文章编号:1671-6833(2024)03-0111-08

改进天牛群算法在柔性作业车间调度中的应用

丁 凯, 赵欣悦, 吕景祥, 朱 斌

(长安大学 智能制造系统研究所,陕西 西安 710064)

摘 要: 为解决柔性作业车间调度问题,在模拟自然界中天牛觅食行为的天牛须算法基础上,结合群智能优化理论,提出了一种基于莱维飞行、反向搜索和自适应参数调整混合策略的改进天牛群算法(LRA-BSO)。首先,建立柔性作业车间调度模型;其次,提出了基于 Tent 混沌映射生成初始种群的方法,以提高初始种群质量;再次,应用莱维飞行策略和反向搜索策略,并通过适应度反馈自适应调整天牛群的搜索步长以及搜索距离,以改善算法全局搜索能力,避免陷入局部极值;最后,为验证改进的天牛群算法的性能,通过 6 个多维度标准测试函数验证了 LRA-BSO 算法的寻优能力。通过 FJSP 的 10 个标准算例和 1 个实际案例验证了 LRA-BSO 算法在 FJSP 中的适用性。测试结果表明:改进的天牛群算法在 8 个标准算例中的表现均优于或持平于其他智能优化算法,表现出了较好的寻优能力;在实际案例验证中,改进后的算法相对于原始的天牛群算法,在收敛速度上提升了 48%。

关键词: 柔性作业车间调度; 天牛群算法; 莱维飞行策略; 反向搜索策略; 自适应参数调整

中图分类号: TH165;TH18 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2024.03.012

为适应不断变化的市场需求和更加多样化的客户需求,传统的大规模制造模式逐渐转变为大规模定制生产模式。在该模式下,制造车间应具备较高的生产柔性,以实现需求驱动的生产过程智能化组织。柔性作业车间调度问题(flexible job shop scheduling problem,FJSP)更符合柔性化和智能化车间生产组织的要求[1]。

求解 FJSP 问题的算法主要涉及精确方法和近似方法。FJSP 属于 NP-hard 问题,是一种典型的离散型组合优化问题,由于离散变量的存在及其问题的复杂性,求解难度大,很难使用精确方法在合理的时间得到满意的结果。随着研究的不断深入,学者们趋向于采用智能优化算法以获得满足约束和目标函数的近似最优解。张朝阳等[2]应用改进的狼群算法对多目标柔性作业车间调度模型进行求解,改进后的算法具有很好的全局搜索能力;王彦杰等[3]提出一种改进的蚁狮优化算法,并将其应用在柔性作业车间调度问题中;Gayathri 等[4]考虑实际生产中存在多种目标组合优化的情况,提出一种改进的混合自适应萤火虫算法求解柔性作业车间调度问题;Meng 等[5]结合了人工蜂群算法和候鸟优化算法的特点,提出一种混合人工蜂群算法求解具有重复操作的柔性作业车间调度问题;杜凌浩等[6]提出了一种改进的多邻域候鸟优化算法,优化柔性作业车间调度问题的最大完工时间。然而,目前的智能优化算法求解 FJSP 时,初始种群的质量会影响算法的求解精度,求解过程中容易陷入局部最优,并且需要调整许多参数才能使算法获得更好的性能。为解决以上智能优化算法存在的缺陷,有必要探索新的算法以获得更优的求解结果。

天牛须搜索算法(beetle antennae search,BAS)是由 Jiang 等[7]在 2017 年提出的一种通过模拟自然界中天牛的觅食行为旨在解决复杂的优化问题的新型智能优化算法。天牛在寻找食物的过程中,通过触须不断感知外界环境中的信息,通过计算左右两侧的气味浓度差异选择移动方向。天牛不断向着气味浓度更高的方向移动,最终天牛能够准确寻找到食物的位置[8]。该算法具有搜索能力强、收敛速度快等特点,目前在很多领域的优化问题中得到应用。Fan 等[9]提出了一种结合 BAS 算法和 PID 策略的

收稿日期:2023-10-23;**修订日期**:2023-11-20

基金项目:国家自然科学基金资助项目(51705030);中国博士后科学基金特别资助项目(2022T150073)

作者简介:丁凯(1989—),男,江苏淮安人,长安大学教授,博士,主要从事制造系统智能化研究,E-mail:kding@chd.edu.cn。

引用本文:丁凯,赵欣悦,吕景祥,等. 改进天牛群算法在柔性作业车间调度中的应用[J]. 郑州大学学报(工学版),2024,45(3):111-118. (DING K, ZHAO X Y, LYU J X, et al. Improved beetle swarm optimization algorithm for flexible job-shop scheduling[J]. Journal of Zhengzhou University (Engineering Science),2024,45(3):111-118.)

