管理科学与工程

文章编号：2095-1248(2025)06-0089-08

改进细菌觅食算法在柔性作业车间调度问题中的应用

王晓燕，王帅文，王韬亮

（沈阳航空航天大学 经济与管理学院，沈阳 110136）

摘要：为优化柔性作业车间调度、缩短制造企业的生产周期，以最小化最大完工时间为目标建立了优化模型，并采用细菌觅食算法求解。针对细菌觅食算法存在的易早熟收敛等缺陷进行一系列改进，设计Logistic-Circle混合映射提高初始种群质量；设计自适应动态步长提高寻优效率及精度；该复制操作避免早熟收敛；设计自适应迁徙概率避免精英个体“逃逸”。通过算例测试验证了该算法在柔性作业车间调度问题中缩短生产周期的有效性。

关键词：柔性作业车间调度；最大完工时间；混沌映射；自适应；改进细菌觅食算法

中图分类号：TP18；TH165 **文献标志码：**A

DOI：10.3969/j.issn.2095-1248.2025.06.011

Application of improved bacterial foraging algorithm in flexible job shop scheduling problem

WANG Xiaoyan，WANG Shuaiwen，WANG Taoliang

（College of Economics and Management，Shenyang Aerospace University，Shenyang 110136，China）

Abstract：To optimize flexible job shop scheduling and shorten the production cycle of manufacturing enterprises，an optimization model was established with the objective of minimizing the makespan，and the bacterial foraging algorithm was used to solve it. In order to solve the defects of the bacterial foraging algorithm，such as prone to premature convergence，a series of improvements were made. The Logistic-Circle chaos mapping was designed and introduced to enhance the quality of the initial population. An adaptive dynamic step size was designed to ensure the optimization efficiency and accuracy，which improved the replication operation to avoid premature convergence. An adaptive migration probability was designed to avoid the “escape” of elite individuals. The effectiveness of the improved algorithm in shortening the production cycle in the flexible job shop scheduling problem was verified through the example solution.

Key words：flexible job shop scheduling；makespan；chaos mapping；adaptive；improved bacterial foraging algorithm

收稿日期：2024-12-05 **修回日期：**2025-03-26 **接受日期：**2025-03-27

基金项目：辽宁省教育厅面上项目（项目编号：JYT2020103）。

作者简介：王晓燕（1975—），女，辽宁沈阳人，教授，博士，主要研究方向为精益管理，E-mail：Wlfn2005@163.com。

随着云计算、人工智能等信息技术的发展与成熟，制造行业由制造向“智”造升级，需求向个性化、定制化转变。为了应对需求定制化带来的生产压力及制造智能化引发的激烈竞争，制造企业必须进一步优化作业车间调度以高效规划资源，缩短生产周期。作业车间调度是指在具有一组功能不同的机床加工系统中，将待加工工件的各工序安排到各机器，并安排各工序的加工次序及加工开始时间的过程。由于机器柔性，传统的作业车间调度已不符合当下快节奏的生产模式，柔性作业车间调度应运而生[1]。与传统作业车间调度相比，在柔性作业车间调度中，至少有一道工序不受限于机器约束，调度更为复杂、灵活，更具研究价值与实践意义[2]。

针对柔性作业车间调度优化的算法研究是该领域的研究热点，米恬怡等[3]针对柔性作业车间调度问题，提出融合了强化学习的变邻域搜索算法，该方法利用强化学习实现了参数的动态调整，进而提升了算法的寻优效率及稳定性。Tutumlu等[4]研究了考虑作业拆分的柔性作业车间调度问题，并提出了结合作业拆分的混合遗传算法，该方法求得了小规模问题的最优解与大规模问题的可行解。Akram等[5]提出了多目标动态绿色柔性作业车间调度问题，并通过模拟黑寡妇蜘蛛的行为，提出多目标黑寡妇蜘蛛算法，实现了生产周期与能耗的平衡。现有的优化算法各具优势，但仍存在一定缺点，如只能应用于特定问题、实现简单但精度较低、复杂度高、不易应用等。因此，虽然各类优化算法被相继提出，设计简单高效、稳定精确的求解算法仍是柔性作业车间调度领域的重要研究方向[6]。基于此，拟优选细菌觅食算法并将其拓展应用于柔性作业车间调度优化问题。

