

# 不同支配关系的 NSGA-III 算法在机器人制造单元调度问题中的应用<sup>①</sup>



李晓辉<sup>1</sup>, 高 铎<sup>1</sup>, 杨 晰<sup>2</sup>, 刘元东<sup>1</sup>, 赵 毅<sup>1</sup>, 董 媛<sup>1</sup>

<sup>1</sup>(长安大学 电子与控制工程学院, 西安 710054)

<sup>2</sup>(陕西航天时代导航设备有限公司, 宝鸡 721099)

通信作者: 董 媛, E-mail: yuandong@chd.edu.cn

**摘 要:** 随着经济的发展, 机器人制造单元对制造行业的生产效率和生产质量有很大提高. 相对于传统的柔性制造单元, 带机器人搬运的车间的调度问题还考虑了加工物料的搬运环节. 因此, 生产调度所面临的问题越来越复杂. 针对 Pareto 支配关系在高维多目标优化中的支配能力不足, 本文将 Lorenz 支配和 CDAS 支配分别与 NSGA-III 算法相结合, 并首次应用到带机器人制造单元的高维多目标车间调度问题上来. 考虑到现代生产过程的复杂化, 本文提出对最大完工时间、加工总能耗、交货期提前量、延迟量、生产总成本等多个目标同时进行优化, 用于确定机器人工作时操作状态和搬运顺序, 提高生产效率. 通过实验发现基于 Lorenz 支配和 CDAS 支配的 NSGA-III 算法在该生产调度问题上比传统的 NSGA-III 在解的收敛性和均匀性上表现更优.

**关键词:** NSGA-III; 机器人制造单元调度; 高维多目标优化; 不同支配关系

引用格式: 李晓辉, 高铎, 杨晰, 刘元东, 赵毅, 董媛. 不同支配关系的 NSGA-III 算法在机器人制造单元调度问题中的应用. 计算机系统应用, 2022, 31(2): 279-284. <http://www.c-s-a.org.cn/1003-3254/8336.html>

## Application of NSGA-III Algorithm Based on Different Dominance Relations in Robotic Cell Scheduling Problem

LI Xiao-Hui<sup>1</sup>, GAO Duo<sup>1</sup>, YANG Xi<sup>2</sup>, LIU Yuan-Dong<sup>1</sup>, ZHAO Yi<sup>1</sup>, DONG Yuan<sup>1</sup>

<sup>1</sup>(School of Electronics and Control Engineering, Chang'an University, Xi'an 710054, China)

<sup>2</sup>(Shaanxi Aerospace Times Navigation Equipment Co. Ltd., Baoji 721099, China)

**Abstract:** With the development of economy, robotic of economy, robotic cells have greatly improved the production efficiency and quality of the manufacturing industry. Compared with that of the traditional flexible manufacturing cells, the scheduling problem of shops with robot handling also involves material handling. As a result, the production scheduling problem is becoming increasingly complex. In view of the lack of dominance of the Pareto dominance relation in high-dimensional multi-objective optimization, the Lorenz domination and CDAS domination are combined, respectively, with the non-dominated sorting genetic algorithm-III (NSGA-III) algorithm in this study and applied for the first time to the high-dimensional multi-objective scheduling of shops with robotic cells. Considering the complexity of modern production processes, this study proposes to optimize multiple objectives, such as maximum completion time, total processing energy consumption, delivery lead time, delivery delay time, and total production cost, at the same time to determine the operating state and handling sequence of the robot and improve production efficiency. Experiments show that on the abovementioned production scheduling problem, the NSGA-III algorithm based on Lorenz domination or CDAS domination performs better than the traditional NSGA-III algorithm in the solution convergence and uniformity.