复合PID控制器,并将其用于驱动电液伺服系统的位置伺服控制系统;Kim等[10]提出了一种基于混合BAS算法的Gabor滤波检测织物表面缺陷的方法,并通过实验验证该方法具有良好的稳定性和鲁棒性。

尽管天牛须搜索算法代码简单、复杂度低,但是由于寻优过程中只考虑单个天牛的搜索行为,算法的单一性使其容易陷入局部极值。Wang等[11]受群优化算法的启发,对天牛须搜索算法进行改进,将个体扩展为群体,提出天牛群算法(beetle swarm optimization,BSO),提高了优化性能,使该算法具备一定跳出局部极值的能力;Bhagavathi等[12]提出一种基于混合的BAS-PSO算法,并应用于云计算中虚拟机的整合和迁移;Mu等[13]提出了一种基于天牛群算法的三维路线规划方法。

天牛群算法目前常应用于连续函数优化问题,对求解FJSP等离散优化问题的研究较少,求解过程可能陷入局部极值,尚缺少应用验证。

本文提出一种基于莱维飞行、反向搜索和自适应参数调整混合策略的改进天牛群算法(hybrid Levy flight, reverse search, and adaptive parameter adjustment strategy improved beetle swarm optimization algorithm, LRA-BSO)来求解FJSP问题,改进思路如下:由于群智能算法对初始种群质量有要求,本文在种群初始化时采用Tent混沌序列代替随机序列,以提高初始种群质量;在迭代优化过程中,采用基于莱维飞行及反向搜索的优化策略,以改善算法的全局搜索能力,逃离局部极值;BSO算法较天牛须搜索算法有更多参数,为减少前期参数设置对算法性能的影响,采用基于适应度反馈的认知因子及步长更新策略。在此基础上,采用6种测试函数对本文提出的改进策略进行有效性验证,并应用10个标准算例和1个实际案例验证该算法在求解FJSP问题时的可行性和有效性。

1 FJSP问题建模

1.1 问题描述

FJSP问题最早由Brucker等[14]在1990年提出,该问题具有灵活性和复杂性,更符合实际的生产场景。在FJSP问题中,假设有n个工件在m台机器上加工,其中每个工件包含多道工序,每道工序可以在不同机器上加工处理,不同的机器上的加工时间可能不同。该问题主要涉及以下两个子问题:①按什么顺序加工工件;②如何合理地分配机器。因此,解决FJSP问题就是对每道工序选择合适的机器进行加工,并合理地为每一道工序安排机器进行处理,使完工时间最少。

本文以最大完工时间最小化为优化目标,在建立模型之前,做出以下假设:①机器之间是相互独立的;②在零时刻,所有机器和工件可加工;③同一工件的工序间存在优先约束,而不同工件之间是相互独立的;④同一时刻一台机器只能加工一个工件;⑤不考虑机器设置时间和工件运输时间;⑥加工具有不可中断性。

1.2 相关符号说明

本文模型中的相关参数符号和定义如表1所示。

表1 模型参数符号和定义

Table 1 Parameter symbols and definitions of model

符号	定义
$I=\{i_1,i_2,i_3,\cdots,i_n\}$	模型中所有的工件集合
$M=\{M_1,M_2,\cdots,M_m\}$	模型中所有的设备集合
$o_i=\{o_{i1},o_{i2},o_{i3},\cdots,o_{iq}\}$	工件i工序集合
o_{ij}	工件i的第j道工序
$M_{ij}=\{M_1,M_2,\cdots,M_k\}$	工序o_{ij}的机器集合
s_{ijk}	工序o_{ij}在机器M_k上的开工时间
e_{ijk}	工序o_{ij}在机器M_k上的完工时间
t_{ijk}	工序o_{ij}在机器M_k上的处理时间
c_i	工件i的完工时间,$1\leqslant i\leqslant n$
c_{ij}	工序o_{ij}的完工时间
x_{ijk}	决策变量,若工序o_{ij}在机器M_k上加工则为1,反之则为0

1.3 模型建立

基于问题描述,建立FJSP问题模型如下:

$$f=\min(\max c_i); \tag{1}$$

$$\sum_{k=1}^{m}x_{ijk}=1; \tag{2}$$

$$c_{ij}\geqslant s_{ij}+\sum_{k=1}^{m}x_{ijk}\cdot t_{ijk}; \tag{3}$$

$$s_{i(j+1)}\geqslant e_{ij}; \tag{4}$$

$$s_{ijk}\geqslant e_{hgk}。 \tag{5}$$

其中,式(1)表示优化目标;式(2)表示每道工序只能在一台机器上进行一次加工;式(3)表示加工工件i的第j道工序的完工时间应该大于等于其在机器M_k上的开始时间与加工时间之和;式(4)表示同一工件的相邻工序必须按照工序的优先约束依次加工;式(5)表示在同一台机器上加工的工序必须按照机器的工序顺序依次进行,工序o_{hg}为工序o_{ij}在机器M_k上加工的前一道工序。

