

求解有限缓冲区流水线调度问题的混合蝙蝠算法

杜田田 李芳 武超然

(上海理工大学管理学院 上海 200093)

摘要 针对最小化最大完成时间的有限缓冲区流水线调度问题(LBPFSP),提出一种混合蝙蝠算法(HBA)。在算法设计中,采用基于 SPV 的编码规则以实现连续的实数向离散的作业序列的转变,在种群初始化时引入 NEH 启发式算法,为提高算法的搜索效率,以一定的概率执行基于 Pairwise 的邻域搜索。对标准测试问题在不同缓冲区下的最优值进行仿真并与其他算法进行对比,并对基于 Pairwise 的邻域搜索的执行概率值对算法性能的影响进行讨论,其结果验证了 HBA 求解 LBPFSP 的有效性。

关键词 流水线调度 有限缓冲区 蝙蝠算法 混合优化 邻域搜索

中图分类号 TP181 文献标识码 A DOI:10.3969/j.issn.1000-386x.2015.06.057

HYBRID BAT ALGORITHM FOR SOLVING PERMUTATION FLOW-SHOP SCHEDULING PROBLEM WITH LIMITED BUFFERS

Du Tiantian Li Fang Wu Chaoran

(School of Management, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract This paper proposes an effective hybrid bat algorithm (HBA) to solve the problem of permutation flow-shop scheduling with limited buffers (LBPFSP) which minimises the makespan. In algorithm design, we use the smallest position value-based coding rule to convert consecutive real number to discrete job sequences, and introduce NEH heuristic algorithm when the population is initialising. In order to improve the search efficiency of the algorithm, the neighbourhood search based on Pairwise is performed at a certain probability. In addition, we simulate the optimal values of standard test problem in different buffer sizes and compare them with other algorithms, and discuss the effects of the performing probability value of Pairwise-based neighbourhood search on algorithm's performance. Simulation and comparisons results have verified the effectiveness of solving LBPFSP with HBA.

Keywords Flow-shop scheduling Limited buffers Bat algorithm Hybrid optimisation Neighbourhood search

0 引言

传统典型流水线调度问题是每台机器上所有工件加工顺序相同的置换流水线调度问题(PFSP),其模型假定两台机器之间的缓冲区无限大。然而在实际生产中,受缓冲区空间或者存储设备的限制,缓冲区的大小是有限的或者不存在缓冲区,比如化工、电池制造和钢铁等实际生产系统。虽然与传统的 PFSP 相比,关于有限缓冲区流水线调度(LBPFSP)的研究不多,但是越来越受到学者们的关注。早些年研究 LBPFSP 的方法主要有数学规划、启发式算法,如 Reddi^[1]提出了一种动态规划算法来解决中间存储有限的工件排序问题;Leiste^[2]通过对若干基于优先权的启发式调度算法的比较,证明了 NEH 启发式方法是解决这种问题最好的启发式方法;Nowicki^[3]提出了禁忌搜索方法控制的局部搜索方法。近年来随着智能算法的兴起,越来越多的学者将其用于求解 LBPFSP。如 Bin Qian 等^[4]提出了一种混合差分进化法来解决多目标 LBPFSP,胡蓉则用该方法解决了加工时间随机分布的 LBPFSP^[5]。王凌等提出了多搜索模式遗传算法来解决以最小化最大完成时间为目标的 LBPFSP^[6]; Hsieh 等采用了一种有效的免疫算法进行求解^[7], Ling Wang 采用了混合

遗传算法(HGA)^[8], Bo Liu 则采用了混合粒子群算法对该问题进行求解^[9]。仿生智能算法因其良好的寻优性能,越来越多的用于求解组合优化问题。蝙蝠算法(BA)作为一种新兴的仿生智能仿真算法,除了函数优化方面的应用,学术界也越来越多地将其用于求解流水线调度问题。

蝙蝠算法是剑桥大学的 Xin-She Yang 于 2010 年提出的新型启发式算法,作者用标准函数进行测试,与其他算法相比,表现出了良好的寻优性^[10]。在短短的几年时间内,也有越来越多的学者将其用于连续函数寻优,并推广到了组合优化方面,如文献[11]提出了一种差分列维飞行蝙蝠算法用来求解 PFSP 问题,文献[12]则用蝙蝠算法对多级混合流水车间的调度问题进行了研究,均表现出了良好的效果。但目前还没有人将蝙蝠算法用于求解 LBPFSP,为了使蝙蝠算法适用于解决 LBPFSP,本文设计了一种混合蝙蝠算法,引入了基于 NEH 的初始化方法和基于 Pairwise 的邻域搜索,避免了算法陷入局部最优,加强了算法的搜索性能。