细菌觅食算法是一种群体智能算法，具有简单易实现、并行搜索等优点，已广泛应用于解决图像增强等优化问题。然而，求解实际问题时，细菌觅食算法仍存在无法兼顾效率与精度、易早熟收敛、精英个体易“逃逸”等问题，因此近年来已有许多改进研究。牛奔等[7]在周期性细菌觅食算法中引入了多元学习策略，改进了算法的趋向性操作及复制操作，在加快收敛速度的同时保证了种群多样性。Pang等[8]在细菌觅食算法的趋向性操作中引入了自适应搜索步长，减少了个体不必要的翻滚与游动行为，提升了寻优效率且避免了算法陷入局部最优。李冬琴等[9]在细菌觅食算法的趋向性操作中引入了自适应动态步长及粒子群算法的记忆步骤，提升了算法的寻优效率及全局搜索能力。已有的细菌觅食算法改进方案显著提升了该算法的寻优性能，但并未解决细菌觅食算法存在的全部问题，如文献[10]的改进方案仅引入了自适应搜索步长，并未解决固定迁徙概率导致的精英个体“逃逸”问题。

本文提出一种针对细菌觅食算法全流程的方案，解决该算法现有问题，提升算法性能，并将其应用于柔性作业车间调度优化问题，缩短制造企业生产周期。改进的细菌觅食算法称为改进细菌觅食算法，该算法通过两段式个体编码实现连续位置与离散调度解的映射，设计并引入Logistic-Circle混合映射优化初始种群，设计自适应动态步长并保证算法的求解效率与精度，改进复制操作保证种群多样性，避免算法陷入早熟收敛，设计自适应迁徙概率避免精英个体“逃逸”。通过对大量算例进行仿真测试，并与细菌觅食算法、粒子群算法等算法进行比较，测试了改进算法的性能，验证了其有效性。

1 柔性作业车间调度问题模型

1.1 问题描述

柔性作业车间调度问题（flexible job shop scheduling problem，FJSP）可以描述为：在某柔性作业车间中将m个工件分配给n台机器进行加工；每个工件的加工需经过确定顺序的多道

工序；每道工序有至少 1 台、至多 n 台可用机器，不同的机器加工时间不同；每台机器同一时刻只能加工一道工序；调度的目的即为确定各个工序的加工机器及机器上各工序的加工顺序，以高效完成生产计划。柔性作业车间调度问题的确定需要满足以下假设：1）所有的工件及机器在初始时刻均已准备就绪，即可以立刻开始加工；2）在同一时刻，同一工件只能在一台机器上进行加工；3）在同一时刻，同一机器只能加工一道工序；4）加工作业必须连续进行，不允许中断；5）所有工件均具有相同的加工优先级；6）同一工件需按照工序进行加工，不同工件的工序之间不存在顺序约束。

1.2　问题模型

问题的优化目标为最大完工时间。最大完工时间即所有工件完成加工的最大时间，对最大完工时间进行优化，可以有效缩短生产周期。问题模型为

$$z=\min T_{\max} \tag{1}$$

$$\sum_{j=1}^{n}X_{iqj}=1 \tag{2}$$

$$S_{iq}+T_{iqj}\leqslant S_{kl}+R\left(1-Y_{iqklj}\right) \tag{3}$$

$$E_{i(q-1)}\leqslant S_{iq}+R\left(1-Y_{i(q-1)iqj}\right) \tag{4}$$

$$T_j\leqslant T_n \tag{5}$$

$$E_{i(q-1)}\leqslant S_{iq} \tag{6}$$

$$S_{iq}+T_{iqj}X_{iqj}\leqslant E_{iq} \tag{7}$$

$$T_i\leqslant T_{\max} \tag{8}$$

$$E_{iq}-S_{iq}=T_{iqj} \tag{9}$$

$$S_{iq}\geqslant 0,T_{iqj}\geqslant 0,E_{iq}\geqslant 0,T_j\geqslant 0 \tag{10}$$

