基于改进蝙蝠算法的柔性流水车间排产优化问题研究*

韩忠华[1,2,3]，朱伯秋[1†]，史海波[2,3]，林 硕[1]

(1.沈阳建筑大学 信息与控制工程学院，沈阳 110168；2.中国科学院沈阳自动化研究所 数字工厂研究室，沈阳 110016；3.中国科学院网络化控制系统重点实验室，沈阳 110016)

摘 要： 为解决柔性流水车间调度问题(flexible flow shop scheduling problem，FFSP)，提出了一种基于精英个体集的自适应蝙蝠算法(self-adaptive elite bat algorithm，SEBA)。针对蝙蝠算法存在求解离散问题具有局限性、易陷入局部极值、优化结果精度低等问题，该算法采用 ROV(ranked order value)编码方式，使算法适用于求解离散型的 FFSP；提出基于汉明距离的精英个体集，由多个适应度高但相似度低的精英个体轮流引导种群进化，增强种群进化活力，避免寻优过程陷入局部极值；提出自适应位置更新机制，提高算法优化精度。最后采用不同规模的标准实例对改进算法进行测试，与已有算法进行对比，实验结果验证了改进蝙蝠算法求解 FFSP 问题的有效性。

关键词： 柔性流水车间问题；蝙蝠算法；精英个体集；汉明距离

中图分类号： TP301.6 **文献标志码：** A **文章编号：** 1001-3695(2017)07-1935-04

doi:10.3969/j.issn.1001-3695.2017.07.003

Study for flexible flow shop scheduling problem based on advanced bat algorithm

Han Zhonghua[1,2,3], Zhu Boqiu[1†], Shi Haibo[2,3], Lin Shuo[1]

(1.*Information & Control Engineering Faculty, Shenyang Jianzhu University, Shenyang* 110168, *China*; 2.*Digital Factory Research Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang* 110016, *China*; 3.*Key Laboratory of Networked Control System, Chinese Academy of Sciences, Shenyang* 110016, *China*)

Abstract: In order to solve the flexible flow shop scheduling problem, this paper proposed the SEBA. The existing BA could not solve the discrete problem because it was easily trapped in local extremum and had low accuracy of the optimization results. SEBA adopted the ROV coding method, which made the algorithm suitable for solving discrete FFSP problems. This paper designed the set of the elite individuals based on hamming distance, which had higher fitness and lower similarities. It could also take turns to lead the population evolution, enhance the vitality of population evolution and avoid optimization process trap in local extremum. It designed an adaptive position update method to improve the accuracy of algorithm. Finally, it measured the SEBA by the datas from different scale scheduling benchmark problems with comparison of several algorithms. Simulation results show that SEBA is efficient for solving FFSP.

Key words: flexible flow shop scheduling problem(FFSP); bat algorithm; elite individual set; Hamming distance

0 引言

柔性流水车间调度问题 FFSP 可以被描述为 n 个工件按照相同的加工顺序在 $m(m\geqslant 2)$ 道工序上进行加工，m 道工序中至少有一道工序的并行机器数大于一台。汽车装配、半导体封装等行业的生产都可被归类为柔性流水车间生产模型。文献[1,2]证明了 FFSP 是典型的 NP-hard 问题。作为 NP-hard 问题中最难解决的问题之一，对 FFSP 问题的研究具有重要的理论价值和实际意义。

2010 年，受到蝙蝠在复杂情况下准确定位、捕猎的情景和算法在高维度空间寻优情况相似的启迪，Yang[3] 提出了蝙蝠算法。自提出以来，蝙蝠算法已被证明可以被应用于多种实际问题的求解。陈媛媛等人[4] 将基本蝙蝠算法与 Lévy 飞行搜索策略结合，同时进行离散化处理，提出了一种新型的红外光谱特征选择算法；Baziar 等人[5] 将改进后的蝙蝠算法应用于优化微型电网；Gao 等人[6] 通过调整 BA 参数设置将其应用到视觉追踪领域；Goyal 等人[7] 将蝙蝠算法与菌群算法结合，应用于无线传感器网络的定位。在生产调度领域中，许多专家学者应用并改进了蝙蝠算法，如徐华等人[8] 重新定义速度与位置的加法操作来实现粒子的位移，提出一种新型蝙蝠算法来求解柔性流水车间调度问题，具有较好的性能，但其在求解精确度上仍可提高；杜田田等人[9] 针对有限缓冲区的流水线调度问题