2 LRA-BSO算法求解

2.1 算法流程

本文所提出的LRA-BSO算法求解FJSP问题的

流程如图 1 所示,具体步骤如下。

步骤 1　读取车间加工数据,设置算法参数。

步骤 2　设置种群编码与解码方式,采用 Tent 混沌序列代替随机序列,初始化调度解种群。

步骤 3　计算每个天牛个体的适应度值,并根据天牛个体左右须适应度大小判断方向。

步骤 4　采用莱维飞行和反向搜索策略进行种群迭代搜索,寻找最优种群个体。

步骤 5　更新天牛个体的速度和位置并计算适应度值。

步骤 6　更新天牛个体最优位置,并采用精英策略合成新的种群。

步骤 7　判断算法是否达到终止条件,若达到,则得到最佳调度方案,算法结束;若未达到,则对算法中的权重、搜索直径、搜索步长、认知参数等参数进行动态调整,并返回步骤 3 继续迭代,直至算法结束。

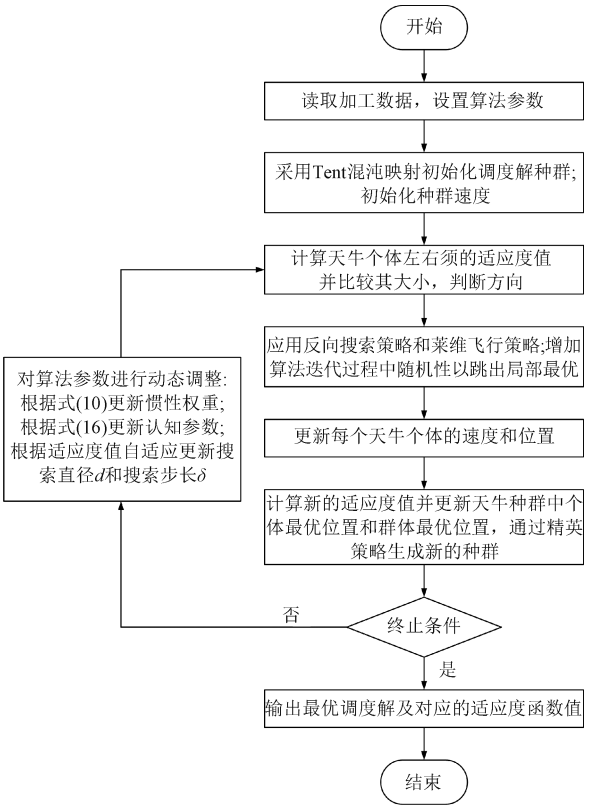

图 1　LRA-BSO 算法求解 FJSP 流程图

Figure 1　Flowchart of LRA-BSO for solving FJSP

2.2　编码与解码

2.2.1　*编码*

求解 FJSP 问题涉及工序排序和机器选择两个子问题,因此本文采用了基于工序选择和机器选择的编码方法。编码方式如表 2 所示。表 2 中数字 1-1 代表工件 1 的第 1 道工序;第 2 列表示对应工序的机器选择编码,由于柔性作业车间中工件的每道工序都有不止一台机器可以选择,因此机器选择编码中的 M_k 表示对应的工序编码可选加工机器集合中的第 k 台机器。

表 2　编码方式示例

Table 2　Coding method example

工序	机器	工序	机器
1-1	M_1	3-1	M_2
2-1	M_2	3-2	M_1
1-2	M_3		

注:数字 1-1 表示第 1 个工件的第 1 道工序,机器 M_1 表示此道工序选择加工机器 M_1。

2.2.2　*解码*

种群解码是指按照不同的调度规则解码生成可行的调度方案。在解码过程中,首先根据工序选择编码确定每道工序在调度方案中的加工顺序,然后根据机器选择编码确定每道工序的机器选择。

在解码过程中,本文采用贪婪插入的方式,按照工序序列从左到右依次进行排序,如算法 1 所示。

算法 1　贪婪插入。

输入:一个可行解;

输出:贪婪插入后的可行解。

① For i = 1 to *total_op_num* do

② 获取当前工序 o_i 所在机器 M 的空闲时间段 $B=\{b_1,b_2,\cdots,b_n\}$;

③ For j = 1 to n do

④ if o_i的开始时间≤空闲时间 b_j;

o_i的开始时间=b_j的开始时间;

o_i 的完成时间=b_j 的开始时间+o_i的加工时间;

end if

⑤ end

⑥ end

2.3　基于 Tent 混沌映射的种群初始化

种群初始化策略影响天牛群算法的求解效率和求解质量。对于 FJSP 问题而言,种群初始化主要包括工序序列初始化和机器序列初始化。

(1)机器序列初始化。机器序列的初始化方案包括全局选择、局部选择和随机选择,本文初始化种群时 3 种生成规则所占比例分别设置为 0.6、0.3、0.1。这种混合策略生成初始种群既保障了调度过程中负载的均匀性又保证了种群的多样性。