式中：式（1）为目标函数，即最小化最大完工时间，$T_{\max}$ 为最大完工时间；式（2）表示加工过程中同时刻同一道工序只能在一台机器上处理，其中，X_{iqj} 为决策变量，若工序 O_{iq} 在机器 N_j 上加工则为 1，否则为 0；O_{iq} 为工件 M_i 的第 q 道工序，$\{i|1,2,\cdots,m\}$，$\{j|1,2,\cdots,n\}$；式（3）与式（4）为加工过程中的机器约束，同时刻的某一台机器上只允许加工一道工序，且机器上工序存在加工顺序，其中，S_{iq} 为工序 O_{iq} 的开工时间；E_{iq} 为工序 O_{iq} 的完工时间；T_{iqj} 为工序 O_{iq} 在机器 N_j 上的加工时间；R 为无穷大的正整数；Y_{iqklj} 为决策变量，若在机器 N_j 上工序 O_{iq} 先于工序 O_{kl} 加工则为 1，否则为 0；式（5）表示每个机器的加工时间不能超过机器总加工时间，其中，T_j 为机器 N_j 的加工时长，T_n 为机器总加工时长；式（6）与式（7）表示加工过程存在先后顺序，且所有工序的开工时间不能大于其完工时间；式（8）表示每个工件的加工时间不能超过最大完工时间，其中，T_i 为工件 M_i 的完工时间；式（9）表示加工过程不能中断；式（10）表示非负约束。

2　改进细菌觅食算法

2.1　细菌觅食算法

细菌觅食算法由 Passino[11] 提出，是一种受生物启发的群体智能优化算法。细菌觅食算法基于大肠杆菌的觅食行为，通过模拟细菌觅食过程中的趋向性操作、复制操作及迁徙操作来实现寻优[12]。其中，趋向性操作是对由个体的翻转与游动构成的移动行为的模拟；复制操作是对个体的优胜劣汰与繁殖行为的模拟；迁徙操作是对由于环境突变等情况导致的个体迁徙的模拟[13]。细菌觅食算法具有简单易实现、并行搜索、鲁棒性强等优点，但仍存在无法兼顾效率与精度、易早熟收敛等缺点。故针对细菌觅食算法存在的问题进行改进，针对初始解质量低问题，设计并引入 Logistic-Circle 混合映射；针对无法同时兼顾效率与精度问题，设计并引入自适应动态步长；针对易早熟收敛问题，结合政治优化算法的“子群转换”思想改进复制操作；针对精英个体易“逃逸”问题设计并引入自适应迁徙概率。

2.2　编码机制及转换机制

为建立连续个体位置与离散调度解之间的映射，应用细菌觅食算法求解柔性作业车间调度问题时需进行离散编码。FJSP 中包含两个问题，即工序排列与机器选择，因此每个个

体采用前后等长的两段式编码。为实现算法解空间与问题空间之间的转换，在工序排列与机器选择部分分别采用不同的转换机制。编码机制与转化机制参考文献[14]。

机器选择即为各工序选用加工机器。在机器选择中，个体位置向量与机器选择相互转换时分别采用式(11)与式(12)。

$$u(h)=\mathrm{round}\{\frac{1}{2\varepsilon}[X(h)+\varepsilon][Z(h)-1]+1\},1\leqslant h\leqslant l \tag{11}$$

式中：$u(h)$为所选机器在工序可用机器集中的编号；$Z(h)$为对应工序可选机器的数量；$X(h)$为个体位置向量；l为总工序数；ε为工件个数。

$$X(h)=\frac{2\varepsilon}{Z(h)-1}[u(h)-1]-\varepsilon \tag{12}$$

当$Z(h)=1$时，$X(h)$在$(-\varepsilon,\varepsilon)$中取任意值。

工序排列即确定各机器所加工的顺序，首先采用升序排列(ranked order value，ROV)规则为每个位置向量赋予唯一的ROV值，由个体位置向量转换为工序排列时，根据ROV值即可构造工序排列方案；由工序排列转换为个体位置向量时，则根据工序编号重新排列ROV值并根据ROV值确定个体位置向量。

2.3 基于混合映射的初始化

细菌觅食算法的初始种群是随机生成的，种群质量无法保证，影响算法寻优精度，故将Circle映射引入种群初始化，以提高初始种群质量。Circle映射是一种基于圆周运动的混沌映射，具有较好的遍历性及稳定性，其表达式为

$$x_{i+1}=\mathrm{mod}[x_i+b-\frac{a}{2\pi}\sin(2\pi x_i),1] \tag{13}$$

式中：mod为求余函数；a与b为控制参数，常取值0.5与0.2。

Circle映射输出的初始种群较为规律，种群多样性较低，故应用随机性较强的Logistic映射对Circle映射参数b进行扰动，设计并引入Logistic-Circle混合映射，以提升初始种群多样性，避免算法早熟收敛。Logistic映射是常用混沌映射之一，其表达式为

