 s, 40 s], [120 s, 160 s], [1 500 s, 2 000 s], [300 s, 400 s], [180 s, 240 s], [120 s, 160 s], [60 s, 80 s], [180 s, 240 s], [60 s, 80 s]. 仿真实验参数设置^[17]如表 1 所示.

表 1 参数设置

参 数	符号	值
工件数	n	13
机器数	m	11
种群大小	PS	20
交叉概率	pc	0.9
变异概率	mu	0.1
每个电池组件批次数量	l	$randIn() \bmod 6 + 1$
交货期	d_j	$randIn() \bmod (15m + 1) + 1\ 500$
外部储备集	EAs	100
每个算例最大运行时间	$time$	4 290 s
每个算例独立运行次数	t	30

(1) 验证初始化策略有效性. 已有的 PFE 启发式算法, 执行一次只能产生一个初始化解, 从而减缓了种群的收敛速度, 为此, 本文提出了 vPFE 启发式方法, 执行一次能够产生 β 个性能较好的初始解, 实验中设置 $\beta = 8$. 为验证所提方法的有效性, 表 2 给出了分别采用 PFE, vPFE 以及随机方法初始化种群, 得到的最优 Pareto 最优解集. 由表 2 可知, 本文所提方法得到 4 个非支配解, 并分别占优 PFE 和随机得到的非支配解集中的 2 个解和 1 个解, 说明所提的 vPFE 启发式方法能够产生高质量的初始种群.

表 2 PFE, vPFE 和随机方法初始化种群的结果

算法	最优调度序列	最优的目标函数/h	
		最大完工时间	推迟时间
vPFE	2,6,10,9,5,8,4,3,7,13,12,11,1	12.141 1	75.743 3
	7,12,13,9,3,5,8,6,11,2,4,10,1	12.151 1	75.333 1
	2,9,12,1,4,10,6,8,7,3,5,13,11	12.162 2	68.799 4
	6,9,12,7,13,8,2,1,4,10,3,11,5	12.177 5	67.221 7
PFE	9,4,8,7,5,6,13,12,11,10,3,2,1	12.141 7	82.370 0
	9,2,13,12,11,7,1,4,6,10,5,8,3	12.152 8	73.951 4
	10,7,6,1,12,2,5,13,9,4,3,8,11	12.166 4	69.241 1
随机方法	2,4,7,6,5,10,12,13,8,11,9,1,3	12.152 2	74.517 2
	11,12,5,7,3,2,9,10,13,6,4,8,1	12.148 9	76.019 4

(2) 验证交叉过程中利用非支配解信息的有效性. 本文方法利用非支配解有价值信息作为局部引导方向, 以变异过程得到的部分解作为全局引导, 产生新个体. 为验证非支配解对种群的收敛起到引导作用, 表 3 给出了交叉过程中利用非支配解信息和无非支配解信息(DNSGA-II_n), 得到的最优 Pareto 最优解集. 由表 3 可知, 本文利用非支配解信息作为邻域搜索对象, 能够得到 2 个 Pareto 最优解, 并占优 DNAGA-II_n产生的 2 个非支配解. 这说明, 交叉过程中非支配解的有价值信息能够引导种群快速收敛.

表 3 DNSGA-II 和 DNSGA-II_n 结果

算法	优化调度序列	优化的目标函数/h	
		最大完工时间	推迟时间
DNSGA-II	2,12,9,6,7,3,4,10,13,11,5,8,1	12.141 1	62.582 8
	2,12,9,6,7,3,4,10,13,1,11,5,8	12.162 5	61.990 6
	2,12,9,6,7,3,4,10,13,1,5,11,8	12.162 5	62.211 7
DNSGA-II _n	2,12,9,6,7,3,4,10,13,1,5,8,11	12.162 2	62.449 4
	2,12,9,6,7,3,10,4,13,1,11,8,5	12.162 8	62.028 3