**Key words:** NSGA-III; robotic cell scheduling; high-dimensional multi-objective optimization; different dominance relations

<sup>①</sup> 基金项目: 国家重点研发项目 (2020YFB1600400)

收稿时间: 2021-04-20; 修改时间: 2021-05-28; 采用时间: 2021-06-08; csa 在线出版时间: 2022-01-17

由于机器人制造单元对制造行业的生产效率和生产质量有很大提高,近30年来,生产调度问题<sup>[1]</sup>已成为国内外自动控制领域、计算机科学等领域的研究热点.现代智能制造市场的需求越来越个性化、多样化,使传统的量产模式受到了极大的挑战.机器人制造单元在实现量产的同时,能够保持小批量定制生产的灵活性,能够更好地适应当前智能制造发展的需求,而合理的优化调度是机器人制造单元整体控制的一个重要方面.制造企业为了在激烈的竞争环境中生存和发展,需要提供能够满足客户各种需求的资源调度方案.单一目标的寻优已经满足不了人们日常生活的需要,工业上对于优化目标个数增多的要求使得求解问题的难度显著增加.目前,调度问题的理论和实践方法研究,仍不能满足现实生产需求.传统的优化算法在解决复杂多目标问题中<sup>[2-5]</sup>常常效率低下,难以达到预期.智能优化算法<sup>[6-9]</sup>通常是基于非劣解概念基础上的现代启发式算法,具有更好的处理高维多目标问题的能力.因此,在经典车间调整理论的基础上,本文要针对Pareto支配关系在高维多目标优化中的支配能力不足,将另外两种支配关系与NSGA-III算法相结合去对生产调度问题进行求解去解决机器人制造单元优化上的多目标协同调度优化问题.

### 1 问题描述

本文主要研究的是有关job-shop类型的机器人的智能制造车间的搬运问题. Job-shop类型的机器人制造单元主要由加工站、搬运机器人和控制系统等组成.根据车间调度的工业路线特点,可将job-shop类型的机器人制造单元调度过程描述为:有 $n$ 个工件 $J = \{J_1, J_2, \dots, J_n\}$ 将在 $m$ 台加工机器 $M = \{M_1, M_2, \dots, M_m\}$ 上进行加工,其中 $M_0$ 为装载站兼卸载站.每个工件 $J_i$ 有 $p$ 道加工工序 $O_{ij} = \{O_{i1}, O_{i2}, \dots, O_{ip}\}$ ,且每个工件都有预先设定好的工序加工顺序,所有的工序都严格按照预定的加工工序顺序在 $m$ 台机器上进行.现已知每道工序的加工时间、每道工序对应的加工机器和任意两台机器之间的搬运时间.如图1所示:调度基本过程可描述为,首先, $n$ 个工件在 $M_0$ 上等待加工,当加工任务开始时,机器人按照工序顺序,当工件处于装载区或工件上一道工序已完成时,从该工件所在位置取走工件并搬运到该工件下一道工序对应加工机器上,根据当前加工机器状态决定是否开始加工.当最后一道加工工序完

成后并由机器人将该工件搬运至卸载站上时,则为加工任务结束.

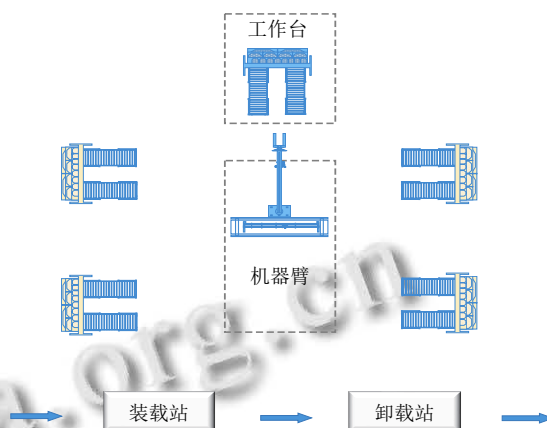


图1 以搬运机械人为中心的制造单元

近年来,在生产过程中人们越来越关注能够使多项决策目标都能得到满足的生产调度方案.通常考虑多个目标共同优化的方案.同时,由于对环境问题的重视,本文又引入了加工总能耗,生产总成本等一系列的绿色调度目标.以下是本文要优化的5个目标:

- 1) 最大完工时间:

$$C_{\max} = \max(C_i), i = 1, 2, \dots, N \tag{1}$$

- 2) 加工总能耗:

$$E_{\text{total}} = \sum_{i=1}^M \left( \sum_{j=1}^{N_i} t_{ij} P_{m,i} + t_{i,\text{idle}} P_{i,\text{idle}} \right) + \sum_{i=1}^{N_b} t_{b,i} P_b + t_{b,\text{idle}} P_{b,\text{idle}} \tag{2}$$