$$x_{i+1}=rx_i(1-x_i) \tag{14}$$

式中：r为控制参数，取值为4时系统处于完全混沌。

引入Logistic映射后，Circle映射的参数b采用式(15)计算。

$$b=b_0+0.1\times(x_i^L+0.5) \tag{15}$$

式中：b_0为参数b初始值，取值为0.2；x_i^L为Logistic映射的输出。

2.4 趋向性操作改进

在趋向性操作中，个体通过翻转确定游动方向，若向此方向游动会有更好的适应度值，则个体会沿此方向游动。当达到最大游动步数或适应度值开始下降时，个体会再次翻转，重新确定游动方向[15]。在细菌觅食算法中，游动步长采用固定值，但固定步长会导致算法收敛速度降低或者陷入局部极值[16]，故在趋向性操作中设计自适应动态步长，达到在算法前期采用较大步长加快收敛速度、在算法后期采用较小步长提高求解精度的目的，同时保证算法的寻优效率及精度。余弦函数在区间$(0,\pi)$内递减，且斜率变化规律契合趋向性操作步长需求，故将余弦函数引入自适应动态步长的计算，其公式为

$$C(i)=C_0(i)\times\{\frac{1}{2}\times[1+\cos(\pi\frac{\sqrt{j-1}}{\sqrt{Nc-1}})]\} \tag{16}$$

式中：$C(i)$为自适应动态步长；Nc为趋向性次数；j为第j代趋向性操作；$C_0(i)$为固定游动步长。

引入自适应动态步长后，个体位置更新如式(17)所示。

$$\begin{cases}\theta^i(j+1,k,i)=\theta^i(j,k,i)+C(i)\varphi(j)\\ \varphi(j)=\dfrac{\Delta(i)}{\sqrt{\Delta^{\mathrm{T}}(i)\cdot\Delta(i)}}\end{cases} \tag{17}$$

式中：$\theta^i(j,k,l)$为个体i进行第j代趋向性操作、第k代复制操作和第l代迁徙操作之后在寻优空间中的位置信息；$\varphi(j)$为个体旋转后随机选择的游动方向；$\Delta(i)$为随机向量，$\Delta(i)\in[-1,1]$。

2.5　复制操作改进

在复制操作中，健康度较低的个体会被淘汰，保留个体则在同一位置分裂为两个相同的个体[11]。健康度即个体在趋向性操作中所得适应度值之和，计算如式(18)所示。

$$J_{\text{health}}^{i}=\sum_{j=1}^{Nc}J(i,j,k,l) \tag{18}$$

式中：J_{health}^{i}为个体i的健康度；Nc为趋向性次数。

复制操作以较优的一半个体取代较差的一半个体，以加快收敛速度。然而，此操作会降低种群多样性，导致算法早熟收敛。为保证种群多样性，引入政治优化(political optimizer, PO)算法的“子群转换”思想对复制操作进行如下改进：首先，统计个体当前健康度，并降序排列；其次，采用精英保留策略，将健康度排序前20%的精英个体保留，其余80%的个体随机分配，平均组建为两个“子群”，计算各“子群”总健康度；然后，两个“子群”成员随机交换，组建为两个新“子群”并计算各“子群”总健康度，采用贪婪选择策略，保留总健康度最高“子群”；最后，进行“子群”间淘汰，将保留“子群”中健康度排名后25%的个体淘汰，并对经此操作保留的50%个体进行复制。

2.6　迁徙操作改进

在迁徙操作中，迁徙事件以固定概率Ped发生，发生迁徙时部分个体会突然消失，而后寻优空间中会随机产生相同数目的新个体。迁徙操作有利于算法跳出局部最优，但固定迁徙概率易导致优质个体消失，发生“逃逸”现象，降低算法精度[17]。因此，设计自适应迁徙概率，使较优个体迁徙概率降低，以此保留寻优过程中的次优解。改进的迁徙操作步骤如下：首先，统计每个个体当前健康度，并根据健康度大小降序排列；其次，根据自适应迁徙概率进行迁徙。自适应迁徙概率的计算为

$$\text{Ped}(i)=\frac{J_{\text{health}}^{\max}-J_{\text{health}}^{i}}{J_{\text{health}}^{\max}-J_{\text{health}}^{\min}}\cdot\text{Ped} \tag{19}$$