表 4 DNSGA-II, NSGA-II 和 DABC 实验结果

算法	最优调度序列	最优的目标函数/h	
		最大完工时间	推迟时间
DNSGA-II	2,12,1,6,7,3,4,10,13,11,5,8,1	12.141 1	62.582 8
	2,12,9,6,7,3,4,10,13,1,11,5,8	12.162 5	61.990 6
	2,12,9,6,7,3,10,4,13,1,11,8,5	12.162 7	62.028 3
NSGA-II	2,12,9,6,7,3,4,10,13,1,5,8,11	12.162 2	62.494 4
	2,12,9,6,7,3,4,13,10,11,5,8,1	12.141 1	62.593 8
	2,12,9,6,7,3,4,10,13,11,1,5,8	12.162 5	62.350 0
DABC	12,2,6,9,10,7,13,3,4,11,5,8,1	12.152 5	63.358 6
	2,12,3,9,6,7,4,13,10,11,5,8,1	12.141 1	63.545 8

(3) 验证所提 DNSGA-II 方法有效性. 本文采用 DNSGA-II、NSGA-II 和 DABC 算法, 对 13 种不同

型号太阳能电池组件生产进行仿真.表 4 给出了 3 种算法分别运行 4 290 s 之后所得最优解集.由表 4 可知,尽管 NSGA-II 得到了 4 个非支配解,但是本文方法得到的最优解支配 NSGA-II 得到的 3 个非支配解,并均占优 DABC 得到的 2 个非支配解.从结果分析可知,DNSGA-II 明显优于 2 个对比方法.

为了检验所提算法的收敛性,图 3 以参考解集与所有对比算法得到的 Pareto 解集之间的距离为指标^[17]给出了 3 种算法随时间增长的进化曲线.

$$D(S(i)) = \frac{1}{|S^*|} \sum_{y \in S^*} d_y(S(i)),$$

$$d_y(S(i)) = \min_{x \in S(i)} \sqrt{\sum_{k=1}^2 \left(\frac{f_k(x) - f_k(y)}{f_k^{\max} - f_k^{\min}} \right)^2}, \quad (11)$$

其中 $f_k(\cdot)$ 表示第 k 个目标函数值, f_k^{\max} 和 f_k^{\min} 分别代表在参考解集 S^* 中第 k 个目标函数的最大最小值, $S(i)$ 表示算法 i 得到的非支配解, $i \in \{\text{DNSGA-II}, \text{NSGA-II}, \text{DABC}\}$, S^* 是所有算法运行 3 100 毫秒后得到的最优 Pareto 解集. $|S^*|$ 表示参考解集 S^* 中包含的非支配解个数.由公式(11)可知,距离值越小,说明该算法得到的 Pareto 最优解性能越好.由图 3 可见,随着时间增长,本文所提方法的进化曲线明显低于对比算法的进化曲线,说明所提方法的收敛性非常好,能够以较快的收敛速度,接近于真实 Pareto 前沿.

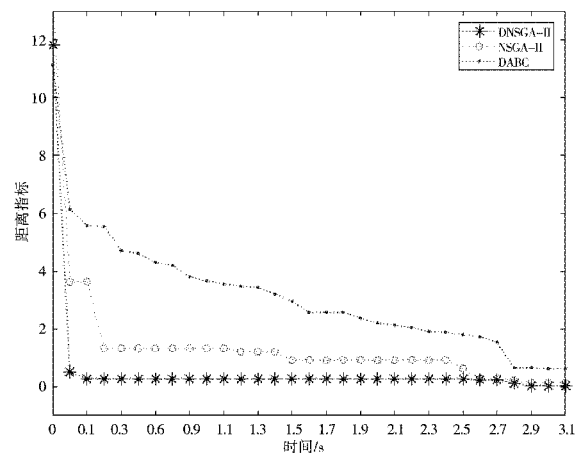


图 3 进化曲线

4 结论

在传统的 LSFS 问题的基础上,融入有限缓冲区这一约束条件,构建了有限缓冲区的多目标 LSFS 问题的数学模型,采用离散 NSGA-II 算法对上述构建的优化问题进行求解.该算法充分利用非支配解信息,引导种群进化,并通过 13 个太阳能电池组件生产调度算例,验证所提算法的有效性和可行性.

下一步考虑将本文方法在更多实际调度问题中的应用,以全面评价所提方法的性能.此外,针对该优化问题,还需考虑加工时间不确定、机器故障、交货期不确定等不确定因素,构建更符合实际生产的数学模型.