收稿日期：2016-04-25；修回日期：2016-07-12 基金项目：国家自然科学基金资助项目(61503259)；辽宁省社会科学规划基金资助项目(L15BGL017)；校涵育项目(XKHY2-61)

作者简介：韩忠华(1977-)，男，黑龙江齐齐哈尔人，教授，博士，主要研究方向为生产与运作管理、企业自动化系统集成技术、车间排产与生产调度算法；朱伯秋(1992-)，男(通信作者)，硕士研究生，主要研究方向为车间排产优化问题、生产调度优化算法的工程应用(zbq1525260501@163.com)；史海波(1966-)，男，研究员，博导，主要研究方向为生产与运作管理理论、制造过程建模与仿真技术、制造执行系统技术、数字化装备与智能系统技术；林硕(1981-)，男，副教授，主要研究方向为生产调度优化算法的工程应用.

?1994-2017 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

提出一种新型蝙蝠算法，改进算法表现出良好的效果，但算法在解决大规模问题时求解速度较慢；Luo 等人[10]在蝙蝠算法的基础上引入 NEH 启发式算法，并将调度问题分解为若干个子问题，求解了置换流水车间调度问题。蝙蝠算法作为一种新兴的智能仿生算法，在各个领域的研究还不完善。标准的 FFSP 属于离散问题，需要对蝙蝠算法进行改进，使其能够应用于 FFSP。本文将基于精英个体集的自适应蝙蝠算法（SEBA）应用到 FFSP 上，把仿真结果与蝙蝠算法、遗传算法、紧致遗传算法进行对比，验证了本文的改进方式具备良好的优化性能。

1 FFSP 描述

1.1 主要参数

n 表示待加工的工件总数；J_i 表示序号为 i 的工件，$i\in\{1,2,\cdots,n\}$；m 表示加工的工序总数；O_j 表示第 j 道工序，$j\in\{1,2,\cdots,m\}$；$S_{i,j}$表示工件 J_i 在工序 O_j 上的开工时间；$C_{i,j}$表示工件 J_i 在工序 O_j 上的完工时间；$T_{i,j}$表示工件 J_i 在工序 O_j 上的加工时间；M_j 表示每道工序上的并行机数；$C_{\max}$表示最大完工时间。

1.2 问题描述

基于上述提出的参数，FFSP 可以描述为：n 个待加工的工件，需经过 m 道工序的加工。至少有一道工序的并行机器数 $M_j\geqslant 2$。同一工件在同一工序中的任意机器上的加工时间相同，所有工件各工序的加工顺序相同。并且所有工件在所有工序上的加工时间 $T_{i,j}$ 已知。

1.3 基本假设与约束条件

1.3.1 约束条件与评价指标

$$C_{i,j}=S_{i,j}+T_{i,j}\quad i\in\{1,2,\cdots,n\},j\in\{1,2,\cdots,m\} \tag{1}$$

$$S_{i,j}\geqslant C_{i,j-1}\quad i\in\{1,2,\cdots,n\},j\in\{1,2,\cdots,m\} \tag{2}$$

$$C_{\max}=\max\{C_{1,m},C_{2,m},\cdots,C_{n,m}\} \tag{3}$$

$$\min\ C_{\max} \tag{4}$$

式(1)表示同一道工序的完工时间与开工时间的关系；式(2)确保上道工序加工完成，下道工序才可以开工；式(3)(4)说明以最小化最大完工时间作为评价指标。FFSP 的优化性能指标为总加工时间最小，调度的目标是求解 n 个工件的最优加工调度顺序，使总完工时间 $C_{\max}$值最小。

