

1-4-2019

Improved Intrusion Weed Algorithm for Solving Flexible Job Shop Scheduling Problem

Zhang Xin

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Li Ke

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Dahu Yan

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Zhicheng Ji

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Improved Intrusion Weed Algorithm for Solving Flexible Job Shop Scheduling Problem

Abstract

Abstract: An improved invasive weed optimization algorithm is proposed for solving multi-objective flexible job shop scheduling problem (FJSP) with released time and delivery date. The minimum completion time of jobs, the maximum work load of machines and the total work load of all machines are taken as the optimization goals to establish a FJSP model. *A random key encoding scheme based on transformed sequences is proposed* and a mapping relationship is set up between the continuous space and the discrete space of FJSP. *An adaptive Gauss mutation operator is introduced to diversify the population in the process of weed breeding.* In spatial diffusion stage, *the principle of Levy flight is taken to improve the global search ability*, which contributes to escape from local optimal solution. The algorithm is compared with other different algorithms and the statistical results show that the algorithm is effective for solving the multi-objective FJSP.

Keywords

flexible job shop scheduling, weed optimization, adaptive Gauss mutation, Levy flight, random key coding

Recommended Citation

Zhang Xin, Li Ke, Yan Dahu, Ji Zhicheng. Improved Intrusion Weed Algorithm for Solving Flexible Job Shop Scheduling Problem[J]. Journal of System Simulation, 2018, 30(11): 4469-4476.

改进入侵杂草算法求解柔性作业车间调度问题

张新, 李珂, 严大虎, 纪志成

(江南大学教育部物联网技术应用工程中心, 江苏 无锡 214122)

摘要: 将入侵杂草算法用于解决考虑工件释放时间、工件交货期的柔性作业车间调度问题, 以工件的最大完工时间、最大机器负荷、机器总负荷为优化目标建立了多目标柔性作业车间调度模型, 提出了一种改进的入侵杂草优化算法。提出一种基于转化序列的随机键编码方式, 实现杂草连续空间与 FJSP 离散空间之间的映射。在种子繁殖阶段, 通过引入自适应高斯变异算子来增加种群多样性。接着, 在种群扩散阶段, 借助 Levy 飞行随机游走策略提高算法全局搜索能力, 跳出局部最优解。最后, 通过对比实验证明算法对于求解多目标柔性作业车间调度问题是有效的。

关键词: 柔性作业车间调度; 杂草优化; 自适应高斯变异; Levy 飞行; 随机键编码

中图分类号: TP391.9 文献标识码: A 文章编号: 1004-731X (2018) 11-4469-08

DOI: 10.16182/j.issn1004731x.joss.201811050

Improved Intrusion Weed Algorithm for Solving Flexible Job Shop Scheduling Problem

Zhang Xin, Li Ke, Yan Dahu, Ji Zhicheng

(Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: An improved invasive weed optimization algorithm is proposed for solving multi-objective flexible job shop scheduling problem (FJSP) with released time and delivery date. The minimum completion time of jobs, the maximum work load of machines and the total work load of all machines are taken as the optimization goals to establish a FJSP model. A random key encoding scheme based on transformed sequences is proposed and a mapping relationship is set up between the continuous space and the discrete space of FJSP. An adaptive Gauss mutation operator is introduced to diversify the population in the process of weed breeding. In spatial diffusion stage, the principle of Levy flight is taken to improve the global search ability, which contributes to escape from local optimal solution. The algorithm is compared with other different algorithms and the statistical results show that the algorithm is effective for solving the multi-objective FJSP.

Keywords: flexible job shop scheduling; weed optimization; adaptive Gauss mutation; Levy flight; random key coding

引言

调度的根本目标是以最小化成本去完成车间



收稿日期: 2018-06-11 修回日期: 2018-07-05;
基金项目: 江苏省产学研联合创新资金-前瞻性联合
研究项目(BY2016022-24);
作者简介: 张新(1993-), 男, 江苏南京, 硕士生,
研究方向为智能调度优化; 李珂(1992-), 男, 安徽
安庆, 硕士生, 研究方向为柔性作业车间调度。

的生产任务, 从而获得最大的经济效益, 所以这就
要求对生产过程中涉及的人和物进行合理的分配。
因此, 车间生产调度问题(Job-shop Scheduling
Problem, JSP)成为了研究的热点, 但是 JSP 的应
用场景与生产实际情况严重不符, 集中表现在其对
机器约束的限制。基于此, Bucker 等提出了柔性
作业车间调度问题(Flexible Job-shop Scheduling