(2)工序序列初始化。在大多数研究中,工序序列初始化一般采用随机生成的方式。在生成群智能优化算法的初始种群时,使用 Tent 混沌映射可以提高种群的多样性,有助于提高算法的搜索能力[15]。本文在生成工序序列的过程中采用 Tent 混

沌序列代替原本的随机序列。

Tent 混沌序列的表达式如下:

$$x(t+1)=\begin{cases}\dfrac{x(t)}{a},0\leqslant x(t)<a;\\ \dfrac{1-x(t)}{1-a},a\leqslant x(t)<1。\end{cases}\tag{6}$$

式中:a 一般取值为[0,1],经过多次实验验证,本文选择 $a=0.499$。

2.4 基于反向搜索和莱维飞行改善全局搜索能力

BSO 算法在天牛须搜索算法中引入群体的概念,能够在一定程度上跳出局部最优,但是由于 FJSP 属于 NP-hard 问题,采用智能优化算法求解时,容易陷入局部极值。本文基于反向搜索和莱维飞行策略来改进 BSO 算法,提升全局搜索能力。

BSO 算法中,天牛个体的数量即为种群的规模。每个天牛个体都有自己的位置和速度,在每次迭代中,天牛个体会朝着个体的最优解移动,所有天牛个体朝着种群最优的天牛个体的方向移动,直至达到最大迭代次数。天牛个体的速度和位置更新公式如下:

$$v_{ij}(t+1)=\omega v_{ij}(t)+c_1r_1[(p_{ij})_j(t)-x_{ij}(t)]+c_2r_2[(p_{ij})_j(t)-x_{ij}(t)]+c_3r_3\xi(t);\tag{7}$$

$$\xi(t+1)=\delta(t)\cdot\vec{\boldsymbol{b}}\cdot\operatorname{sign}(f(x_{\text{right}}(t))-f(x_{\text{left}}(t)));\tag{8}$$

$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1)。\tag{9}$$

式中:c_1、c_2 两个参数同粒子群算法中的加速因子,一般取[0,2];c_3 为认知因子;$\xi(t)$ 为认知方向;r_1、r_2、r_3 为[0,1]上的随机数;ω 为惯性权重。

ω 的更新方式为

$$\omega=\omega_{\max}-\frac{\omega_{\max}-\omega_{\min}}{T}\times t。\tag{10}$$

式中:T 为最大迭代次数;t 为当前迭代次数;$\omega_{\max}$、$\omega_{\min}$ 分别表示惯性权重的最大值和最小值,本文中 $\omega_{\max}=0.9$,$\omega_{\min}=0.4$,这样设置参数在迭代开始时算法具有更大的搜索范围,随着迭代次数增加,ω 逐渐减少,天牛速度降低,算法进行局部搜索。

2.4.1 反向搜索策略

群智能算法迭代过程中,种群多样性通常会随着迭代次数的增加而逐渐减少[16],导致算法陷入局部最优而无法找到全局最优解。为了解决这个问题,通过反向搜索策略引入随机扰动,以增加种群的多样性。其计算公式为

$$y=\begin{cases}-1,\text{rand}<0.5;\\ 1,\text{rand}>0.5。\end{cases}\tag{11}$$

2.4.2 莱维飞行策略

莱维飞行是一种随机搜索路径,每次移动的步长服从莱维分布,大部分情况下生成较小的步长,但是会以较小的概率出现较大步长的跳跃[17]。利用莱维飞行策略的随机性调整天牛群个体的位置,能够有效帮助 BSO 算法提高寻优能力[18]。莱维飞行的步长符合莱维分布:

$$s=\frac{\mu}{|v|^{\frac{1}{\beta}}};\tag{12}$$

$$\sigma_\mu=\left\{\frac{\Gamma(1+\beta)\times\sin\dfrac{\pi\beta}{2}}{\Gamma\left(\dfrac{1+\beta}{2}\right)\times\beta\times2^{\frac{\beta-1}{2}}}\right\}^{\frac{1}{\beta}};\tag{13}$$

$$\sigma_v=1。\tag{14}$$

式中:$\mu\sim N(0,\sigma_\mu^2)$;$v\sim N(0,\sigma_v^2)$;$\beta$ 通常取 1.5。

通过莱维飞行策略和反向搜索策略改善算法的全局搜索能力,天牛个体的位置更新公式为

$$x_{ij}(t+1)=y\times(x_{ij}(t)+x_{ij}(t)\oplus Levy)+v_{ij}(t+1)。\tag{15}$$

2.5 基于适应度值反馈的算法参数调整策略

2.5.1 认知因子的调整策略

天牛个体速度更新公式(式(7))中,$c_3r_3\xi(t)$ 表示天牛群速度更新的个体认知,认知因子 c_3 和认知方向 $\xi(t)$ 与天牛个体左右两须所在方向的适应度大小有关[19]。认知因子 c_3 的大小决定了个体在搜索过程中是更倾向于全局最优位置还是局部最优位置。认知因子越大,个体越容易跳出局部极值,但收敛速度可能会受到影响;反之,个体更可能陷入局部极值。本文对认知因子进行调整以兼顾算法的搜索能力,更新公式如下:

$$c_3=\begin{cases}c_3-\dfrac{(c_{\max}-c_{\min})(f-f_{\min})}{f_{\max}-f_{\text{avg}}},f\geqslant f_{\text{avg}};\\ c_3,f<f_{\text{avg}}。\end{cases}\tag{16}$$