1.3.2 基本假设

a) 每个工件可以在不同的机器上加工；

b) 每个工件一旦开始加工就不能中断；

c) 每台机器同时只能加工一个工件，每个工件同时只能被一台机器加工；

d) 工件的准备时间和移动时间包含在加工时间之内；

e) 工件可等待下一道工序开工，机器在工件就位前可闲置。

2 蝙蝠算法

2.1 蝙蝠算法的主要步骤

a) 初始化算法基本参数，目标函数$f(X_k^t)$，$X_k^t=(x_1^k,x_2^k,\cdots,x_d^k)^{\mathrm{T}}$，$V_k^t=(v_1^k,v_2^k,\cdots,v_d^k)^{\mathrm{T}}$。在本文研究问题中，问题的维度 d 即工件个数 n，设置蝙蝠数目 NP，F 表示搜索脉冲频率，其范围为 $[F_{\min},F_{\max}]$。设置脉冲发生率 r，响度 A_0 范围均为 $[0,1]$。最大进化代数 $\mathrm{Gen}_{\max}$。

b) 随机初始化蝙蝠的初始位置 X_k^0 和初始速度 V_k^0，其中 $k\in\{1,2,\cdots,NP\}$。找出适应度值高的 y 个蝙蝠形成最佳解集 $\{X\}$，并找出处于最佳位置的蝙蝠 X^*。

c) 根据式(5)～(7)初始化搜索脉冲频率 F，并产生新的位置 X_k^t 和速度 V_k^t。

d) 生成随机数 rand_1，如果 $\mathrm{rand}_1>r_k$，从最佳解集$\{X\}$中选择一个解，根据式(8)对选出的解随机扰动，用扰动产生的位置代替蝙蝠当前位置。

e) 通过随机飞行产生一个新解。

f) 生成随机数 rand_2，如果 $\mathrm{rand}_2>A_k$，且蝙蝠当前位置得到改善，则移动到更新后的位置。

g) 更新之后的蝙蝠位置若优于最佳蝙蝠位置，根据式(9)(10)更新脉冲频度 r 和脉冲音强 A。

h) 评估蝙蝠群体，找出最佳蝙蝠位置并更新最佳解集$\{X\}$。

i) 若满足最大搜索代数，输出最优个体值；否则转入 c)，进行下一代搜索。

2.2 蝙蝠的速度和位置的更新式

$$F_k=F_{\min}+(F_{\max}-F_{\min})\beta \tag{5}$$

$$V_k^t=V_k^{t-1}+(X_k^t-X^*)F_k \tag{6}$$

$$X_k^t=X_k^{t-1}+V_k^t \tag{7}$$

其中：F_k 是第 k 只蝙蝠搜索猎物时使用的脉冲频率；$[F_{\min},F_{\max}]$ 是搜索脉冲频率范围；β 是 $[0,1]$ 上服从均匀分布的随机因子；V_k^{t-1} 和 V_k^t 分别表示第 k 只蝙蝠在 $t-1$ 和 t 时刻的飞行速度；X_k^t 表示第 k 只蝙蝠在 t 时刻的空间位置；X^* 表示当前搜索过程中蝙蝠群体中的最佳位置。

2.3 局部搜索时蝙蝠的更新式

$$X_{\mathrm{new}}=X_{\mathrm{old}}+\varepsilon\times A_t \tag{8}$$

其中：ε 是一个 $[-1,1]$ 的随机数，$A_t=\langle A_k^t\rangle$ 是所有蝙蝠在此时刻响度的平均值。蝙蝠搜索猎物过程中发射脉冲的频度和脉冲音强的更新式如式(9)(10)所示。

$$A_k^{t+1}=\alpha\times A_k^t \tag{9}$$

$$r_k^{t+1}=r_k^0[1-\exp(-\gamma t)] \tag{10}$$

其中：α 和 γ 为恒定量。

3 改进蝙蝠算法

3.1 蝙蝠算法存在的问题

问题 1 蝙蝠算法的测试函数都为连续函数，而柔性流水车间调度问题等实际工程问题大多都是离散问题，导致蝙蝠算法的适用性降低。

问题 2 在局部搜索和位置更新部分，新位置是由当前最佳位置扰动产生。这使得进化过程由单一个体引导，易造成种群多样性不足，陷入局部极值。

问题 3 蝙蝠在运动过程中，飞行速度对空间位置的影响程度不随搜索代数变化。随着进化的进行，蝙蝠距离最优解位置越来越近，大范围地搜索易导致蝙蝠偏离最佳位置降低搜索效率，并且使优化精度下降。