收稿日期:2013-12-02。国家自然科学基金项目(71271138);上海理工大学国家级项目与文科基地培育计划项目(5812114243)。杜田田,硕士生,主研领域:工业工程,优化调度。李芳,副教授。武超然,硕士生。

1 有限缓冲区流水线调度问题的模型

LBPFSFSP 可描述如下^[13]: n 个工件 $J = \{1, 2, \dots, n\}$ 依次在 m 台机器 $M = \{1, 2, \dots, m\}$ 上进行加工, 记 $p_{i,j}$ 为工件 j 在机器 i 上的加工时间; B_i 为相邻机器 i 和 $i + 1$ 之间的缓冲区大小; $S_{i,j}$ 表示工件 j 在机器 i 上的开始加工时刻, $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ 表示所有工件的一个排序, Π 为所有排序的集合。并且假设如下:

(1) 在任意时刻, 每个工件至多在一台机器上加工, 而每台机器也同时至多加工一个工件;

(2) 每台机器上所有工件的加工次序均相同, 即所有工件在缓冲去均服从先入先出的规则。

本文以最小化最大完成时间作为优化目标, 则 LBPFSFSP 的数学模型可表示如下:

$$S_{1,\pi(1)} = 0 \tag{1}$$

$$S_{i,\pi(i)} = S_{i-1,\pi(i)} + p_{i-1,\pi(i)} \quad i = 2, \dots, m \tag{2}$$

$$S_{i,\pi(j)} = \max \{ S_{i-1,\pi(j)} + p_{i-1,\pi(j)} S_{i,\pi(j-1)} + p_{i,\pi(j-1)} \} \quad i = 2, \dots, n \quad j = 2, \dots, B_i + 1 \tag{3}$$

$$S_{i,\pi(j)} = \max \{ S_{i-1,\pi(j)} + p_{i-1,\pi(j)} S_{i,\pi(j-1)} + p_{i,\pi(j-1)} S_{i+1,\pi(j-1-B_{i+1})} \} \quad j > B_i + 1 \tag{4}$$

$$C_{\max}(\pi) = S_{m,\pi(n)} + p_{m,\pi(n)} \tag{5}$$

$$\pi^* = \arg \{ C_{\max}(\pi) \} \rightarrow \min \quad \forall \pi \in \Pi \tag{6}$$

其中式(5)即为最大完成时间, 式(6)为最优目标值的工件排序。不难看出, 当 $B_i > n - 1$ 时, 问题则退化为传统 PFSP, 即缓冲区无穷大。若 $B_i = 0$, 则认为是阻塞流水线调度问题。

以 3 个工件为例, 举例说明如何求解有限缓冲区流水线调度问题。3 个作业在 3 台机器上的作业所需的时间如表 1 所示。

表 1 时间矩阵表

机器	作业		
	J ₁	J ₂	J ₃
M ₁	3	2	2
M ₂	5	3	4
M ₃	2	4	3

当两个机器之间的缓冲区 B_i 均为 2 时, 由式(1) - 式(6) 可以计算出该问题的最小最大完成时间为 16, 对应最优加工顺序为 $\{J_3, J_2, J_1\}$, 对应的甘特图如图 1 所示。

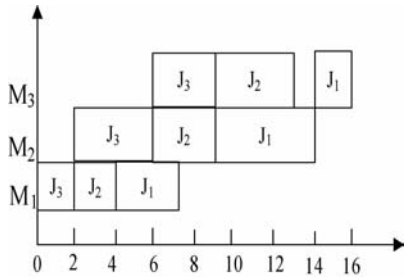


图 1 问题的甘特图表示

2 蝙蝠算法

蝙蝠是唯一会飞的哺乳动物, 与一般动物不同, 它拥有极强

的回声定位能力。在飞行过程中, 它们会发出响亮的脉冲, 然后聆听被周围物体反射回来的回声, 利用发出和探测回声的时间延迟、双耳的时间差以及回声响度的变化对这些物体进行定位, 以此来探测猎物, 避免障碍物^[14]。另外, 它们会以较大的脉冲响度和较小的发射频率接近猎物, 随着向猎物靠近, 其响度会逐渐降低而频率则逐渐升高, 在靠近目标时, 甚至会停止发射脉冲, 达到静音的状态。