式中:$c_{\max}$、$c_{\min}$ 分别为认知因子取值的最大值和最小值;f、$f_{\min}$、$f_{\max}$、f_{avg} 分别为当前天牛个体的适应度值、天牛群最小适应度值、天牛群最大适应度值、天牛群平均适应度值。

若此时天牛个体的适应度不如当前群体的平均适应度,则认为该天牛个体所处位置性能不佳,应该对其采用更大的认知因子来增加其全局搜索能力;反之,减小其认知因子以保证个体能够在其位置周围精细搜索,更快地收敛到全局最优解。

2.5.2 搜索步长和搜索距离的调整策略

由于 LRA-BSO 算法参数多,其性能和初始的搜

索步长、天牛的搜索距离以及他们对应的衰减系数有关[20],合理调整搜索步长是帮助算法逃离局部最优的可行方法。

若 BSO 算法步长衰减过快,容易使算法提前收敛,陷入局部最优解。本文引入基于适应度值反馈的步长进行调整策略:如果适应度较好,则步长不更新;反之,若当前步长不能提高适应度值,则调整搜索步长。步长更新方法如算法 2 所示。

算法 2　更新搜索步长和搜索距离。

输入:当前天牛个体的适应度 f、当前天牛群的最佳适应度值 f_{best}、$\delta(t)$、$d(t)$;

输出:更新后的搜索步长 $\delta(t+1)$、搜索距离 $d(t+1)$。

① if　$f < f_{\text{best}}$　// 当前 δ 和 d 不能提高适应度

$$\delta(t+1) = \eta_\delta \times \delta(t);$$

$$d(t+1) = \eta_d \times d(t);$$

② else

$$\delta(t+1) = \delta(t);$$

$$d(t+1) = d(t);$$

③ end

3　算法测试

本文选取 6 种广泛应用于智能优化算法性能测试的标准测试函数。通过对这些函数进行测试并比较算法得到的最优解和理论最优解,对所提出的 LRA-BSO 算法本身的性能进行评估。

如表 3 所示,6 种测试函数中,f_1 和 f_2 为单峰测试函数,f_3、f_4 和 f_5 为多峰测试函数,f_6 为固定峰值测试函数。对于相同的给定条件,如果得到的解接近或达到理论解,则可以认为该算法具有更强的寻优能力和更高的求解精度。

算法运行环境为 Win10 操作系统、8 GB 内存、Intel Core(TM) i7-1165G7 CPU 2.80 GHz。考虑到 LRA-BSO 算法是基于 BSO 改进的一种群智能算法,而 BSO 则是在 BAS 的基础上引入粒子群概念而提出的群智能算法,本文将 LRA-BSO 与 BSO、粒子群优化算法(PSO)进行比较,各算法的种群规模均设置为 50,最大迭代次数 $T=1\ 000$。LRA-BSO 算法参数设置为 $\omega_{\min}=0.4$,$\omega_{\max}=0.9$,$c_1=c_2=1.79$,$c_{3\max}=2$,$c_{3\min}=1$。

表 4 给出了 30 次独立重复实验下 3 种算法对给出的测试函数的求解结果对比,其中不同函数在 3 种算法中的最优结果已加粗,其中 BSO、PSO 的计算结果参见文献[11]。

由表 4 可以看出,在 6 种基准函数测试实验中,LRA-BSO 算法得到了 4 种测试函数的最优结果,其中 f_4、f_5、f_6 均达到了测试函数的理论最优值。在处理单峰测试函数 f_1、f_2 时,LRA-BSO 与 BSO、PSO 算法相比在收敛能力上还有差距,但是 LRA-BSO 算法的性能更加稳定。在处理多峰值测试函数 f_3、f_4、f_5 时,LRA-BSO 算法的性能明显优于其他算法。实验结果表明,本文提出的改进策略使得 LRA-BSO 算法能够有效避免陷入局部最优解。在处理固定峰函数 f_6 时,LRA-BSO 算法不仅达到了理论最优值,而且与其他 2 种算法相比,LRA-BSO 算法的性能也更加稳定。

综合考虑 3 种不同类型的测试函数,本文提出的 LRA-BSO 算法具有较好收敛精度和较强的稳定性,能够跳出局部最优,避免早熟。该算法具有较强的寻优能力,求解复杂 FJSP 问题具有优势。