<http://www.china-simulation.com>

• 4469 •

Problem, FJSP)^[1], 减小了对机器约束的限制, 允许同一道工序可任意选择一台有加工能力的机器进行工件加工。

从 FJSP 问题提出以来, 许多经典算法被应用其中, 其中具有代表性的当属遗传算法^[2], 张国辉等^[3]提出将一种具有高质量初始解的改进遗传算法应用于多目标 FJSP 问题, 实例仿真证明了该算法的求解效率和有效性。但经典算法的局限性在于, 它们较为依赖于初始解, 不易根据具体生产情况进行调节, 并且涉及参数较多不易于实现。自 2015 年来, 随着群智能算法的研究浪潮, 越来越多的学者开始使用其作为 FJSP 问题的研究工具。吴秀丽等^[4]使用 5 种细菌算法对 FJSP 问题展开研究, 并对其中寻优能力最强的细菌觅食算法做进一步结构优化, 完善算法结构并最终提升了稳定性。这些算法都意图通过模拟真实世界的生物种群习性由概率模型搜索最优解, 如粒子群算法^[5], 蜂群算法^[6]等。开辟了 FJSP 问题研究的又一思路。

本文介绍的是 2006 年由 Mehrabian^[7]提出的入侵杂草算法(Invasive Weed Optimization Algorithm, IWO), 该算法被运用在多种场景的各类优化问题求解上。在此基础上, Yan-Ying Bai 等^[8]更进一步将两种群智能算法——杂草算法与粒子群算法互补结合, 得到了一种拥有更强全局寻优特性的混合算法。Ahmadi 和 Mojallali^[9]利用混沌搜索的特点对杂草算法进行改进, 提出了一种混沌入侵的杂草算法。这些学者的工作拓展了算法的优势, 也拓宽了算法的优化思路, 能够更加有效解决工件排序和机器选择等若干问题。

IWO 算法在生产调度领域的研究尚不成熟是因为传统的 IWO 算法具有易陷入局部最优, 收敛效果差等缺陷。本文一方面将 IWO 的应用对象扩展到 FJSP 问题上; 另一方面, 在借鉴上述学者的改良思路后, 对种子繁殖阶段和种群扩散阶段两部分进行了优化。为了增加种子在繁殖阶段的种群多样性, 本文在迭代过程中引入了高斯变异算子; 对于种群扩散阶段, 通过使用 Levy 飞行随机游走策

略改善杂草位置, 降低了算法陷入局部最优的概率。并且通过相关实验证明改进后的 IWO 算法在静态调度问题上缩短了最大完工时间, 最后得到了对应的甘特图。

1 问题描述

柔性作业车间调度的应用场景可以描述为: 假设车间中存在生产加工机器 m 台和待加工工件 n 个, 并且每个工件至少经过一道加工工序。在对经典 JSP 问题的研究基础上, FJSP 问题引入了柔性加工路径元素, 即每道工序可以在有加工能力的机器中任意选择一台, 对应的加工时间取决于所选择的机器的性能。为了最优化选定的调度性能指标, 可将多目标 FJSP 问题分解为机器分配和作业排序两个子问题。在本文中, 需同时优化三个调度目标, 分别为最大作业完成时间、机器总负载时间、以及机器最大负载时间。

1.1 变量定义

针对本文中涉及的变量, 作出如下定义:

n : 需加工工件数量;

m : 正常运行机器台数;

i : 第 i 个工件;

j : 第 j 道工序;

k : 第 k 台机器;

L : 工件工序总和;

D : 一个足够大的正数;

K : 工件集合;

M : 可用机器集合;

J_i : 工件 i 所包含的所有工序数;

Q_i : 工件 i 对应的所有工序集合;

T_{ijk} : 机器 M_k 完成第 O_{ij} 道工序所耗费的加工时间;

S_{ijk} : 机器 M_k 进行第 O_{ij} 道工序加工的开始时间;

Z_{ijk} : 机器 M_k 完成第 O_{ij} 道工序的结束时间;

C_i : 工件 i 最后一道工序完成时间即完工时间;

$C_{\max}$: 最后完工工件的完工时间, 即最大完工

时间;

R_i : 工件 i 的释放时间;

D_i : 工件 i 的交货期;

H : 种子搜索随机步长

$Y_{ijk} = \begin{cases} 1, & \text{工序 } O_{ij} \text{ 在机器 } M_k \text{ 上加工} \\ 0, & \text{其他} \end{cases}$

$X_{ijghk} = \begin{cases} 1, & \text{工序 } O_{ij} \text{ 先于工序 } O_{gh} \text{ 在机器 } M_k \text{ 上加工} \\ 0, & \text{其他} \end{cases}$