3.2 改进蝙蝠算法

针对问题 1，为了能让蝙蝠算法应用于离散型问题，本文采用基于工件升序排列（ROV）[11]的编码方式对个体进行编码，从而实现将连续位置矢量转换成工件排序的序列，使蝙蝠算法适用于求解离散型生产调度优化问题。此方法具有很好的普适性。采用 n 维矢量 $X_k^t=[x_1^k,x_2^k,\cdots,x_n^k]$ 表示个体位置，利用 ROV 编码规则把矢量分量的值转变为大小顺序。如表 1

?1994-2017 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

所示，矢量中最小的分量被赋值为 1，第二小的分量被赋值为 2，依次类推，直到所有的分量赋值完成。

表 1　ROV 编码前后

ROV 编码前	3.235	0.235	2.152	9.325	1.236
位置分量	1	2	3	4	5
ROV 编码后	4	1	3	5	2

针对问题 2，提出基于汉明距离的最优个体集。

为解决由单一个体引导种群进化导致的种群多样性不足的问题，提出含有多个最优个体的最优个体集。但若最优个体集中存在相似个体，最优个体集的策略就不会体现出很好的效果。所以选择适应度高但相似度低的个体组成最优个体集十分必要，提出了基于汉明距离的最优个体集。具体操作如下：

在进化的过程中首先根据适应度值选择一定数量的优秀个体，然后比较个体之间的相似度与设置的相似度阈值 Rt，进一步筛选出适应度高但相似度低的两个个体构成最优个体集。其中 S 为两个个体经过 ROV 编码后相同工序占总工序的比例，计算如式(11) ~(13) 所示。

$$d_i=\begin{cases}0 & x_i^{k1}\neq x_i^{k2}\\ 1 & x_i^{k1}=x_i^{k2}\end{cases} \tag{11}$$

假设变量 $d_i(i\in\{1,2,\cdots,n\})$ 表示两个不同的个体 X^{k_1}、X^{k_2} 中，对应位置工序是否相同，如果相同，则 $d_i=1$，否则 $d_i=0$。

$$N_S=\sum_{i=1}^{n}d_i \tag{12}$$

其中：N_S 表示两个个体相同基因片段的个数。

$$S=\frac{N_S}{n} \tag{13}$$

如果 S 小于相似度阈值 Rt，则说明两个个体不相似。在个体速度更新和局部搜索的过程中，从最优个体集中随机选择一个个体引导进化的进行。最优个体集随着进化的进行不断更新，使得进化由多个精英个体引导，避免进化陷入局部极值。

针对问题 3，提出自适应位置更新机制。

自适应位置更新提出自适应系数 G，即飞行速度对位置更新的影响权重。

$$G=\left(1-\frac{\mathrm{Gen}_{\mathrm{current}}}{\mathrm{Gen}_{\max}}\right)^2 \tag{14}$$

其中：$\mathrm{Cen}_{\max}$ 为最大进化代数，$\mathrm{Cen}_{\mathrm{current}}$ 为当前运行代数。

自适应位置更新如式(15) 所示。

$$X_k^t=X_k^{t-1}+G\times V_k^t \tag{15}$$

蝙蝠的位置更新由飞行速度和当前进化代数共同决定。根据式(14) (15) 可知，随着进化代数的增加，自适应系数 G 逐渐减小，飞行速度对位置更新的影响权重变小，使个体在最优解附近进行搜索，提高搜索效率，做到精确搜索。

3.3　编码和解码

采用基于对工件排序的编码方式编码。个体基因为 $\{g_1,g_2,\cdots,g_i,\cdots,g_n\}$，$g_i$ 表示第 i 位的基因记录的工件。如果是由五个工件组成的排产任务，个体{5,3,2,4,1} 表示工件的加工顺序为{J_5,J_3,J_2,J_4,J_1}。

个体的解码过程：首先将个体解码为工件的第 1 道工序加工顺序，然后再按照指定的工件加工顺序依次安排工件的加工工位，在指派加工工位工程中选用最先空闲机器优先原则(first available machine first rule，FAMFR) 为工件安排工位加工。