式中：Ped为固定迁徙概率；Ped(i)为自适应迁徙概率；$J_{\text{health}}^{\max}$为健康度排序中最高值；$J_{\text{health}}^{\min}$为健康度排序中最低值；J_{health}^{i}为个体健康度；J_{health}^{i}越大则迁徙概率越小。

3　算例分析

3.1　初始化改进测试

针对随机初始化、Circle映射初始化及Logistic-Circle混合映射初始化3种初始化方案进行测试，测试采用的硬件为Win11系统下RAM为16GB、处理器为11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz 3.19GHz的计算机；测试采用的软件为MATLAB R2021a。算法参数设置如下：种群规模S为100、最大迭代次数为400、趋向性操作次数Nc为25、每次趋向性操作时细菌个体沿同一方向游动的最大次数Ns为4、游动一次的单位步长$C_0(i)$为0.1、复制操作次数Nre为4、迁徙操作次数Ned为4、迁徙概率Ped为0.6。测试算例采用文献[18]中的算例MK01、MK07及MK10，针对每个算例独立运行10次后获得的初始化方法对比结果如表1所示。表1中，$m\times n$为问题规模，BEST为运行10次后得到的最优目标值，AVG为运行10次后得到的目标值的平均值。

表1　初始化方法对比

算例	$m\times n$	随机		Circle		Logistic-Circle	
		BEST	AVG	BEST	AVG	BEST	AVG
MK01	10×6	65	70.1	68	74.9	58	64.1
MK07	20×5	236	244.5	233	242.1	230	236.9
MK10	20×15	475	488.4	439	464.3	432	449.1

由表1可知，整体而言，采用混沌映射初始化解得的3个算例的BEST值及AVG值均小于采用随机初始化解得的BEST值及AVG值，说明采用混沌映射初始化优于采用随机初始化；而采用Circle映射初始化解得的较小规模算例

MK01的BEST值及AVG值分别为68及74.9，均高于采用随机初始化解得的算例MK01的BEST值65及AVG值70.1，说明Circle映射初始化求解小规模问题的性能较差；采用Logistic-Circle混合映射初始化解得了3个算例的BEST值及AVG值的最优值，说明该方法具有较好的全局搜索能力且适用于各规模问题，故所提Logistic-Circle混合映射初始化方法更有效。

3.2 自适应步长测试

为测试所提自适应动态步长是否有效，应用引入了不同自适应动态步长的细菌觅食算法，针对文献[18]中的算例MK05、MK06及MK09进行求解，并与不应用自适应动态步长的细菌觅食算法进行比较，测试采用的软硬件条件及算法参数设置与3.1节相同。应用自适应动态步长的细菌觅食算法分别为ABFOA-A和ABFOA-B，ABFOA-B引入了2.4节所提出的自适应动态步长，ABFOA-A引入了文献[17]提出的自适应动态步长，其公式为

$$C(i)=C_0(i)\cdot\exp[-(\lambda\cdot\frac{j}{Nc})] \tag{20}$$

式中：$C(i)$为自适应动态步长；Nc为趋向性次数；j为第j代趋向性操作；$C_0(i)$为固定游动步长；λ为控制系数，经实验测试，λ取值为13时精度最优，故λ取值13。

应用3种细菌觅食算法，针对每个算例独立运行10次后解得的自适应步长对比结果如表2所示。表2中，$m\times n$为问题规模，TIME为运行该算例10次后得到的运行时间的平均值，ZAVG为运行该算例10次后得到的目标值的平均值。

由表2可知，3个算例下ABFOA-A的TIME值均为最小而ZAVG值均为最高，说明文献[17]提出的自适应动态步长可以提高寻优效率但无法保证求解精度。3个算例下ABFOA-B的TIME值高于ABFOA-A的TIME值，但仍低于BFOA的TIME值，而ABFOA-B的ZAVG值小于ABFOA-A与BFOA的ZAVG值，说明所提自适应动态步长可以兼顾寻优效率及精度。

表2 自适应步长对比

算例	$m\times n$	ABFOA-A		ABFOA-B		BFOA	
		TIME	ZAVG	TIME	ZAVG	TIME	ZAVG
MK05	15×4	26.472	226	27.993	213.5	29.956	218.4
MK06	10×15	36.617	168	37.800	163.9	38.019	166.3
MK09	20×15	61.431	622.5	63.638	558.5	67.305	560.7