蝙蝠算法正是受这种回声定位行为的启发, 在 D 维搜索空间内根据蝙蝠所发出波的频率 f 不断调整当前位置 x 和速度 v ; 在局部范围内, 根据脉冲响度 A 和脉冲频率 R 不断去搜索个体最优位置, 以下是其更新的数学表达式。

已知在 $t - 1$ 时刻蝙蝠的位置和速度, 则在 t 时刻蝙蝠的位置 x_t^i 和 v_t^i 更新可由如下公式确定:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{7}$$

$$v_t^i = v_{t-1}^i + (x_t^i - x_*)f_i \tag{8}$$

$$x_t^i = x_{t-1}^i + v_t^i \tag{9}$$

其中, β 是属于 $[-1, 1]$ 的随机数; f_{\min} f_{\max} 分别表示蝙蝠所发出波的最小和最大频率; x_* 表示蝙蝠的当前全局最优个体所在的位置。

从蝙蝠种群个体最优位置中随机选择一个个体, 新位置在其附近随机产生, 可按如下公式进行更新:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t \tag{10}$$

这种行为也可以理解为局部搜索, 其中 ε 是属于 $[-1, 1]$ 的随机数, A^t 表示所有蝙蝠在时刻 t 的平均响度。随着迭代的进行, 脉冲响度 A^t 会逐渐降低, 脉冲频率 R^t 逐渐提高。可用如下更新公式表示:

$$A_i^{t+1} = \alpha A_i^t, R_i^t = R_i^0 [1 - \exp(-\gamma t)] \tag{11}$$

在这里 α 和 γ 是恒量, 对于任何 $0 < \alpha < 1$ 和 $\gamma > 0$, 当 $t \rightarrow \infty$ 时, 都有 $A_i^t \rightarrow 0, R_i^t \rightarrow R_i^0, A_i^t = 0$ 表示蝙蝠刚刚发现猎物, 暂时停止发出声音的状态。

BA 的具体步骤如下:

步骤 1 随机初始化蝙蝠种群, 设置基本参数。

步骤 2 根据式(8)和式(9)对蝙蝠的速度和位置进行更新。

步骤 3 以一定的概率对当前任意的个体最优解根据式(10)进行局部搜索。

步骤 4 评价并更新种群个体。若 $\text{rand} < A_i^t$ 且新个体的适应度优于原来的个体, 则更新个体。

步骤 5 根据式(11)更新响度 A_i^t 和脉冲频率 R_i^t , 找出全局最优解并判断是否达到终止条件, 否则重复步骤 2。

3 求解有限缓冲区流水线调度问题

标准的蝙蝠算法在函数优化方面表现了良好的计算性能, 并具有良好的全局寻优性能。然而在解决 LBPFSFSP 时执行效率不高, 且容易在局部出现早熟。为了更好地提高搜索的效率, 在 BA 的基础上提出了 HBA。

3.1 基于 SPV 编码规则

由于蝙蝠算法中蝙蝠的位置为连续值矢量, 无法实现对工件排序的更新^[15], 本文采用基于 SPV (smallest position value) 的

续表 2

问题	m, n	C*	Buffer = 0				Buffer = 1				Buffer = 2				Buffer = 4				Buffer = ∞			
			HGA		HBA		HGA		HBA		HGA		HBA		HGA		HBA		HGA		HBA	
			BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE	BRE	ARE
Car5	10,6	7720	6.45	7.19	6.45	6.8	0	0.17	0	0.1	0	0.04	0	0.04	0	0.19	0	0.12	0	0	0	0.12
Car6	8,9	8505	4.33	4.71	4.33	4.33	0	0.57	0	0	0	0.53	0	0	0	0.5	0	0	0	0.57	0	0.04
Car7	7,7	6590	3.00	3.00	3.00	3.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Car8	8,8	8366	2.62	3.41	2.62	2.62	0	0.78	0	0	0	0.52	0	0	0	0.48	0	0	0	0.26	0	0
Rec1	20,5	1247	15.80	17.80	14.76	16.18	1.84	3.34	0.32	1.76	0	1.32	0	0.5	0	1.53	0	0.69	0	1.36	0	1.08
Rec7	20,10	1566	13.22	14.26	10.86	11.93	1.79	3.51	1.15	1.38	1.15	2.69	0.96	1.23	0	2.17	0.13	1.16	0	1.91	0.77	1.20
Rec13	20,15	1950	9.64	10.24	9.27	10.49	1.97	2.50	1.55	2.3	1.35	2.22	0.98	2.14	1.14	2.15	0.78	1.98	0.73	2.08	0.26	2.14
Rec19	30,10	2093	16.96	18.11	16.63	17.72	3.11	4.27	2.87	3.84	1.58	3.07	1.62	2.51	1.29	2.96	1.05	2.45	1.29	2.85	1.29	2.22