1.2 FJSP 的调度指标

不同的实例模型采用不同的调度指标, 针对基准调度算例, 本文结合最大完工时间、机器最大负载以及机器总负载 3 个调度指标优化求解。在求解脱期调度实例问题, 采用最大完工时间、最大脱期时间以及平均流经时间 3 个调度优化指标。具体 FJSP 的调度目标和约束条件如下:

$$(1) \text{ 最小化最大完工时间} \\ f_1 = \min(C_{\max}) \quad (1)$$

$$(2) \text{ 最小化机器 } k \text{ 的最大负载} \\ f_2 = \min \left\{ \max_{1 \leq i \leq n} \sum_{j=1}^n \sum_{k=1}^{n_i} T_{ijk} Y_{ijk} \right\} \quad (2)$$

$$(3) \text{ 最小化机器总负载} \\ f_3 = \min \left\{ \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^{n_i} T_{ijk} Y_{ijk} \right\} \quad (3)$$

$$(4) \text{ 最小化最大拖期时间} \\ f_4 = \min \sum_{i=1}^n (\max \{ C_i - D_i \}) \quad (4)$$

$$(5) \text{ 最小化平均流经时间} \\ f_5 = \min \left\{ \sum_{i=1}^n (C_i - R_i) / n \right\} \quad (5)$$

S.t.

$$\begin{cases} S_{ijk} + X_{ijghk} \geq C_{ghk}; \\ \forall i \leq j, \forall j, h, \forall k \in K_{ij} \cap K_{gh}; \end{cases} \quad (6)$$

$$\begin{cases} S_{ghk} + (1 - X_{ijghk}) * D \geq C_{ijk}; \\ \forall i \leq j, \forall j, h, \forall k \in K_{ij} \cap K_{gh}; \end{cases} \quad (7)$$

$$S_{ijk} + C_{ijk} \leq Y_{ijk} * D; \forall i, j, k \in K_{ij}; \quad (8)$$

$$\sum_k Y_{ijk} = 1, k \in K_{ij}, \forall i, j; \quad (9)$$

$$\begin{cases} \sum_k S_{ijk} \geq \sum_k C_{i(j-1)k}; \\ k \in K_{ij}, \forall i, \forall j = 2, 3, \dots, J_i; \end{cases} \quad (10)$$

公式(6)和公式(7)保证了同一台机器不能同时进行多道工序的加工, 公式(8)把开始时间和完工时间初始化为零, 公式(9)限定了每道工序只能指定一台机器完成, 公式(10)保证了工件是按照规定工序进行加工的。

2 改进的入侵杂草算法

2.1 入侵杂草算法

入侵杂草算法是受到自然界杂草扩张性繁殖现象启发而得到的智能优化算法。在该算法中, 问题的可行解用杂草表示。种子可由杂草产生并在杂草附近以正态分布的规律生长为新的杂草, 杂草与种子共同构成待研究问题的可行解集合。当种群数量达到设定的最大种群时, 以适应度值衡量杂草的竞争力并进行“优胜劣汰”。IWO 算法由以下几个部分构成:

(1) 繁殖阶段: 由杂草对应的适应度值决定产生种子的数目, 适应度值越高则杂草产生的后代种子越多, 反之越少。种子数目由公式(11)决定:

$$weed_n = \frac{f - f_{\min}}{f_{\max} - f_{\min}} (s_{\max} - s_{\min}) + s_{\min} \quad (11)$$

式中: f 代表当前杂草对应的适应度值; $f_{\max}$ 和 $f_{\min}$ 分别代表当前杂草种群中杂草适应度的最大和最小值; $s_{\max}$ 和 $s_{\min}$ 代表杂草种群中杂草产生种子的最大和最小数目。计算过程示意图如图 1 所示。

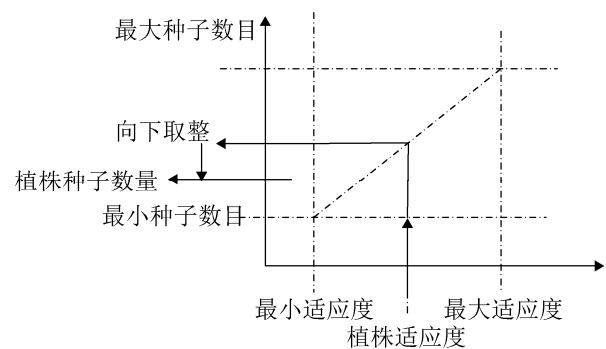


图 1 杂草种子数目计算方法示意图
Fig. 1 Schematic diagram of calculating the number of weed seeds