3.3 改进算法性能测试

为进一步验证所提改进细菌觅食算法的有效性，针对文献[18]中的算例MK01—MK10进行求解，并与粒子群算法、果蝇算法、细菌觅食算法及文献[17]所提自适应细菌觅食算法进行对比，测试采用的软硬件条件与3.1节相同。

测试中，果蝇算法主要参数设置如下：种群规模为100、最大迭代次数为400；粒子群算法主要参数设置如下：种群规模为100、最大迭代次数为400、$c1$与$c2$为1.5、w为0.7、V_{max}为1.5；3种细菌觅食算法的参数设置与3.1节相同。

针对算例MK01—MK10，分别独立运行10次后所得运行结果如表3所示。表3中，PSO为粒子群算法；FOA为果蝇算法；BFOA为细菌觅食优化算法；ABFOA为文献[17]所提自适应细菌觅食算法；IBFOA为所提改进细菌觅食算法；$m\times n$为问题规模；BEST为运行10次后得到的最优目标值，即生产周期的最优值；AVG为运行10次后得到的目标值的平均值，即生产周期的平均值；MEAN为10个算例下该算法RPD的平均值，MEAN值越小算法的求解性能越优；RPD为相对百分比偏差计算，如式(21)所示。

$$\mathrm{RPD}=100\times\frac{\mathrm{BEST}-\mathrm{MIN}}{\mathrm{MIN}} \tag{21}$$

式中：MIN为该算例下BEST值的最小值。

表3　运行结果对比

算例	$m\times n$	PSO			FOA			BFOA			ABFOA			IBFOA		
		BEST	AVG	RPD	BEST	AVG	RPD	BEST	AVG	RPD	BEST	AVG	RPD	BEST	AVG	RPD
MK01	10×6	52	54.6	0.000	57	57.0	9.615	65	70.1	25.000	74	77.3	42.308	55	57.9	5.769
MK02	10×6	52	54.1	13.043	67	70.2	45.652	51	56.3	10.870	62	64.7	34.783	46	51.2	0.000
MK03	15×8	281	315.5	2.182	513	518.3	86.545	301	312.5	9.455	325	341.1	18.181	275	298.5	0.000
MK04	15×8	108	108.4	17.391	107	107.1	16.304	100	103.7	8.696	106	109.8	15.217	92	96.0	0.000
MK05	15×4	233	233.9	15.347	233	233.4	15.347	213	218.4	5.446	225	230.6	11.386	202	209.4	0.000
MK06	10×15	164	170.1	2.500	204	207.2	27.500	163	166.3	1.875	169	171.4	5.625	160	162.5	0.000
MK07	20×5	240	252.0	8.108	299	304.0	34.685	236	244.5	6.306	242	249.4	9.009	222	232.1	0.000
MK08	20×10	683	683.0	16.156	683	683.0	16.156	609	618.0	3.571	629	644.4	6.973	588	608.5	0.000
MK09	20×10	494	509.9	5.556	504	518.2	7.692	527	560.7	12.607	600	616.6	28.205	468	498.3	0.000
MK10	20×15	508	516.0	21.531	555	560.4	32.775	475	488.4	13.636	509	523.4	21.77	418	448.1	0.000
MEAN			10.181			29.227			9.746			19.347			0.577	

由表3可以看出，与粒子群算法及果蝇算法相比，细菌觅食算法在除MK01、MK03、MK09外的7个算例中均获得了BEST与AVG的较优值，且细菌觅食算法的MEAN值为9.746，小于粒子群算法的10.181及果蝇算法的29.227，故细菌觅食算法性能较优。所提改进细菌觅食算法在10个算例的求解上各值均优于细菌觅食算法和文献[17]所提的自适应细菌觅食算法，同时，在除MK01外的9个算例中均获得了BEST与AVG的最优值，且所提改进细菌觅食算法的MEAN值为0.577，显著小于其他算法，故所提改进细菌觅食算法具有一定优越性。改进细菌觅食算法可以优化柔性作业车间调度，缩短生产周期，如与细菌觅食算法相比，改进细菌觅食算法解得的算例MK01的AVG值为57.9，较细菌觅食算法解得的算例MK01的AVG值70.1减少了约17.404%。与细菌觅食算法相比，改进细菌觅食算法解得的10个算例的AVG值平均降低了约7.763%，即平均生产周期平均减少了约7.763%，故所提改进细菌觅食算法在优化柔性作业车间调度方面具有一定的有效性。