从表 2 可以看出:(1)除了 Buffer = 2 条件下的 Rec19、Buffer = 4 条件下的 Rec7、Buffer = ∞ 条件下的 Rec7 外,其他 57 个算例中,HBA 搜寻到的最优值均小于 HGA 搜寻到的最优值,可见 HBA 相对于 HGA 具有更好地寻优性能。(2)除了 Buffer = ∞ 条件下的 Rec13、Buffer = 1 下的 Car4 外,其他 58 个算例中,HBA 仿真所得的 ARE 均小于 HGA 所得的 ARE,由此表明 HBA 比 HGA 具有更强的鲁棒性。(3)由图 3 可以看出,同一算例在不同缓冲区大小条件下的 BRE 值随着缓冲区大小 B_i 的增大而逐渐减小,并且当 B_i 从 0 变为 1 时,减小的幅度最大。由此可见在实际生产中缓冲区的设置对减少最大完工时间发挥着重要的作用,另一方面设置缓冲区会增加成本,所以实际生产中应该权衡时间和成本,设计最合适的缓冲区大小。

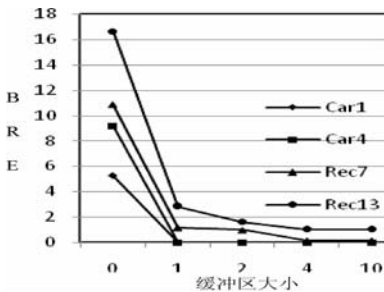


图 3 不同缓冲区大小的 BRE

4.2 基于 Pairwise 的邻域搜索的执行概率对算法性能的影响

为考查基于 Pairwise 的邻域搜索的执行概率对 HBA 算法性能的影响,现以标准问题 Rec1 为例,对缓冲区 $B = 2$ 的情况下的有限缓冲区流水线调度问题进行仿真,仿真结果如表 3 所示。

表 3 邻域搜索的执行概率对算法质量的影响

问题/m, n/C*	IBA			
	概率值	BRE	ARE	WRE
Rec01/20, 5/1247	0	3.45	6.07	6.90
	0.1	0.16	0.5	1.52
	0.2	0.16	0.42	1.92
	0.3	0.16	0.47	2.17
	0.4	0.16	0.34	1.44
	0.5	0.16	0.47	2.49
	0.6	0.16	0.39	1.36
	0.7	0.16	0.27	0.72
	0.8	0.16	0.29	0.72
	0.9	0.16	0.30	0.96
1	0.16	0.24	0.56	

由表 3 可以看出当邻域搜索的执行概率从 0 变为 0.1,其搜索质量有明显的提高,这说明了基于 Pairwise 的邻域搜索是有效的;邻域搜索对算法性能的影响如图 4 所示,从中可以看出,搜索时间与执行概率成近似线性递增关系;随着执行概率的增加,搜索质量变化不大,但搜索时间却大幅增加。综合考虑这两方面的因素,建议执行概率取值 0.1。