- 3) 交货期提前量:

$$ET = \sum_{j=1}^N (DT_j - RT_j), DT_j > RT_j \tag{3}$$

- 4) 交货期延迟量:

$$LT = \sum_{j=1}^N (RT_j - DT_j), DT_j < RT_j \tag{4}$$

- 5) 生产总成本:

$$C_{\text{total}} = \text{cost}_e E_{\text{total}} + \sum_{i=1}^M \sum_{j=1}^{N_i} t_{ij} C_{u,i} \tag{5}$$

其中,最大完工时间指所有工件完成加工后的总加工时间, $N$ 为工序总数; $C_i$ 为第 $i$ 个工件的完工时间.加工总能耗指调度过程中机器加工总功率与机器人搬运总功率之和. $M$ 为机器总个数; $N_i$ 为第 $i$ 个机器上的加工工序数; $t_{ij}$ 为第 $i$ 个机器上第 $j$ 道工序的加工时间; $P_{m,i}$ 为

第*i*个机器的加工功率;  $t_{i,\text{idle}}$ 为第*i*台机器空载时间;  $P_{i,\text{idle}}$ 为第*i*台机器空载功率;  $N_b$ 为机器人搬运次数;  $t_{b,i}$ 为第*i*次搬运所花费的时间;  $P_b$ 为机器人搬运功率;  $t_{b,\text{idle}}$ 为机器人空载时间;  $P_{b,\text{idle}}$ 为机器人空载功率. 交货期为制造企业交付加工产品的时间. 若工件在理想交货时间前完成生产则为提前, 在理想交货时间后完成则为延迟.  $N$ 为工件个数;  $DT_j$ 为第*j*工件的理想交货时间;  $RT_j$ 为第*j*工件的实际交货时间. 生产总成本为加工所需电费与机床使用成本之和.  $\text{cost}_e$ 为电价(单位能耗成本);  $E_{\text{total}}$ 为加工总能耗;  $C_{u,i}$ 为第*i*台机器单位时间的使用成本.

## 2 算法描述

NSGA-III<sup>[10]</sup>是Deb和Jain在2014年提出来的用于处理多个优化目标的进化优化算法. 其主要思路是在NSGA-II的基础上引入了参考点机制, 对于那些非支配并且接近参考点的种群个体进行保留, 相比NSGA-II拥有优秀的处理更高维度目标的能力.

### 2.1 NSGA-III 基本算法流程

本文以NSGA-III的算法基本框架, 全局搜索以遗传算法为基础, 使用了一种符合调度问题特点的交叉变异策略, 并提出了两种更改支配关系的改进措施. 改进后的NSGA-III将具有更好的处理高维数据的能力, 用于解决本文所提出的机器人制造单元的高维多目标调度问题. NSGA-III的算法流程如下.

Step 1. 初始化, 随机产生大小为 $N$ 的父代种群 $P_t$ . 更新操作, 父代通过交叉、变异产生大小为 $N$ 的子代种群 $Q_t$ , 合并为大小为 $2N$ 的种群 $R_t = P_t + Q_t$ .

Step 2. 计算目标值, 通过非支配排序, 将 $R_t$ 划分为不同的非支配层( $F_1, F_2, \dots$ ).

Step 3. 选择优先级高的非支配子集保留到下一代, 用 $S_t$ 来保留. 设 $F_l$ 为临界层, 临界层为加上该层解数目大于等于 $N$ 时的非支配层. 若数目等于 $N$ 则 $S_t = S_t \cup F_l$ ,  $i = 1, 2, \dots, l$ , 否则 $S_t = S_t \cup F_l, i = 1, 2, \dots, l-1$ . 令 $P_{t+1} = S_t$ . 用 $K = N - |P_{t+1}|$ 来记录需要从 $F_l$ 中补充的解的个数.

Step 4. 每一维度最大值作为极值点, 最小值作为理想点, 对每个解的目标值做归一化处理.

Step 5. 以极值点构造超平面, 产生均布的 $C(M+H-1, H)$ 个参考点,  $M$ 为维度数,  $H$ 为每维分割份数.