4　案例验证

采用 Brandimarte 基准算例[21](Mk01 ~ Mk10)

表 3　测试函数

Table 3　Test function

测试函数方程	维度	变量搜索范围	理论最优解
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[-100,100]	0
$f_3(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500,500]	-1.256 9e+04
$f_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$f_5(x) = \frac{1}{4\ 000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$f_6(x) = -\sum_{i=1}^{10} [(x-a_i)(x-a_i)^T + c_i]^{-1}$	30	[0,10]	-1.053 6e+01

表 4 6 种测试函数的实验结果比较

Table 4 Comparison of experimental results of 6 test functions

算法	f_1			f_2			f_3		
	aver	*best*	*std*	*aver*	*best*	*std*	*aver*	*best*	*std*
LRA-BSO	9.593 6e-08	1.820 1e-14	2.815 6e-07	3.454 8e-08	8.185 6e-14	5.478 3e-08	**1.088 6e+04**	**-1.240 2e+04**	8.192 0e+02
BSO	**0**		9.360 0e-76	**0**		3.310 0e-72	-1.790 0e+03		1.733 5e+02
PSO	**0**		0	1.670 0e+02		9.128 7e+02	-1.400 0e+03		8.574 8 e+01
算法	f_4			f_5			f_6		
	aver	*best*	*std*	*aver*	*best*	*std*	*aver*	*best*	*std*
LRA-BSO	**2.294 4e-10**	**0**	8.514 0e-10	**3.355 0e-09**	**0**	1.760 9e-08	**-1.053 6e+01**	**-1.053 6e+01**	1.260 0e-02
BSO	4.311 0e-01		9.305 0e-01	1.267 0e-01		8.490 0e-02	-9.106 9e+00		2.411 1e+00
PSO	5.178 5e+00		9.005 7e+00	1.348 0e-01		9.260 0e-02	-1.216 1e+00		6.276 0e-01

及 1 个实际案例[2]对本文提出的 LRA-BSO 算法在 FJSP 工程问题中的应用进行有效性验证。经过多次参数实验,设置 LRA-BSO 算法参数如下:种群规模为 50;最大迭代次数 $T=1\ 000$;$\omega_{\min}=0.4$;$\omega_{\max}=0.9$;$c_1=c_2=1.79$;$c_{3\max}=2$;$c_{3\min}=1$。

4.1 标准算例验证

LRA-BSO、BSO、PSO、混合灰狼优化算法(HGWO)[22]、改进蚁狮优化算法(IALO)[3]、改进多邻域候鸟优化算法(IMMBO)[6]等群智能优化算法求解 Mk01~Mk10 的结果如表 5 所示。

表 5 Brandimarte 标准算例对比

Table 5 Comparison of Brandimarte benchmark studies

算法	Mk01	Mk02	Mk03	Mk04	Mk05	Mk06	Mk07	Mk08	Mk09	Mk10
LRA-BSO	**40**	**28**	**204**	**62**	176	**70**	**144**	**523**	**312**	237
BSO	**40**	30	**204**	67	182	74	155	**523**	321	244
PSO	46	35	212	71	185	98	176	557	345	251
HGWO	**40**	29	**204**	65	175	79	149	**523**	325	253
IALO	**40**	**28**	**204**	67	174	73	149	**523**	323	**234**
IMMBO	**40**	**28**	**204**	66	173	72	**144**	**523**	325	257

由表 5 可知,相比于其他算法,除了求解 Mk05、Mk10 算例的结果略差,LRA-BSO 算法在求解其余 8 个算例时,结果均优于或持平于其他算法。因此,该算法在解决 FJSP 问题方面具有优势。

4.2 实例验证

采用文献[2]中的 FJSP 实例对本文提出的 LRA-BSO 算法进行有效性验证,并与其结果进行对比。在该实例中,6 个工件在 6 台机器上加工,分别采用 LRA-BSO、BSO、PSO 以及改进狼群算法[2]分别对该实例进行求解,求解结果如表 6 所示。由表 6 可知,LRA-BSO 算法求解该实例的最大完工时间为 35,优于其他算法。

对比 LRA-BSO、BSO 和 PSO 算法,图 2 给出了 LRA-BSO、BSO、PSO 这 3 种算法在求解该实例时的迭代曲线。由图 2 可知,在寻优能力上,LRA-BSO 算法优于 BSO 算法和 PSO 算法;在收敛速度上,LRA-BSO 算法在 73 代收敛,优于 BSO 算法但次于 PSO 算法。虽然 PSO 算法在 15 代收敛,但该算法寻优能力不足,易陷入局部最优,表 6 显示其最大完工时间为 38,次于 LRA-BSO 算法结果。