(2) 扩散阶段: 在 IWO 算法中, 假设种群产生的种子服从标准正态分布, 设定步长 $H \in [-\sigma, \sigma]$ 。 σ 的变化公式为:

$$\sigma = \frac{(iter_{\max} - iter)^n}{(iter_{\max})^n} (\sigma_{init} - \sigma_{final}) + \sigma_{final} \quad (12)$$

式中: n 为非线性调和因子; $iter$ 为当前进化代数; $iter_{\max}$ 为最大进化代数; σ_{init} 和 σ_{final} 分别为标准差的初始值和最终值。

(3) 竞争生存阶段: 随着种群不断的繁殖和扩散, 后代数目将超过环境资源的承受能力, 所以需要设定种群上限最大值, 限制杂草种群无限制生长和扩散。当种群数量达到数量上限值, 则比较杂草与种子的适应值大小, 淘汰弱势个体, 选取适应度强的前 $P_{\max}$ 个个体组成新种群。图 2 为本阶段实现过程。

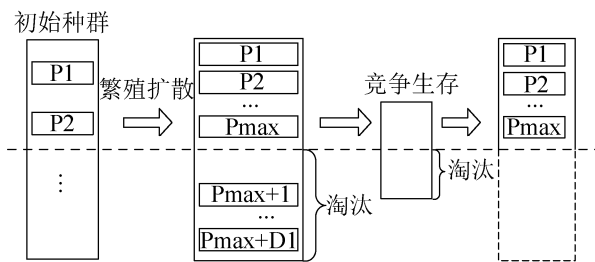


图 2 竞争生存阶段

Fig. 2 Competitive survival stage

2.2 改进入侵杂草算法

IWO 算法主要适用于求解连续空间域的优化问题, 而调度问题属于离散空间的组合优化问题。因此, 本文首先建立一种基于随机键的编码方式, 实现连续空间到离散空间的映射。为改善 IWO 易于陷入局部最优的缺陷, 在原始的算法基础上引入了高斯随机变异算子以提高种群多样性, 得到了一种改良后的入侵杂草算法(IIWO), 并借助 Levy 飞行随机游走策略提高算法全局搜索能力, 具体内容如 2.2.1~2.2.3 节。

2.2.1 编码机制

本文在原有的随机键编码方式上进行改进, 引入了转化序列的策略, 对工序进行编码。

(1) 对于变量 $\zeta_i = [\zeta_{i,1}, \zeta_{i,2}, \zeta_{i,3}, \dots, \zeta_{i,g}]$, 排列后搜索中位数得到中间数列 $\omega_i = [\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,g}]$, 接着由公式 $\beta_{i,\omega_{i,k}} = k (k=1, 2, \dots, g)$, g 为一个很大的正整数, 获得转化序列 $\beta_i = [\beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,g}]$, 结合初始工序排序 χ_0 得到序列 χ_1 , 根据 χ_1 得到新的工序序列。

假设有一杂草的位置向量为 $\zeta_i = [0.62, 0.46, 0.17, 0.25, 0.54, 0.20, 0.94]$, 依次按照顺序编码为 k , 初始工序位置顺序为 $\chi_0 = [1, 1, 1, 2, 2, 3, 3]$, 杂草位置最大值为 0.94, 其对应顺序为 1, 其次为 0.62, 其对应顺序为 2, 以此类推, 最终得到位置向量对应顺序数列 $\omega_i = [2, 4, 7, 5, 3, 6, 1]$ 。根据转化规则, 当 $k=1$ 时 $\omega_{i,1} = 2$, 可得到 $\beta_{i,\omega_{i,k}} = \beta_{i,2} = 1$, 当 $k=2$ 时, 有 $\omega_{i,2} = 4$, $\beta_{i,\omega_{i,k}} = \beta_{i,4} = 2$, 逐一转换, 可得到转化序列 $\beta_i = [7, 1, 5, 2, 4, 6, 3]$, 由 β_i 对初始序列 χ_0 操作, 得到序列 $\chi_1 = [3, 1, 2, 1, 2, 3, 1]$, 最终调度方案中的工序可由 χ_1 确定, 具体编码方案见表 1。

表 1 工序编码方案

Tab. 1 Coding scheme of working procedure							
k	1	2	3	4	5	6	7
ζ_i	0.62	0.46	0.17	0.25	0.54	0.20	0.94
χ_0	1	1	1	2	2	3	3
ω_i	2	4	7	5	3	6	1
β_i	7	1	5	2	4	6	3
χ_1	3	1	2	1	2	3	1
工序	O_{31}	O_{11}	O_{21}	O_{12}	O_{22}	O_{32}	O_{13}
	O_{31}	O_{32}	O_{21}	O_{11}	O_{12}	O_{22}	O_{13}