Step 6. 对 $S_t$ 中和 $F_l$ 层中每个解找出距离最近的参考点分别进行关联.

Step 7. 对于 $S_t$ 中关联少的参考点, 找出 $F_l$ 层对应参考点关联的一个最近的解进行补充. 每添加一个解, 就将该解从 $F_l$ 中删除, 并将 $S_t$ 中关联解的个数更新排序. 重复Step 7直至补充 $K$ 个解.

Step 8. 返回Step 2, 直至迭代次数达到预设值, 对当前解集进行非支配排序找出第一非支配层的解作为最终解集.

### 2.2 改进的 NSGA-III

随着目标数量的增加, 算法对解的选择压力变小, 基于Pareto支配, 解之间很难存在支配关系, 这导致解集中大部分的解都是非支配的. 而算法本身就强调保留种群中的非支配解, 所以在每一代种群中, 就没有太多的空间来保留新解. 这样就减缓了算法的搜索过程, 使算法效率变得低下. 使用改变支配关系的方法, 将解的目标值进行调整, 可以扩大每个解的支配空间, 从而提高解之间的区分度. 本文分别将Lorenz支配和CDAS支配与NSGA-III结合来进行优化. 第2.2.1节和第2.2.2节分别介绍了Lorenz支配和CDAS支配.

#### 2.2.1 Lorenz 支配

$$\begin{cases} f(x) = [f_1(x), f_2(x), \dots, f_m(x)] \\ f'(x) = [f_1(x), f_1(x) + f_2(x), f_1(x) + f_2(x) \\ \quad + f_3(x), \dots, \sum_{i=1}^m f_i(x)] \end{cases} \quad (6)$$

Lorenz支配(式(6))是将标准化后的每一个解的目标值, 按照从小到大进行排列, 之后在一个新列表里, 第*i*个位置存放前*i*个目标值的和( $i = 1, 2, \dots, m$ ),  $m$ 为维度数. 所有解的目标值 $f(x)$ 都用Lorenz修改过后, 用此新的目标值 $f'(x)$ 来进行非支配排序, 修改过支配关系后, 每个解的支配空间将会变大, 每个解之间就有更好的区分度.

如图2所示是一个双目标的例子, 对于解 $S$ , 基于Pareto支配区域为(a), 经Lorenz支配 $S$ 的支配空间增加到(a)、(b)、(c). 图2中,  $X$ 是一个解序列所求出对应的目标值的坐标.

#### 2.2.2 CDAS 支配

$$\begin{cases} \varphi_i = S \times \pi \\ f'_i(x) = \frac{r \times \sin(\omega_i + \varphi_i)}{\sin \varphi_i}, i = 1, 2, \dots, m \end{cases} \quad (7)$$

CDAS支配(式(7))如图3所示, 式中,  $r$ 为该解的目标值点与原点之间的距离;  $\omega_i$ 为该点与原点连线和

第  $i$  维的夹角;  $\varphi_i$  为该点与第  $i$  维上一个可控的夹角, 其大小由系数  $S$  控制. 当  $S = 0.5$  时, CDAS 支配与 Pareto 支配效果相同; 当  $S < 0.5$  时, CDAS 支配将使原目标值点的支配区域扩大; 当  $S > 0.5$  时, CDAS 支配将使原目标值点的支配区域变小. 图 3 中,  $X$  是一个解序列所求出对应的目标值的坐标.

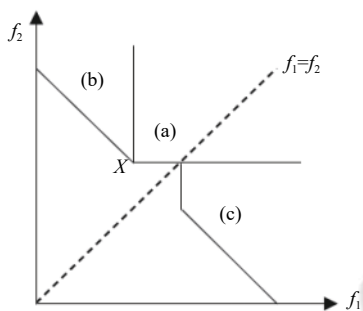


图 2 Lorenz 支配关系

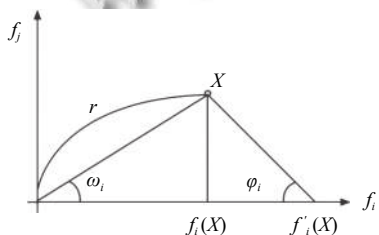


图 3 解  $x$  修改前后第  $i$  维目标值

通过试验发现, 当  $S$  取 0.25 时, 求解多目标优化问题的效果最佳. 此时根据 CDAS 支配将原目标值  $f(x)$  修改为  $f'(x)$ , 扩大该解的支配区域.