综上所述,求解 FJSP 实例问题时,本文提出的 LRA-BSO 算法在寻优能力和收敛速度上具有一定的综合优势。对应的甘特图如图 3 所示。

表 6 实例对比结果

Table 6 Compared results of the instance

算法	最大完工时间
LRA-BSO 算法	35
BSO 算法	37
PSO 算法	38
改进狼群算法[2]	37

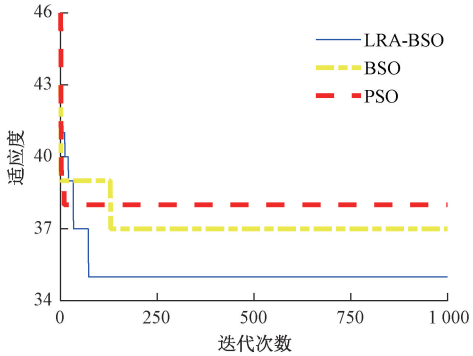

图 2 不同算法迭代曲线

Figure 2 Iteration curves of different algorithms

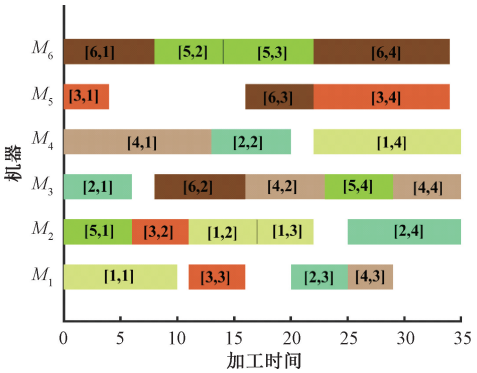

图 3　基于 LRA-BSO 算法的 FJSP 甘特图

Figure 3　Gantt chart for LRA-BSO based FJSP

5　结论

针对柔性作业车间调度问题,提出了一种基于莱维飞行、反向搜索和自适应参数调整混合策略的改进天牛群算法。改进后的算法在初始种群的质量上有所提升,同时提高了算法的全局搜索能力,能够有效避免陷入局部最优解,可用于解决 FJSP 等复杂度比较高的离散型组合优化问题。通过 6 个多维度标准测试函数对改进后的算法本身的性能进行测试。结果表明,LRA-BSO 算法具有较好的收敛精度和较强的稳定性。通过 10 个 FJSP 标准算例及 1 个实例验证了 LRA-BSO 算法在求解 FJSP 问题上的可行性和有效性。实验结果表明,在求解 FJSP 实例问题时,LRA-BSO 算法在寻优能力和收敛速度上具有较好的综合性能。

参考文献:

[1] 吴秀丽, 张志强. 求解柔性作业车间调度问题的细菌算法对比及改进[J]. 郑州大学学报(工学版), 2018, 39(3): 34-39.
WU X L, ZHANG Z Q. The comparison and improvement of bacterial algorithms for flexible job scheduling problem[J]. Journal of Zhengzhou University (Engineering Science), 2018, 39(3): 34-39.

[2] 张朝阳, 徐莉萍, 李健, 等. 基于改进狼群算法的柔性作业车间调度研究[J]. 系统仿真学报, 2023, 35(3): 534-543.
ZHANG C Y, XU L P, LI J, et al. Flexible job-shop scheduling problem based on improved wolf pack algorithm[J]. Journal of System Simulation, 2023, 35(3): 534-543.

[3] 王彦杰, 向凤红. 基于改进蚁狮优化算法的柔性作业车间调度研究[J]. 机电工程, 2022, 39(9): 1325-1332.
WANG Y J, XIANG F H. Flexible job shop scheduling based on improved ant lion algorithm[J]. Journal of Mechanical & Electrical Engineering, 2022, 39(9): 1325-1332.

[4] GAYATHRI D K, MISHRA R S, MADAN A K. A dynamic adaptive firefly algorithm for flexible job shop scheduling[J]. Intelligent Automation & Soft Computing, 2022, 31(1): 429-448.

[5] MENG T, PAN Q K, SANG H Y. A hybrid artificial bee colony algorithm for a flexible job shop scheduling problem with overlapping in operations[J]. International Journal of Production Research, 2018, 56(16): 5278-5292.

[6] 杜凌浩, 向凤红. 改进多邻域候鸟优化算法的柔性作业车间调度研究[J]. 兵器装备工程学报, 2022, 43(12): 299-306.
DU L H, XIANG F H. Research on flexible job shop scheduling based on improved multi-neighborhood migratory bird optimization algorithm[J]. Journal of Ordnance Equipment Engineering, 2022, 43(12): 299-306.

[7] JIANG X Y, LI S. BAS: beetle antennae search algorithm for optimization problems[J]. International Journal of Robotics and Control, 2018, 1(1): 1.

[8] JIANG Y F, WANG S Q, LI Y C. Localizing and quantifying structural damage by means of a beetle swarm optimization algorithm[J]. Advances in Structural Engineering, 2021, 24(2): 370-384.