(2) 确定了调度方案中工序的加工顺序后, 调度的柔性问题就体现在对每个加工工序的机器选择上。在这里借鉴了文献[10]中的转换方法, 通过式(13)将空闲工位的可选机器序号转换为个体的位置向量, 即可获得机器的编号。这里 $z(h)$ 代表 h 所对应的可选且满足工序加工要求的机器所组成的集合, $u(h)$ 代表可选机器在 $z(h)$ 中对应的位置序号, $u(h) \in [1, z(h)]$ 代表工序。

$$u(h) = \text{round}\left(\frac{\zeta_{i,h} + \varepsilon}{2\varepsilon} (z(h) + 1) + 1\right), 1 \leq h \leq g \quad (13)$$

2.2.2 自适应高斯变异算子

在迭代过程中, 为避免算法陷入寻优过程中可能出现的局部最优解, 本文在原始的算法基础上引

入了随机数算子以提高种群多样性。这里的随机数算子由初始个体上加入扰动项得到,扰动项服从均值为 μ ,方差为 σ 的正态分布。在全局搜索过程中,通过改进的随机变异算子,能够得到所需的高斯变异杂草。

加入随机数扰动后的杂草位置更新公式为:

$$X_{i+1} = X_i + X_i \cdot \lambda \cdot \delta \quad (14)$$

式中: $\delta \sim N(0, 1)$ 满足均值为 0, 方差为 1 的高斯分布;

$$\lambda = 1 - \frac{i}{iter_{max} - 1} \quad (15)$$

式中: $i = 1, 2, \dots, D$, 当 λ 值为 1 时变异效果最显著, 而当 λ 值为 0 时几乎没有变异现象。

2.2.3 基于 Levy 飞行的改进寻优策略

Levy 飞行是一种特殊的随机游走策略, 作为一种随机漫步的数学统计模型, 与变异算子结合, 可以加强变异杂草的随机性。

$$L(s) \sim |s|^{-1-\beta}, 0 < \beta \leq 2 \quad (16)$$

Levy 飞行的随机步长 s 可由式(17)得到:

$$s = u / |v|^{1/\beta} \quad (17)$$

式中: u, v 均服从正态分布, $u \sim N(0, \sigma_u^2)$, $v \sim N(0, \sigma_v^2)$ 。

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \quad (18)$$

图 3 所示为 500 次 Levy 飞行步长的波形示意图, 从图中可以看出, 在寻优过程中会得到一些局部最优解, 使用短距离飞行的 Levy 飞行能够在这些解附近继续搜索; 而使用长距离飞行的 Levy 飞行则可以在远离当前最优解的范围内进行搜索。两者相结合, 则大大扩大了搜索范围。在入侵杂草算法中加入 Levy 飞行策略, 能够使 IWO 算法陷入局部极值可能性显著减小。加入 Levy 飞行的扩散阶段由式(19)实现:

$$\begin{cases} \sigma_{new} = s \oplus \sigma \\ H_{new} \in (-\sigma_{new}, \sigma_{new}) \end{cases} \quad (19)$$

式中: s 为 Levy 飞行的步长; σ_{new} 为新的正态分布标准差; H_{new} 为新的搜索步长。

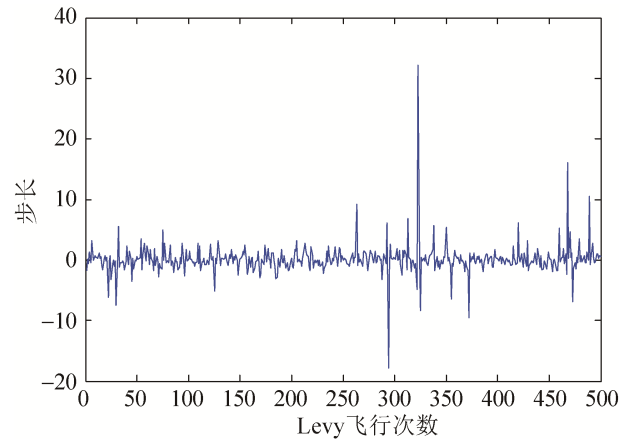


图 3 500 次 Levy 飞行步长
Fig. 3 500 Levy flight step size

IIWO 算法步骤:

step 1: 初始化杂草种群基础信息, 包括初始种群规模、最大种群规模、最大迭代次数、问题维度等;

step 2: 计算所有杂草个体适应度;

step 3: 根据公式(11)计算个体可生成种子数, 并根据公式(19)更新标准差;

step 4: 随机选取部分解, 采用 2.2.2 节随机变异方法进行变异, 并加入到种群中;

step 5: 判断种群是否达到最大规模, 若达到, 则执行 step6, 否则执行 step2;

step 6: 对种群中杂草进行 2.1 节中的竞争淘汰操作, 并挑选出最优解;

step 7: 判断是否达到最大迭代次数, 若达到, 输出最优解, 否则执行 step2。

3 仿真实验与分析

上述改进的入侵杂草优化算法采用 Matlab 编程, 测试分为两部分, 第一部分对基准问题进行测试, 采用的优化调度指标为 f_1 、 f_2 和 f_3 ; 第二部分通过某实际模具生产车间的实例进行测试, 采用的优化调度指标为 f_1 、 f_4 和 f_5 。将求解结果与不同算法求解结果进行对比。