在后续加工工序过程中，按工件在上道工序的完工时间由小到大排列来安排本道工序的加工顺序。当出现完工时间一致的工件时，根据最大剩余加工时间原则(most work remaining rule，MWKR) 安排工件的加工顺序。在指派加工工位的过程中，同样是采用 FAMFR 原则为工件安排工位加工。直到所有工件完成所有工序的加工，输出最大完工时间。SEBA 流程如图 1 所示。

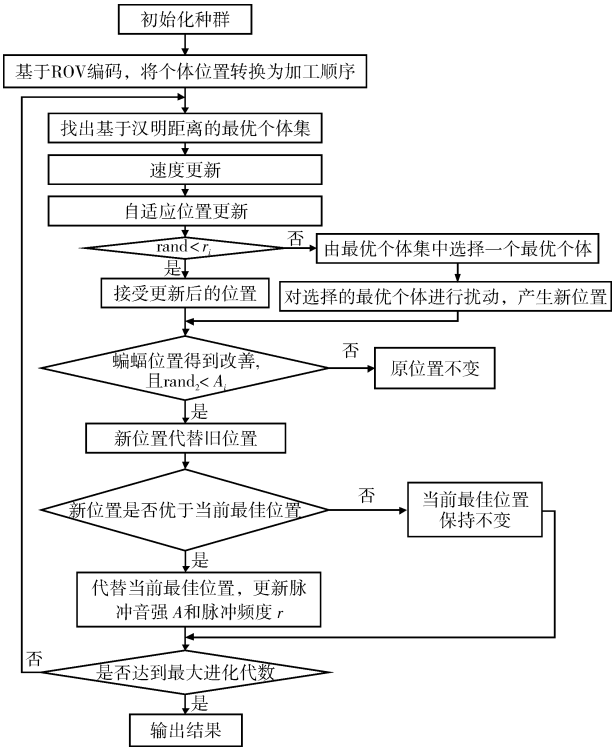

图 1　SEBA 流程

4　仿真实验

本算法基于 MATLAB 实现，运行环境为 Windows 10(64 bit)，Intel® Core i7-5500U@2.40 GHz，MATLAB 2014a，内存为 8 GB。算法参数设置参照文献[3,12]，如表 2 所示。其中，NP 为种群规模；$\mathrm{Gen}_{\max}$ 为最大迭代次数；$C_{\max}$ 为总完工时间。

表 2　算法参数设置

参数名称	NP	$\mathrm{Gen}_{\max}$	$F_{\max}$	$F_{\min}$	α	β
参数值	30	500	1	0	0.9	0.9

采用文献[13] 中所使用的标准案例测试改进蝙蝠算法的优化性能。

4.1　小规模数据测试

该组案例均为小规模问题，主要测试改进蝙蝠算法对小规模数据的优化能力。选择遗传算法(GA)、紧致遗传算法(CGA)、蝙蝠算法(BA) 作为对比算法与 SEBA 算法进行对比。每种算法对每组数据独立运行 20 次，取最优 $C_{\max}$ 值和最优相对误差(best relative error，BRE) 以及平均相对误差(average relative error，ARE) 作为系统的评价指标。其中，C^* 为标准算例理论下限。总完工时间 $C_{\max}$ 值越小，则代表算法优化效果越好。BRE 计算如式(16) 所示，代表各算法独立运行 20 次所得结果的最优值与标准算例理论下限的相对偏差。

$$\frac{[\min(C_{\max})-C^*]}{C^*}\times 100\% \tag{16}$$

根据表 3 小规模数据的对比结果可以看出，各算法对于 J15c5d 类小规模问题的优化性能存在差异性。SEBA 对于 j15c5d 类小规模问题优化所得的总完工时间 $C_{\max}$ 最优值，均小

?1994-2017 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

于其他三种算法。SEBA 所得的 C_{max} 平均值为 83.2，BA 所得的 C_{max} 平均值为 88.6，CGA 所得的 C_{max} 平均值为 91.6，GA 所得的 C_{max} 平均值为 95.8。由此可以分析得出，SEBA 在小规模数据优化性能方面优于其余三种算法，总完工时间 C_{max} 平均值比 GA 小 15.1%，比 CGA 小 10.1%，比 BA 小 6.5%。其中图 2 为各算法 BRE 对比。