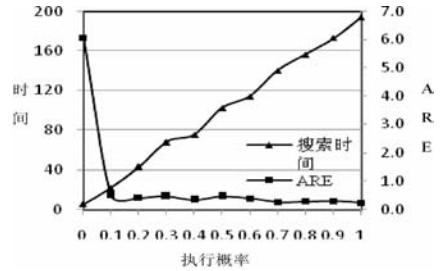


图 4 邻域搜索对算法性能的影响

5 结语

本文提出了一种混合蝙蝠算法来解决有限缓冲区的流水线调度问题,采用了基于 SPV 的编码规则,在种群初始化时引入了 NEH 启发式算法,并在基础的蝙蝠算法更新方式中引入了基于 Pairwise 的邻域搜索。虽然目前 LBFSP 的理论最优值还不知道,但仿真实验结果表明了其值已接近最优解,且与 HGA 算法相比表现出更强的寻优性和鲁棒性。基于 Pairwise 的邻域搜索虽然增强了该算法的局部搜索能力,但在对大规模流水线调度进行求解时,增加了算法的执行时间。有限缓冲区的流水线调度问题在实际生产中有很大的实用价值,有待更多的研究去弥补该算法的不足。

参 考 文 献

[1] Reddi S S. Sequencing with finite intermediate storage[J]. Management Science, 1976, 23(2): 216-217.

[2] Leisten R. Flowshop sequencing problems with limited buffer storage [J]. International Journal of Production Research, 1990, 28(11): 2085-2100.

[3] Nowicki E. The permutation flowshop with buffers: A tabu search approach [J]. European Journal of Operational Research, 1999, 116(20): 5-19.

[4] Qian B, Wang L, Huang D X, et al. An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers[J]. Computers & Operations Research, 2009, 36(1): 209-233.

变化较大、各方向车流速度均较快且均有车流量突变、车流总量整体偏大等路口常见的车流情况。当前固定红绿灯时间的控制方法并不能很好地解决这些问题。

表 3 和表 4 为各方向在相应时间段内各检测点绿灯时间分配的变化情况

表 3 各方向绿灯时间分配情况 (s)

检测点 方向	检测点									
	1	2	3	4	5	6	7	8	9	10
方向 1	35.7	28.5	24.4	21.4	17.6	11.7	8.2	5.2	3	12.4
方向 2	18.6	21.4	20.6	20.6	17.6	18.9	17.3	17.3	17.3	19.7
方向 3	2.9	2.9	5.2	5.2	6.87	7.2	6.9	6.9	6.6	11.4
方向 4	2.8	4.3	3.9	3.9	5.9	7.2	9.9	9.9	9.9	16.6
是否更新	0	1	1	0	1	1	1	0	0	0

表 4 各方向绿灯时间分配情况 (s)

检测点 方向	检测点									
	11	12	13	14	15	16	17	18	19	20
方向 1	12.4	13.9	14.1	17.2	19.7	20.3	19.2	21.2	24.2	22.6
方向 2	16.6	11.2	7.1	3	3.1	3.9	6.4	6.7	8.4	10.8
方向 3	11.4	12.1	12.3	12.7	15.5	11.6	8.2	3	7.4	8.9
方向 4	16.6	16.6	17.6	18.1	21.7	21.23	20.1	21.2	20	14.8
是否更新	1	1	1	1	0	1	1	1	0	1

而从表 3 和表 4 的仿真结果来看,方向 1 绿灯时间在第 10 个检测点处结束 $T_{1i} = 27$ s,方向 2 绿灯时间在第 15 个检测点处结束 $T_{2i} = 15$ s,方向 3 绿灯时间在第 19 个检测点处结束 $T_{3i} = 12$ s,而方向 4 的绿灯时间在这 60 s 时间段结束时仍有剩余时间 11.8 s,考虑到数据量偏大后面的仿真数据未再进行考虑,但仍可以肯定 $T_{4i} \geq 6$ s。各方向实际绿灯时间 T_{ki} 和实际绿灯周期 T_{real} 如表 5 所示。

表 5 各方向实际绿灯时间和实际绿灯周期 (s)

	方向 1	方向 2	方向 3	方向 4	T_{real}
T_{k0}	35.7	17.3	12.7	21.2	
T_{ki}	27	15	12	≥ 6	≥ 60

在 T_{k0} 时间分配上,进入方向 1 时分配的方向 1 绿灯时间 T_{10} 为 35.7 s,方向 3 T_{30} 为 12.7 s,充分体现了对各方向车流量的考虑。在第 14 个检测点时,当前绿灯方向 2 的车流量突变为 0,故马上进入黄灯时间,3 s 后进入方向 3 绿灯时间。且各方向实际绿灯时间周期 T_{real} 也比绿灯时间基准周期 60 s 大。由此可以得出本文提出的算法根据实时监测到的车流量信息自适应地改变各方向绿灯时间比例和各方向绿灯时间总和的大小,确实很好地解决了上述问题。