以双目标优化问题为例, 如图 4 所示, 当  $S = 0.5$ , 即 Pareto 支配关系下解 A、B、C 是互不支配的. 取  $S = 0.25$ , 增大扩张角, 从图中虚线部分可以看到每个解的支配区域都有所扩大, 解之间出现了更细微的序值分布, 此时, 3 个解从互不支配变为解 B 支配解 C.

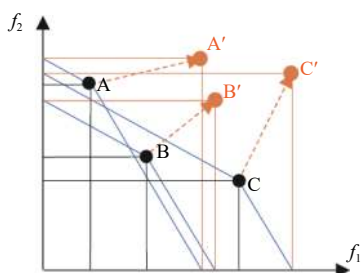


图 4 CDAS 修改后解支配关系

### 3 实验分析

为了验证本文提出的改进算法的有效性, 通过对本文算法与传统的 NSGA-III 算法进行对比. 并从以下 3 个方面来测试算法性能.

- 1) 收敛性: 反映解集与真实 Pareto 前沿趋近程度.
- 2) 多样性: 反映是否所得的 Pareto 最优解集中的解都分布在 Pareto 前沿上.
- 3) 均匀性: 反映所得的 Pareto 最优解集中的解在目标空间中分布的均匀程度.

本次实验采用世代距离 GD (generational distance)、反向世代距离 IGD (inverted generational distance) 和间距 Spacing 作为评价算法性能的指标. 其中, GD 用于评价解集收敛性, IGD 用于评价解集收敛性和多样性的综合性能, Spacing 评价解集均匀程度.

本文提出的改进优化算法以 PyCharm Python 3.8 为开发工具编程实现, 其中相关参数设置如下: 实验设计环境和配置为 Intel(R)Core(TM) i5-7300HQ CPU@2.50、8.00 GB RAM、Windows 10 64 位操作系统. 交叉概率设为 1.0, 变异概率设为 0.5, 迭代次数设为 50, 种群大小设为 40. 表 1 中,  $I_{nd}$  表示案例编号,  $J$  表示该案例工件个数,  $M$  表示加工机器台数,  $S$  表示加工工序总数, 上标 1 为 C-NSGA-III 算法测试结果, 上标 2 为 L-NSGA-III 算法测试结果, 上标 3 为 NSGA-III 算法测试结果, 下标 AVG 为均值, SD 为标准差, MIN 为最小值. 本文挑选了由小到中到大型的案例共 10 个进行了实验, 用来测试本文所提出的算法的性能, 各案例均按照一定规则随机生成. 每个案例运行 10 次, 取均值作为测试结果.

为了更好的描述最优方案的调度过程, 可以通过甘特图来分析工件的加工顺序与机器人的搬运顺序, 如图 5 为案例 CS-3-16-5 的甘特图. 该案例包含 3 个工件, 5 台加工机器, 16 道工序. 图中, 横坐标为时间轴, 纵坐标为加工机器序号, M0 为装载卸载站. 纯色矩形代表加工过程, 虚线矩形代表工件在当前机器上的等待时间, 实线代表机器人带载过程, 虚线代表空载过程. “J00”代表工件 0 的第 0 道工序, 之后以此类推. 则案例 CS-3-16-5 的加工过程可描述为: 机器人将工件 2 从 M0 搬出至 M3 进行工序 J20 的加工, 搬运后机器人返回至 M0 开始搬运工件 0 至 M1 进行工序 J00 的加工, 随后返回 M0 搬运工件 1 至 M5 加工工序 J10. 随后机器人转至 M3 取走在 M3 上完成加工的且已在

输出缓冲区上等待的工件 2, 搬运至 M2 加工工序 J21. 以此类推, 直到所有工件加工结束.