改进的入侵杂草优化算法的运行参数设置如下: 初始种群规模 $P=2$, 每个种群最小杂草个数 $P_{min}=5$, $P_{max}=20$, $S_{min}=1$, $S_{max}=5$, 非线性因子 $n=3$,

最大迭代次数 $K=400$ 。

3.1 基准测试仿真

为验证 IIWO 算法求解 FJSP 问题的有效性,以最大完工时间、最大机器负荷和机器总负荷为优化指标,使用文献[10]中 4×5 、 8×8 和 10×10 三个标准柔性作业车间算例进行测试,并与 Kacem 等^[11]提出的带局部搜索的控制遗传算法(Controlled Genetic Algorithm with Approach by Localization, AL-CGA)、文献[12]提出的小生境粒子群算法(Particle Swarm Optimization and Niching Algorithm, PSO+NI)和文献[13]算法结果进行对比。

3.1.1 8×8 算例

该算例是一个部分柔性作业车间调度问题,共有 8 台机器,8 个工件,总共 27 道工序。将本文算法运行得到的最优 Pareto 非支配解集与其他两种算法进行比较,结果如表 2 所示。图 4 给出了 IIWO 算法 $C_{\max}=15$ 的调度甘特图。

表 2 8×8 实例测试结果
Tab. 2 Test result of 8×8 FJSP

算法	最大完工时间 f_1	最大机器负荷 f_2	机器总负荷 f_3
AL-CGA	16	14	75
	15	13	79
PSO+NI	16	13	75
	15	11	82
IIWO	15	12	73
	15	13	75

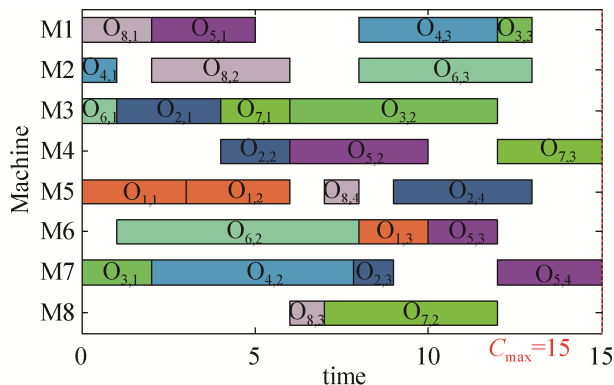


图 4 8×8 算例甘特图

Fig. 4 Gantt chart of 8×8 FJSP

3.1.2 4×5 算例

该算例是一个小规模完全柔性作业车间调度问题,共有 5 台机器,4 个工件,总共 12 道工序。将本文算法运行结果与其他两种算法进行比较,结果如表 3 所示。图 5 给出了 IIWO 算法 $C_{\max}=11$ 的调度甘特图。

表 3 4×5 实例测试结果
Tab. 3 Test result of 4×5 FJSP

算法	最大完工时间 f_1	最大机器负荷 f_2	机器总负荷 f_3
AL-CGA	16	10	34
文献[13]	12	10	32
IIWO	11	10	32
	12	8	33

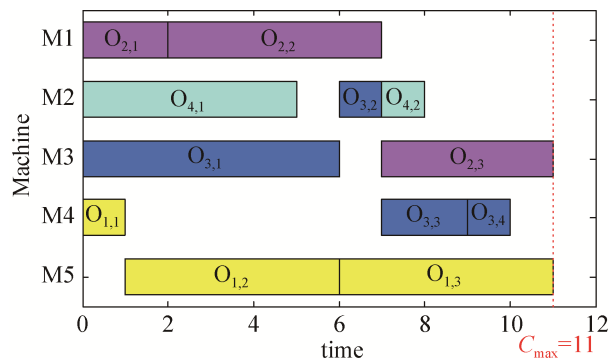


图 5 4×5 算例甘特图

Fig. 5 Gantt chart of 4×5 FJSP

3.1.3 10×10 算例

该算例是一个完全柔性作业车间调度问题,共 10 台机器,10 个工件,总共 30 道工序。对比结果如表 4 所示。图 6 为 IIWO 算法 $C_{\max}=8$ 的调度甘特图。