[9] FAN Y Q, SHAO J P, SUN G T. Optimized PID controller based on beetle antennae search algorithm for electro-hydraulic position servo control system[J]. Sensors, 2019, 19(12): 2727.

[10] KIM J, JO H, RI J, et al. Automatic fabric defect detection using optimal Gabor filter based on hybrid beetle antennae search-gravitational search algorithm[J]. Journal of Optics, 2023,52(4): 1-9.

[11] WANG T T, YANG L. Beetle swarm optimization algorithm: theory and application[EB/OL]. (2018-08-01)[2023-10-12]. https://arxiv.org/abs/1808.00206.

[12] BHAGAVATHI H, RATHINAVELAYATHAM S, SHANMUGAIAH K, et al. Improved beetle swarm optimization algorithm for energy efficient virtual machine consolidation on cloud environment[J]. Concurrency and Computation: Practice and Experience, 2022, 34(10): 1-15.

[13] MU Y Z, LI B K, AN D, et al. Three-dimensional route planning based on the beetle swarm optimization algorithm[J]. IEEE Access, 2019, 7: 117804-117813.

[14] BRUCKER P, SCHLIE R. Job-shop scheduling with multi-purpose machines[J]. Computing, 1990, 45(4): 369-375.

[15] 刘倩, 冯艳红, 陈嶷瑛. 基于混沌初始化和高斯变异的飞蛾火焰优化算法[J]. 郑州大学学报(工学版),

2021, 42(3): 53-58.
LIU Q, FENG Y H, CHEN Y Y. Moth-flame optimization algorithm based on chaotic initialization and Gaussian mutation[J]. Journal of Zhengzhou University (Engineering Science), 2021, 42(3): 53-58.

[16] 徐霜, 万强, 余琍. 基于学习理论的改进粒子群优化算法[J]. 郑州大学学报(工学版), 2019, 40(2): 29-34.
XU S, WAN Q, YU L. An improved particle swarm optimization algorithm based on learning theory[J]. Journal of Zhengzhou University (Engineering Science), 2019, 40(2): 29-34.

[17] ALI M Z, AWAD N H, REYNOLDS R G, et al. A balanced fuzzy cultural algorithm with a modified Levy flight search for real parameter optimization[J]. Information Sciences, 2018, 447: 12-35.

[18] XU X, DENG K L, SHEN B. A beetle antennae search algorithm based on Lévy flights and adaptive strategy[J]. Systems Science & Control Engineering, 2020, 8(1): 35-47.

[19] 赵玉强, 钱谦. 一类带学习与竞技策略的混沌天牛群搜索算法[J]. 通信技术, 2018, 51(11): 2582-2588.
ZHAO Y Q, QIAN Q. Novel chaos beetle swarm searching algorithm with learning and competitive strategies[J]. Communications Technology, 2018, 51(11): 2582-2588.

[20] WANG J Y, CHEN H X. BSAS: beetle swarm antennae search algorithm for optimization problems[EB/OL]. (2018-07-27)[2023-10-12]. https://arxiv.org/abs/1807.10470.

[21] BRANDIMARTE P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3): 157-183.

[22] 姜天华. 混合灰狼优化算法求解柔性作业车间调度问题[J]. 控制与决策, 2018, 33(3): 503-508.
JIANG T H. Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm[J]. Control and Decision, 2018, 33(3): 503-508.

Improved Beetle Swarm Optimization Algorithm for Flexible Job-shop Scheduling

DING Kai, ZHAO Xinyue, LYU Jingxiang, ZHU Bin

(Institute of Smart Manufacturing Systems, Chang'an University, Xi'an 710064, China)

Abstract: To solve the flexible job shop scheduling problem(FJSP), a hybrid Levy flight, reverse search, and parameter adaptive adjustment strategy improved beetle swarm optimization (LRA-BSO) was proposed based on the beetle antennae search algorithm which could simulate the foraging behavior of beetles in nature and the swarm intelligence optimization theory. Firstly, a FJSP model was established. Secondly, the initial population was generated based on the Tent chaotic mapping, which would improve the quality of the initial population. Then, the Levy flight strategy and reverse search strategy were used to improve the global search ability of the LRA-BSO algorithm, and the search step size and the search distance of the beetle swarm were adjusted through fitness feedback to avoid falling into local optimum. Finally, the optimization ability of the algorithm was validated through 6 multi-dimensional standard test functions. In addition, the applicability of the LRA-BSO algorithm in FJSP was verified by 10 standard test cases and 1 practical case. The test results showed that the algorithm performed better or equal to other intelligent optimization algorithms in eight standard test cases and demonstrated good optimization ability. In the practical cases, the improved algorithm had a 48% improvement in convergence speed compared to the original beetle swarm optimization algorithm.

Keywords: flexible job-shop scheduling; beetle swarm optimization; Levy flight; reverse search; adaptive parameter adjustment