4　结论

1）柔性作业车间调度问题是生产流程的核心问题，优化柔性作业车间调度可以缩短制造企业的生产周期。为优化柔性作业车间调度，优选细菌觅食算法，并针对该问题及细菌觅食算法无法兼顾效率与精度、易早熟收敛、精英个体“逃逸”等问题对细菌觅食算法进行了一系列改进，提出了改进细菌觅食算法。

2）改进细菌觅食算法通过两段式个体编码，实现了连续个体位置与离散调度解之间的转换；设计并引入了Logistic-Circle混合映射，提高了初始种群的质量；在趋向性操作中设计自适应动态步长替代固定步长，同时保证了算法的寻优效率及精度；引入政治优化算法的“子群转换”思想改进复制操作，保证了种群多样性，避免算法早熟收敛；在迁徙操作中设计自适应迁徙概率，避免了固定概率引起的精英个体“逃逸”。

3）针对所提Logistic-Circle混合映射初始化及自适应动态步长进行测试，结果证明了所提改进机制的有效性；进行算例仿真测试，仿真结果中所提改进细菌觅食算法的MEAN值

为0.577，显著小于其他算法，证明改进算法性能优越；改进细菌觅食算法解得的10个算例的生产周期较细菌觅食算法平均缩短了约7.763%，证明改进细菌觅食算法在优化柔性作业车间调度、缩短生产周期方面具有一定的有效性。

在未来的研究中，将把改进细菌觅食算法拓展至更复杂的柔性作业车间调度问题，并结合问题设计改善机制，进一步提升该算法的性能。

参考文献（References）：

[1] 周鹏鹏，翟志波，戴玉森.基于改进遗传算法的柔性作业车间调度问题研究[J].组合机床与自动化加工技术，2023(3)：183-186，192.

[2] 方恒，朱建鸿.基于离散多元宇宙算法的柔性作业车间调度[J].组合机床与自动化加工技术，2023(10)：174-178.

[3] 米恬怡，唐秋华，成丽新，等.融合强化学习与变邻域搜索的柔性作业车间调度研究[J].工业工程与管理，2023，28(5)：101-107.

[4] Tutumlu B，Saraç T.A MIP model and a hybrid genetic algorithm for flexible job-shop scheduling problem with job-splitting[J].Computers & Operations Research，2023，155：106222.

[5] Akram K，Bhutta M U，Butt S I，et al.A Pareto-optimality based black widow spider algorithm for energy efficient flexible job shop scheduling problem considering new job insertion[J].Applied Soft Computing，2024，164：111937.

[6] 牛昊一，吴维敏，章庭棋，等.自适应樽海鞘群算法求解考虑运输时间的柔性作业车间调度[J].浙江大学学报(工学版)，2023，57(7)：1267-1277.

[7] 牛奔，郭晨，唐恒.基于多目标多元学习细菌觅食优化算法的混合数据聚类[J].中国管理科学，2022，30(12)：131-140.

[8] Pang S，Chen M C.Optimize railway crew scheduling by using modified bacterial foraging algorithm[J]. Computers & Industrial Engineering，2023，180：109218.

[9] 李冬琴，陈文文.基于粒子群细菌觅食混合算法的船舶动力定位推力分配研究[J].舰船科学技术，2023，45(20)：121-126.

[10] Chen H L，Zhang Q，Luo J，et al.An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine[J]. Applied Soft Computing，2020，86：105884.

[11] Passino K M.Biomimicry of bacterial foraging for distributed optimization and control[J].IEEE Control Systems Magazine，2002，22(3)：52-67.

[12] Guo C，Tang H，Niu B，et al.A survey of bacterial foraging optimization[J]. Neurocomputing，2021，452：728-746.

[13] 彭艺，马晓霖，杨青青.基于自适应细菌觅食优化策略的CR-NOMA功率分配算法[J].山东大学学报(理学版)，2024，59(1)：62-71.

[14] 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.

[15] Vital-Soto A，Azab A，Baki M F. Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility [J]. Journal of Manufacturing Systems，2020，54：74-93.

[16] Li J，Dang J W，Bu F，et al.Analysis and improvement of the bacterial foraging optimization algorithm[J].Journal of Computing Science and Engineering，2014，8(1)：1-10.

[17] 段剑峰，李成群，陈思.基于自适应细菌觅食算法的立体仓库货位优化[J].制造业自动化，2023，45(5)：107-112.

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

（责任编辑：刘划　英文审校：郑学东）