表 3 小规模数据测试对比结果

标准算例	C^*	GA		CGA		BA		SEBA	
		C_{max}	BRE	C_{max}	BRE	C_{max}	BRE	C_{max}	BRE
J15c5d2	82	102	24.4	95	15.9	90	8.9	86	4.6
J15c5d3	77	94	22.1	93	20.8	86	14.2	83	7.8
J15c5d4	61	97	59	93	52.5	90	47.5	85	39.3
J15c5d5	67	96	43.3	89	32.8	84	25.3	80	19.4
J15c5d6	79	90	13.9	88	11.4	83	5.1	82	3.8

根据如图 3 所示 ARE 指标数据来看，SEBA 所求得总完工时间 C_{max} 的平均质量远高于 BA，显示出 SEBA 具有良好的进化能力和鲁棒性，证明改进算法的有效性。其中 ARE 计算如式(17)所示，代表各算法独立运行 20 次所得结果的平均值与标准算例理论下限的相对偏差。

$$ARE = [avg(C_{max}) - C^*] / C^* \times 100\% \quad (17)$$

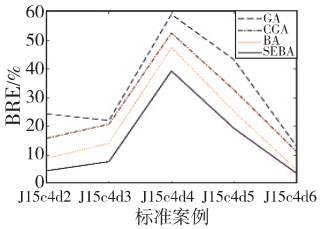

图 2 各算法 BRE 比较

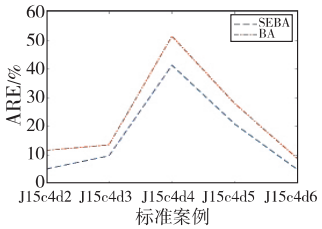

图 3 BA 与 SEBA 算法 ARE 比较

4.2 大规模数据测试

该组数例为 80 个工件 4 道工序、80 个工件 8 道工序、120 个工件 4 道工序、120 个工件 8 道工序的大规模问题，主要测试改进蝙蝠算法对大规模数据的优化能力。选择遗传算法(GA)、紧致遗传算法(CGA)、蝙蝠算法(BA)作为对比算法。每种算法对每组数据独立运行 20 次，取最优 C_{max} 值作为评价指标。最优 C_{max} 值越小，则代表算法优化效果越好。大规模数据测试对比结果如表 4 所示。

表 4 大规模数据测试对比结果

标准算例	GA	CGA	BA	SEBA	标准算例	GA	CGA	BA	SEBA
J80c4a1	1 699	1 627	1 459	1 426	J120c4a2	2 654	2 367	2 353	2 225
J80c4a2	1 667	1 504	1 374	1 320	J120c4a3	2 560	2 285	2 295	2 170
J80c8a1	2 301	2 012	1 819	1 748	J120c4a4	2 514	2 379	2 325	2 192
J80c8a2	2 033	2 016	1 868	1 783	J120c8a1	2 826	2 725	2 527	2 489
J80c8a3	2 034	1 941	1 852	1 785	J120c8a2	2 731	2 588	2 489	2 423
J80c8a4	2 039	1 976	1 865	1 795	J120c8a3	2 863	2 627	2 539	2 452
J120c4a1	2 688	2 289	2 172	2 089	J120c8a4	3 203	2 753	2 601	2 574

对于 J80c8a 类问题，SEBA 的优化结果均值为 1 777.5，比 GA 小 324，比 CGA 小 208.75，比 BA 小 73.25。对于 J120c4a 类问题，SEBA 的优化结果均值为 2 169，比 GA 小 435，比 CGA 小 161，比 BA 小 117.25；对于 J120c8a 类问题，SEBA 的优化结果均值为 2 484.5，比 GA 小 420，比 CGA 小 118.75，比 BA 小 54.5。由此可以得出，SEBA 在大规模数据寻优方面具有很好的效果。

4.3 SEBA 优化效果与迭代次数的关系

采用 120 个工件 4 道工序的 J120c4a1 标准实例作为测试数据。选取 BA 作为对比对象与 SEBA 进行对比。优化所得总完工时间 C_{max} 值与迭代次数的关系如图 4 所示。