表 1、表 2、表 3 分别为 C-NSGA-III、L-NSGA-III 和 NSGA-III 算法在 GD、IGD、Spacing 上的对比结果. 通过对比可以得出, C-NSGA-III 和 L-NSGA-III 都在 GD 上表现良好, 在绝大部分案例上数值都小于 NSGA-III, 且标准差不大, 说明改进算法较为稳定. 在 GD 最小值上, C-NSGA-III 始终为 0, 总体可得两种

改进算法在收敛性上都要优于 NSGA-III. 但在 IGD 值上, NSGA-III 几乎始终占优, 说明 C-NSGA-III 和 L-NSGA-III 在多样性上会变差, 分析原因是两算法均加大了支配空间, 它们的解相当于 Pareto 关系下的解的子集, 所以解的多样性会变差. 比较 Spacing 值可知, 两改进算法在解集均匀程度上, 较优于 NSGA-III. 总体来说, C-NSGA-III 和 L-NSGA-III 算法有效地提升了解集的收敛性和均匀性, 在多样性上表现一般.

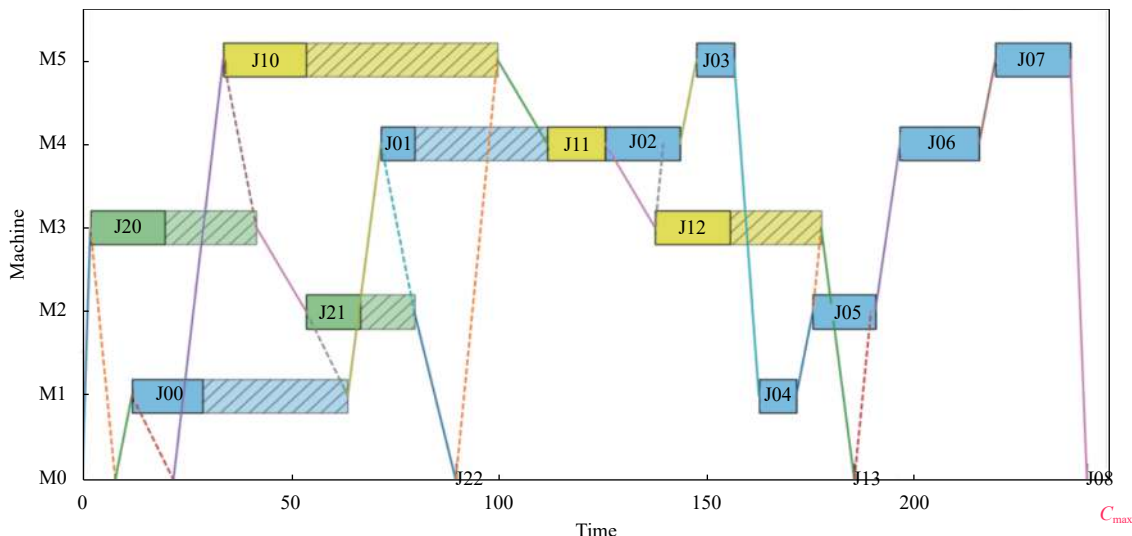


图 5 案例 CS-3-16-5 甘特图

表 1 C-NSGA-III、L-NSGA-III 与 NSGA-III 算法 GD 指标对比表

$I_{nd}$	$J$	$M$	$S$	$GD^1_{AVG}$	$GD^1_{SD}$	$GD^1_{MIN}$	$GD^2_{AVG}$	$GD^2_{SD}$	$GD^2_{MIN}$	$GD^3_{AVG}$	$GD^3_{SD}$	$GD^3_{MIN}$
1	3	5	16	<b>0.002 9</b>	0.002 6	<b>0.000 0</b>	0.004 7	<b>0.002 5</b>	0.002 0	0.007 4	0.003 5	0.001 8
2	4	4	23	0.001 8	0.001 1	<b>0.000 0</b>	<b>0.001 5</b>	<b>0.000 7</b>	0.000 5	0.007 9	0.003 6	0.003 9
3	7	5	40	0.001 5	0.001 4	<b>0.000 0</b>	<b>0.000 8</b>	<b>0.000 8</b>	<b>0.000 0</b>	0.004 5	0.002 1	0.001 7
4	8	6	55	<b>0.004 8</b>	<b>0.005 9</b>	<b>0.000 0</b>	0.011 2	0.007 8	<b>0.000 0</b>	0.014 1	0.006 2	0.004 8
5	11	6	74	<b>0.003 1</b>	<b>0.002 7</b>	<b>0.000 0</b>	0.006 3	0.006 2	<b>0.000 0</b>	0.007 8	0.006 5	<b>0.000 0</b>
6	12	6	86	<b>0.002 6</b>	0.003 1	<b>0.000 0</b>	0.002 8	<b>0.001 8</b>	<b>0.000 0</b>	0.005 3	0.004 1	0.001 1
7	14	8	95	0.009 6	0.007 6	<b>0.000 0</b>	<b>0.007 4</b>	<b>0.004 6</b>	<b>0.000 0</b>	0.011 6	0.008 5	<b>0.000 0</b>