5 结 语

本文提出的基于车流量的交通灯智能控制算法,能够适应不同的车流情况,实现各方向绿灯时间的自适应分配。并可根据路口车流总量智能调节绿灯时间周期的大小。通过仿真证明,本文提出的算法确实很好地解决了传统交通灯控制方法不能解决的问题,对各方向车流量差异较大、车流总量变化较大的

十字路口有很好的交通疏导作用。且算法计算量小,实时性高,硬件平台要求较低,具有很好的实用价值。本算法尚未考虑各车辆等待时间和各方向车速差异等因素,研究有待进一步深化。

参 考 文 献

- [1] 张立成,蒋楠,杭亦文,等. 基于视频控制的智能交通灯[J]. 电子测量技术,2011,34(6):30-32.
- [2] 温志达,梁桂荣,陈碧铭,等. 基于车流量的智能交通灯控制系统[J]. 自动化技术与应用,2009,28(6):115-118.
- [3] 马浩轩. 基于交通流检测的智能交通系统的研究[J]. 科技视界,2012,7(9):5-7.
- [4] 宋依青,张润. 自适应交通灯控制系统的设计与实现[J]. 计算机测量与控制,2008(4):497-499.
- [5] 蔡教武,蔡延光. 一种基于自适应模糊神经网络的交通灯控制系统[J]. 仪表技术,2010(9):32-33.
- [6] 易俊华,朱红求,阳春华. 交叉路口交通灯实时模糊控制系统设计与实现[J]. 工业控制计算机,2011,24(1):47-48.
- [7] 王灿,何淳,吴亚龙,等. 智能交通灯控制系统的设计和仿真[J]. 重庆工商大学学报,2009,26(1):84-87.
- [8] Chong Y, Quek C, Loh P. A novel neuro-cognitive approach to modeling traffic control and flow based on fuzzy neural techniques[J]. Expert Systems With Applications, 2008, 36(3):4788-4803.
- [9] Stefan Lämmer, Dirk Helbing. Self-control of traffic lights and vehicle flows in urban road networks[J]. Theory and Experiment, 2008, 2008(6).

(上接第 235 页)

- [5] 胡蓉,钱斌. 一种求解随机有限缓冲区流水线调度的混合差分进化算法[J]. 自动化学报,2009,35(12):1580-1586.
- [6] 王凌,张亮. 有限缓冲区流水线调度的搜索模式遗传算法[J]. 计算机集成制造系统,2005,11(7):1041-1046.
- [7] Hsieh Y C, You P S, Liou C D. A note of using effective immune based approach for the flow shop scheduling with buffers[J]. Applied Mathematics and Computation, 2009, 215(5):1984-1989.
- [8] Wang L, Zhang L, Zheng D Z. An effective hybrid genetic algorithm for flow shop scheduling with limited buffers[J]. Computers & Operations Research, 2006, 33(10):2960-2971.
- [9] Liu B, Wang L, Jin Y H. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers[J]. Computers & Operations Research, 2008, 35(9):2791-2806.
- [10] Yang X S. Bat algorithm for multi-objective optimization[J]. International Journal of Bio-inspired Computation, 2011, 3(5):267-274.
- [11] Xie J, Zhou Y Q, Tang Z H. Differential lévy-flights bat algorithm for minimization makespan in permutation flow shops[C] // Proceedings of Ninth International Congress on Intelligent Computing Theories and Technology, Nanning, China, 2013.
- [12] Marichelvam M K, Prabakaran T. Solving hybrid flow shop scheduling problems using bat algorithm[J]. International Journal of Logistics Economics and Globalisation, 2013, 5(1):15-29.
- [13] 于艳辉,侯东亮. 具有缓冲区约束的流水车间调度问题综述[J]. 中国管理信息化, 2012, 15(5):61-63.
- [14] 刘长平,叶春明. 具有 Lévy 飞行特征的蝙蝠算法[J]. 智能系统学报, 2013, 8(1):1-8.
- [15] 许爱军. 基于免疫克隆选择算法的混合流水车间调度方法[J]. 计算机应用与软件, 2013, 30(3):75-77.
- [16] Aldowaisan T, Allahverdi A. New heuristics for no-wait flowshops to minimize makespan[J]. Computers & Operations Research, 2003, 30(8):1219-1231.