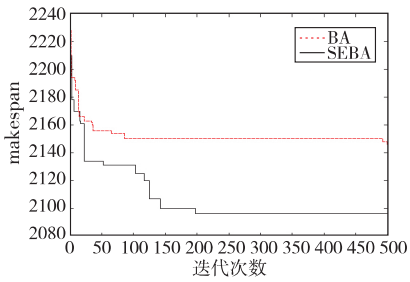

图 4 BA 与 SEBA 优化所得值与迭代次数关系

分析图 4 可知，SEBA 在 25 代之前适应度值下降较快，说明 SEBA 在优化前期具有较强的寻优能力；而 BA 在进入 100 代以后就停止进化，总完工时间 C_{max} 仅达到 2 150 的水平，说明 BA 进入到局部极值；SEBA 由于引入了基于汉明距离的最优个体集，由多个精英个体引导种群进化，使得整个进化过程保持良好的趋势。在进化后期，SEBA 收敛于最优解 2 095，远好于 BA 的 2 155。基于以上分析可得出 SEBA 的有效性。

5 结束语

针对标准蝙蝠算法对于求解 FFSP 的局限性，本文通过采用 ROV 编码，使其可以应用于解决离散问题，扩展了算法的适用范围；加入了基于汉明距离的最优个体集和自适应位置更新机制，增强进化过程中的基因多样性，避免 SEBA 陷入局部最优。通过柔性流水车标准数例测试，实验结果表明，相对于遗传算法、紧致遗传算法、蝙蝠算法，SEBA 具有更好的求解精度和求解稳定性。下一步的工作目标将是研究如何改进初始种群的生成，提高算法解决复杂问题的效率。

参考文献：

[1] Khamseh A，Jolai F，Babaei M. Integrating sequence-dependent group scheduling problem and preventive maintenance in flexible flow shops [J]. The International Journal of Advanced Manufacturing Technology，2015，77(1-4)：173-185.
[2] Gerstl E，Mosheiov G，Sarig A. Batch scheduling in a two-stage flexible flow shop problem [J]. Foundations of Computing and Decision Sciences，2014，39(1)：3-16.
[3] Yang Xinshe. A new metaheuristic bat-inspired algorithm [C] // Nature Inspired Cooperative Strategies for Optimization. Berlin：Springer，2010：65-74.
[4] 陈媛媛，王志斌，王召巴. 基于改进蝙蝠算法的红外光谱特征选择[J]. 红外与激光工程，2014，43(8)：2715-2721.
[5] Baziar A，Kavoosi-Fard A，Zare J. A novel self adaptive modification approach based on bat algorithm for optimal management of renewable MG [J]. Journal of Intelligent Learning Systems and Applications，2013，5(1)：11-18.
[6] Gao Mingliang，Shen Jin，Yin Liju，*et al*. A novel visual tracking method using bat algorithm [J]. Neurocomputing，2016，177(C)：612-619.
[7] Goyal S，Patterh M S. Modified bat algorithm for localization of wireless sensor network [J]. Wireless Personal Communications，2016，86(2)：657-670.
[8] 徐华，张庭. 新型离散蝙蝠算法求解柔性流水车间调度问题[J]. 计算机工程与应用，2016，52(2)：262-265.
[9] 杜田田，李芳，武超然. 求解有限缓冲区流水线调度问题的混合蝙蝠算法[J]. 计算机应用与软件，2015，32(6)：232-235.
[10] Luo Qifang，Zhou Yongquan，Xie Jian，*et al*. Discrete bat algorithm for optimal problem of permutation flow shop scheduling [J]. The Scientific World Journal，2014，2014(1)：1-15
[11] Bean J C. Genetic algorithm and random keys for sequencing and optimization [J]. ORSA Journal of Computing，1994，6(2)：154-160.
[12] Yilmaz S，Küçüksille E U. A new modification approach on bat algorithm for solving optimization problems [J]. Applied Soft Computing，2014，28(5)：259-275.
[13] Carlier J，Neron E. An exact method for solving the multi-processor flowshop [J]. RAIRO-Operations Research，2000，34(1)：1-25.

?1994-2017 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net