表 4 10×10 实例测试结果
Tab. 4 Test result of 10×10 FJSP

算法	最大完工时间 f_1	最大机器负荷 f_2	机器总负荷 f_3
AL-CGA	8	7	41
	8	5	42
PSO+NI	8	7	42
	7	6	42
IIWO	8	5	42
	7	6	42

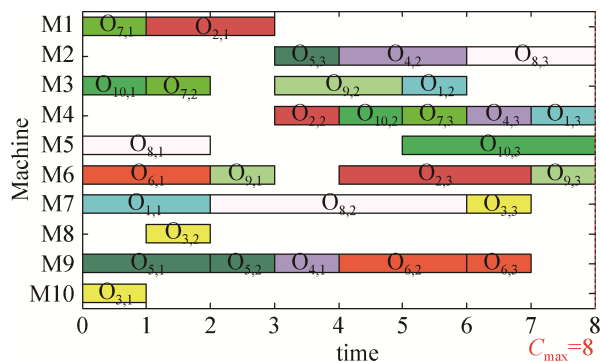


图 6 10×10 算例甘特图
Fig. 6 Gantt chart of 10×10 FJSP

以上三种仿真算例的结果表明, 本文提出的算法在对多目标 FJSP 问题的求解上是有效的。

3.2 实例仿真

为进一步验证 IIWO 算法在解决多目标 FJSP 问题上的可行性和优越性, 通过模拟某离散模具车间, 获取车间的实例数据并取整处理, 得到 12 个工件在 10 台机器上加工共 35 道工序的 FJSP 实例数据, 具体数据信息如附录所示。从表中可以看出, 不同工件具有不同的加工时间、到达时间和交货期, 因此考虑最大完工时间、平均流经时间和最大拖期时间最小三个优化目标, 得到最优 Pareto 非支配解, 并和改进之前的标准入侵杂草算法进行对比, 所得结果如表 5 所示。

表 5 FJSP 问题实例结果比较
Tab. 5 Results comparison on FJSP

算法	最大完工时间 f_1	拖期时间 f_4	平均流经时间 f_5	平均运行时间(s)
IWO	92	23	67.100	290.45
	99	15	71.300	
	97	15	70.900	
IIWO	84	4	60.500	272.737
	87	5	59.600	
	88	5	58.200	

3.3 实验仿真结果分析

从表中数据可以看出, 针对最大完工时间、拖期时间和平均流经时间三个优化目标, 经 IIWO 算法得到的结果均优于标准 IWO 算法, 尤其是针对拖期时间目标, 改进算法优化效果明显。从平均运

行时间上看, 改进之后的算法具有更短的运行时间, 更具优越性。最后, 使用本文改进的入侵杂草算法求解该实例, 得到最大完工时间为 84, 总拖期为 4 时间单位的最优解甘特图如图 7 所示, Pareto 前沿图如图 8 所示。

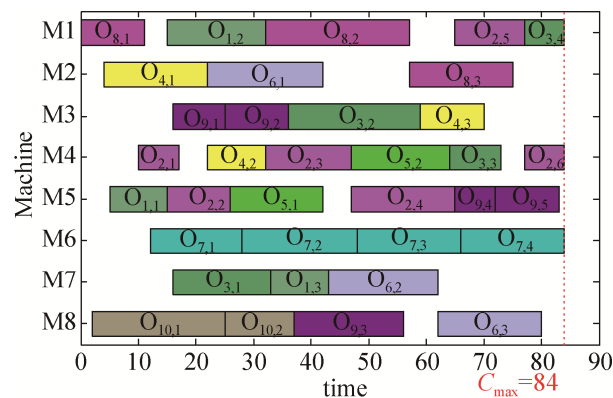


图 7 实例获得的一组调度甘特图
Fig. 7 A set of scheduling Gantt chart obtained by the instance

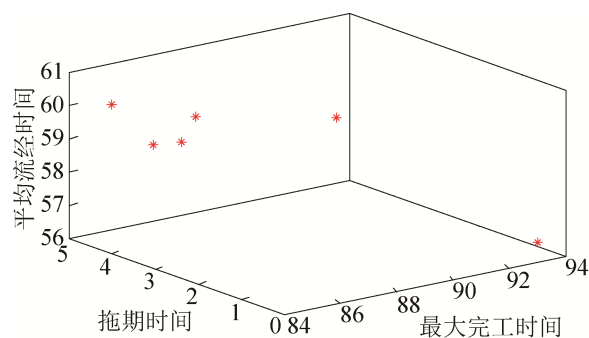


图 8 实例获得的 Pareto 前沿图
Fig. 8 Pareto frontiers obtained by the instance

4 结论

本文将自适应高斯变异算子及 Levy 飞行策略融入 IWO 算法中, 得到了 IIWO 算法, 并将 IIWO 算法应用到多目标柔性作业车间调度问题中。此外, 本文利用基于转化序列的随机键编码方式, 分别在标准测试算例和某实际制造车间案例与其他算法的仿真结果进行比较。仿真结果证明, IIWO 算法收敛速度更快, 精度更高, 获得的调度方案更佳。因此, 在处理多目标 FJSP 问题时, 本文提出的 IIWO 算法是稳定、可靠的。