8	16	8	97	0.005 6	<b>0.003 9</b>	<b>0.000 0</b>	<b>0.004 7</b>	0.004 3	<b>0.000 0</b>	0.007 4	0.004 6	<b>0.000 0</b>
9	19	10	107	<b>0.003 2</b>	<b>0.005 5</b>	<b>0.000 0</b>	0.015 7	0.012 0	<b>0.000 0</b>	0.008 0	0.008 2	<b>0.000 0</b>
10	20	9	135	<b>0.019 6</b>	0.032 2	<b>0.000 0</b>	0.021 0	0.028 3	<b>0.000 0</b>	0.022 2	<b>0.014 0</b>	<b>0.000 0</b>

表 2 C-NSGA-III、L-NSGA-III 与 NSGA-III 算法 IGD 指标对比表

$I_{nd}$	$J$	$M$	$S$	$IGD^1_{AVG}$	$IGD^1_{SD}$	$IGD^1_{MIN}$	$IGD^2_{AVG}$	$IGD^2_{SD}$	$IGD^2_{MIN}$	$IGD^3_{AVG}$	$IGD^3_{SD}$	$IGD^3_{MIN}$
1	3	5	16	0.149 1	0.043 5	0.090 6	0.126 1	0.056 0	0.061 8	<b>0.053 7</b>	<b>0.019 0</b>	<b>0.026 7</b>
2	4	4	23	0.219 4	0.045 9	0.155 4	0.113 3	0.031 2	0.073 8	<b>0.059 3</b>	<b>0.024 7</b>	<b>0.028 7</b>
3	7	5	40	0.166 2	0.038 3	0.079 7	0.107 0	0.036 5	0.059 5	<b>0.065 3</b>	<b>0.027 3</b>	<b>0.026 2</b>
4	8	6	55	0.293 1	0.128 3	0.118 8	0.254 2	0.119 2	0.085 6	<b>0.093 2</b>	<b>0.028 8</b>	<b>0.041 1</b>
5	11	6	74	0.245 1	0.042 5	0.165 7	0.282 4	0.091 8	0.158 9	<b>0.096 8</b>	<b>0.035 0</b>	<b>0.052 6</b>
6	12	6	86	0.278 6	0.085 1	0.172 8	0.176 2	0.054 7	0.072 2	<b>0.080 3</b>	<b>0.030 6</b>	<b>0.031 5</b>
7	14	8	95	0.280 7	0.109 1	0.132 8	0.210 6	0.084 0	0.111 8	<b>0.108 7</b>	<b>0.034 0</b>	<b>0.060 4</b>
8	16	8	97	0.227 9	0.062 7	0.133 6	0.230 9	<b>0.042 0</b>	0.151 2	<b>0.106 5</b>	0.048 4	<b>0.048 1</b>
9	19	10	107	0.260 9	0.070 5	0.191 0	0.306 5	0.113 0	0.189 4	<b>0.125 5</b>	<b>0.075 3</b>	<b>0.044 9</b>
10	20	9	135	0.343 4	<b>0.110 7</b>	0.125 7	0.269 5	0.083 6	0.155 9	<b>0.185 4</b>	0.150 6	<b>0.036 8</b>