参 考 文 献

- [1] 桑红燕.一种基于差分进化的批量流水线调度算法[J].聊城大学学报(自然科学版),2009,22(1):32-35.
- [2] 潘玉霞,潘全科,桑红燕,等.解决批量流水线调度问题的离散微粒群算法[J].聊城大学学报(自然科学版),2009,22(3):90-93.
- [3] Han Y Y, Gong D W, Jin Y C, et al. Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns[J]. IEEE Transactions on Cybernetics, 2017(99):1-14.
- [4] Defersha F M, Chen M. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem [J]. International Journal of Advanced Manufacturing Technology, 2012, 62(1/4):249-265.
- [5] Pan Q K, Duan J H, Liang J J, et al. A novel discrete harmony search algorithm for scheduling lot-streaming flow shops[C]// Chinese Control and Decision Conference, 2010:1531-1536.
- [6] Pan Y X, Pan Q K, Sang H Y. Discrete particle swarm optimization algorithm for lot-streaming flow shop problem[J]. Computer Engineering & Applications, 2010,46(6):52-55.
- [7] Pan Q K, Tasgetiren M F, Suganthan P N, et al. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem[J]. China Mechanical Engineering, 2011, 181(12):2455-2468.
- [8] 李俊青,王永,桑红燕,等.人工蜂群优化及其在资源管理中的应用[J].聊城大学学报(自然科学版),2017,30(3):83-87.
- [9] Pan Q K, Ruiz R. An estimation of distribution algorithm for lot-streaming flow shop problems with setup times[J]. Omega, 2012, 40(2):166-180.

- [10] Tseng C T, Liaob C J. A discrete particle swarm optimization for lot-streaming flowshop scheduling problem[J]. *European Journal of Operational Research*, 2008, 191(2):360-373.
- [11] Marimuthu S, Ponnambalam S G, Jawahar N. Threshold accepting and Ant-colony optimization algorithms for scheduling m-machine flow shops with lot streaming[J]. *Journal of Materials Processing Tech*, 2009, 209(2):1026-1041.
- [12] Han Y Y, Gong D W, Sun X Y, et al. An improved NSGA-II algorithm for multi-objective Lot-streaming flow shop scheduling problem[J]. *Journal of production Research*, 2014, 52(8):2211-2231.
- [13] 张小林. X公司太阳能电池制造过程质量控制研究[D]. 济南: 山东大学, 2014.
- [14] McCormick S T, Pinedo M L, Shenker S, et al. Sequencing in an assembly line with blocking to minimize cycle time [J]. *Operations Research*, 1989, 37:925-936.
- [15] Ronconi D P. A note on constructive heuristics for the flowshop problem with blocking[J]. *International Journal of Production Economics*, 2004, 87(1):39-48.
- [16] 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.
- [17] Han Y Y, Gong D W, Sun J Y. A novel hybrid multi-objective artificial bee colony algorithm for the blocking lot-streaming flow shop scheduling problems[J]. *Knowledge-Based Systems*, 2018, 148:115-130.

Discrete NSGA-II for Multi-objective Lot-streaming Flow Shop Scheduling Problem with Limited Buffers

HAN Yu-yan¹ LI Jun-qing^{1,2} SANG Hong-yan¹ BAO Yun¹

(1. School of Computer Science, Liaocheng University, Liaocheng 252059, China; 2. School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

Abstract For the multi-objective lot-streaming flow shop scheduling problem with limited buffers, a mathematic model is first constructed. Then, a discrete NSGA-II algorithm is proposed to solve the above optimization problem, in which the valuable information of non-dominated solutions is adequately utilized to guide the evolution of the population, accelerate the rate of convergence of the population, and balance the algorithm's capability in exploration and exploitation. The proposed algorithm is applied to 13 instances of the solar cell component production scheduling. The experimental results demonstrate that the proposed algorithm can generate solutions with high quality and effectively solve the multi-objective lot-streaming flow shop scheduling problem with limited buffers.

Key words limited buffers; multi-objective lot-streaming flow shop; discrete; NSGA-II