参考文献:

- [1] Brucker P, Schlie R. Job-shop scheduling with multi purpose machines[J]. Computing (S0010-485X), 1990, 45(4): 369-375.
- [2] Türkyılmaz A, Bulkan S. A Hybrid Algorithm for Total Tardiness Minimisation in Flexible Job Shop: Genetic Algorithm with Parallel VNS Execution[J]. International Journal of Production Research (S0020-7543), 2015, 53(6): 1832-1848.
- [3] 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题[J]. 机械工程学报, 2009, 45(7): 145-151.
Zhang Guohui, Gao Liang, Li Peigen, et al. Improved Genetic Algorithm for the Flexible Job-shop Scheduling Problem [J]. Chinese Journal of Mechanical Engineering, 2009, 45(7): 145-151.
- [4] 吴秀丽, 张志强, 杜彦华, 等. 改进细菌觅食算法求解柔性作业车间调度问题[J]. 计算机集成制造系统, 2015, 21(5): 1262-1270.
Wu Xiuli, Zhang Zhiqiang, Du Yanhua, et al. Improved Bacteria Foraging Optimization Algorithm for Flexible Job Shop Scheduling Problem [J]. Computer Integrated Manufacturing Systems, 2015, 21(5): 1262-1270.
- [5] Singh M R, Singh M, Mahapatra S S, et al. Particle Swarm Optimization Algorithm Embedded with Maximum Deviation Theory for Solving Multi-objective Flexible Job Shop Scheduling Problem[J]. International Journal of Advanced Manufacturing Technology (S0268-3768), 2016, 85(9/10/11/12): 2353-2366.
- [6] 张明, 田娜, 纪志成. 基于适应值欧式距离比的均衡蜂群算法[J]. 系统仿真学报, 2015, 27(5): 980-989.
Zhang Ming, Tian Na, Ji Zhi-cheng. Balanced Bee Colony Algorithm Based on Fitness Euclidean-distance Ratio [J]. Journal of System Simulation, 2015, 27(5): 980-989.
- [7] Mehrabian A R, Lucas C. A Novel Numerical Optimization Algorithm Inspired from Weed Colonization [J]. Ecological Informatics (S1574-9541), 2006, 1(4): 355-366.
- [8] Bai Y Y, Xiao S, Liu C, et al. A Hybrid IWO/PSO Algorithm for Pattern Synthesis of Conformal Phased Arrays[J]. IEEE Transactions on Antennas & Propagation (S1558-2221), 2013, 61(4): 2328-2332.
- [9] Ahmadi M, Mojallali H. Chaotic Invasive Weed Optimization Algorithm with Application to Parameter Estimation of Chaotic Systems[J]. Chaos Solitons & Fractals the Interdisciplinary Journal of Nonlinear Science & Nonequilibrium & Complex Phenomena (S0960-0779), 2012, 45(9/10): 1108-1120.
- [10] Yuan Y, Xu H, Yang J. A Hybrid Harmony Search Algorithm for the Flexible Job Shop Scheduling Problem [J]. Applied Soft Computing (S1568-4946), 2013, 13(7): 3259-3272.
- [11] KACEM I. Genetic Algorithm for the Flexible Job-shop Scheduling Problem[C]// IEEE International Conference on Systems, Man and Cybernetics, 2003, 4: 3464-3469.
- [12] 仲于江, 杨海成, 莫蓉. 基于小生境粒子群算法的柔性作业车间调度优化方法[J]. 计算机集成制造系统, 2015, 21(12): 3231-3238.
Zhong Yujiang, Yang Haicheng, Mo Rong. Optimization Method of Flexible Job-shop Scheduling Problem Based on Niching and Particle Swarm Optimization Algorithms[J]. Computer Integrated Manufacturing Systems, 2015, 21(12): 3231-3238.
- [13] G Moslehi, M Mahnam. A Pareto Approach to Multi-objective Flexible Job-shop Scheduling Problem using Particle Swarm Optimization and Local Search[J]. International Journal of Production Economics (S0925-5273), 2011, 129(1): 14-22.