表3 C-NSGA-III、L-NSGA-III与NSGA-III算法Spacing指标对比表

$I_{nd}$	$J$	$M$	$S$	$S^1_{AVG}$	$S^1_{SD}$	$S^1_{MIN}$	$S^2_{AVG}$	$S^2_{SD}$	$S^2_{MIN}$	$S^3_{AVG}$	$S^3_{SD}$	$S^3_{MIN}$
1	3	5	16	0.049 6	0.036 9	<b>0.010 8</b>	<b>0.046 4</b>	<b>0.025 5</b>	0.018 8	0.107 2	0.048 6	0.044 1
2	4	4	23	<b>0.031 2</b>	0.015 4	<b>0.010 8</b>	0.033 0	<b>0.011 3</b>	0.013 4	0.082 8	0.038 5	0.054 6
3	7	5	40	<b>0.028 5</b>	<b>0.016 2</b>	<b>0.006 8</b>	0.029 3	0.016 9	0.010 7	0.087 3	0.040 3	0.048 8
4	8	6	55	<b>0.053 2</b>	0.036 8	<b>0.009 8</b>	0.058 9	0.022 8	0.022 2	0.086 7	<b>0.020 1</b>	0.058 0
5	11	6	74	<b>0.043 9</b>	<b>0.015 5</b>	0.025 9	0.051 8	0.016 9	<b>0.022 9</b>	0.078 1	0.015 6	0.054 7
6	12	6	86	<b>0.035 4</b>	0.015 9	<b>0.013 4</b>	0.038 7	<b>0.013 9</b>	0.017 5	0.076 5	0.039 9	0.046 7
7	14	8	95	<b>0.053 7</b>	0.028 6	0.029 7	0.062 1	0.024 9	<b>0.023 5</b>	0.085 9	<b>0.024 0</b>	0.059 3
8	16	8	97	<b>0.044 0</b>	<b>0.014 8</b>	<b>0.027 3</b>	0.052 5	0.022 9	0.031 8	0.078 0	0.019 5	0.050 3
9	19	10	107	<b>0.059 4</b>	<b>0.033 6</b>	<b>0.012 9</b>	0.062 5	0.037 7	0.013 1	0.108 2	0.035 0	0.066 8
10	20	9	135	<b>0.061 2</b>	<b>0.031 6</b>	0.018 1	0.076 8	0.046 0	0.034 7	0.102 5	0.049 1	<b>0.017 6</b>

#### 4 结论与展望

本文针对研究机器人制造单元的多目标协同调度问题,同时优化5个目标:最大完工时间,加工总能耗,交货期提前量,交货期延迟量及生产总成本.针对job-shop的特点,分别将Lorenz支配和CDAS支配与NSGA-III结合来进行优化并将结果与NSGA-III进行比较,证明该算法有效.

#### 参考文献

- 1 Haimes YY, Lasdon LS, Wismer DA. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1971, SMC-1(3): 296–297. [doi: 10.1109/TSMC.1971.4308298]
- 2 Zadeh LA. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 1963, 8(1): 59–60. [doi: 10.1109/TAC.1963.1105511]
- 3 Coello CAC. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 2006, 1(1): 28–36. [doi: 10.1109/MCI.2006.1597059]
- 4 Escamilla J, Salido MA, Giret A, *et al.* A metaheuristic technique for energy-efficiency in job-shop scheduling. *The Knowledge Engineering Review*, 2016, 31(5): 475–485. [doi: 10.1017/S026988891600031X]
- 5 Dai M, Tang DB, Giret A, *et al.* Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 2013, 29(5): 418–429. [doi: 10.1016/j.rcim.2013.04.001]
- 6 May G, Stahl B, Taisch M, *et al.* Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, 2015, 53(23): 7071–7089. [doi: 10.1080/00207543.2015.1005248]
- 7 Subulan K. An interval-stochastic programming based approach for a fully uncertain multi-objective and multi-mode resource investment project scheduling problem with an application to ERP project implementation. *Expert Systems with Applications*, 2020, 149: 113189. [doi: 10.1016/j.eswa.2020.113189]
- 8 Soto C, Dorransoro B, Fraire H, *et al.* Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm. *Swarm and Evolutionary Computation*, 2020, 53: 100632. [doi: 10.1016/j.swevo.2019.100632]
- 9 Zhu ZW, Zhou XH. An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. *Computers & Industrial Engineering*, 2020, 140: 106280.
- 10 Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577–601. [doi: 10.1109/TEVC.2013.2281535]