

改进布谷鸟搜索算法求解批量流水线调度问题^①

郑洪清

(广西外国语学院 信息工程学院, 南宁 530222)

摘要: 针对以最大完工时间为目标的批量流水线调度问题, 提出一种改进的布谷鸟搜索算法. 该算法采用排序规则的编码方式, 将连续个体值的布谷鸟搜索算法直接应用于离散的调度问题. 其次, 在布谷鸟搜索算法的基础上, 一个简单而有效的局部搜索用于批量流水线调度问题的探索. 仿真实验表明所提出算法的可行性和有效性.

关键词: 批量流水线调度; 布谷鸟搜索算法; 最大完工时间; 局部搜索算法

Improved Cuckoo Search Algorithm for Lot-streaming Flow Shop Schedule Problem

ZHENG Hong-Qing

(College of Information Engineering, Guangxi University Of Foreign Languages, Nanning 530222, China)

Abstract: An Improved Cuckoo Search(ICS)algorithm is presented for Lot-streaming Flow shop Scheduling Problem(LFSP)with objectives of makespan. Ordering rule is applied to enable the continuous cuckoo search algorithm to be applied to discrete scheduling problem. Then, after the CS-based exploration, a simple but efficient local search, which is designed according to the LFSP' landscape, is applied to emphasize exploitation. Simulation results show the feasibility and effectiveness of the proposed algorithms.

Key words: The cuckoo search algorithm; Lot-streaming flow-shop scheduling; Makespan; Local search

批量流水线调度问题(Lot-streaming Flow shop Scheduling Problem)是在流水线调度问题的基础上而提出的一类问题, Potts 和 Van Wassenhove 在 1992 年的时候指出批量流水线作业的优势: 首先, 分批的工件可以提高客户服务; 其次, 在多阶段生产系统中分批的工件可以使得同一工件的不同操作进行叠加. 目前, 多种方法被用于研究该问题. 如遗传算法^[1], 模拟退火算法^[2], 阈值接受算法^[3], 微粒群算法^[4], 和声算法^[5]等. 这些算法在一定程度上取得了较好的效果, 本文应用布谷鸟搜索算法对其求解, 并与文献中的结果进行比较, 效果明显.

布谷鸟搜索算法(cuckoo search algorithm, CS)是 2009 年由英国剑桥大学学者 YANG Xin-she 和 DEB Suash 基于对布谷鸟寻窝产卵的行为进行模拟, 提出了一种全新的搜索算法^[6]. 由于该算法简单、高效、容易实现, 并且成功应用于工程优化等实际问题中^[7]. 逐

渐成为群智能算法领域的一个新亮点. 研究者根据鸟巢的初始化、候选解的产生方式和解的编码等提出了一些改进的布谷鸟搜索算法, 通过实验展示了算法的有效性, 但大部分改进算法仅限于连续问题, 在离散问题上的应用还非常有限, 可供参考的文献少^[8-11]. 本文提出一种改进的布谷鸟搜索(ICS)算法来求解等规模的 LFSP, 并且融入了局部搜索算子, 大大提高了解的质量, 仿真结果表明了新算法具有良好的性能.

1 批量流水线调度问题模型

假设工件按机床 1 到 m 的顺序加工, 给定一个工件的序列 $\pi = (\pi_1, \pi_2, \dots, \pi_n)$, 令 n_j 为 j 工件的子批量; s_{ij} 为工件 j 的第 1 个子批量在机器上的开始加工时间; p_{ij} 为工件 j 的子批量在机器 i 上的加工时间; ct_{ijk} 为第 j 个工件的第 k 个子批量在机器 i 上的完成时间; c_j 为工件 j 的加工完成时间. 得到如下计算公

^① 收稿时间:2014-02-19;收到修改稿时间:2014-03-17

式:

$$ct_{1,\pi_1,1} = s_{1,\pi_1} + p_{1,\pi_1} \quad (1)$$

$$ct_{i,\pi_1,1} = ct_{i-1,\pi_1,1} + p_{i,\pi_1} + s_{i,\pi_1} \quad i = 2, \dots, m \quad (2)$$

$$ct_{1,\pi_1,k} = ct_{1,\pi_1,k-1} + p_{1,\pi_1} \quad k = 2, \dots, n_{\pi_1} \quad (3)$$

$$ct_{1,\pi_j,1} = ct_{1,\pi_{j-1},n_{\pi_{j-1}}} + p_{1,\pi_j} + s_{1,j} \quad j = 2, \dots, n \quad (4)$$

$$ct_{1,\pi_j,k} = ct_{1,\pi_j,k-1} + p_{1,\pi_j} \quad j = 2, \dots, n, \quad k = 2, \dots, n_{\pi_j} \quad (5)$$

$$ct_{i,\pi_j,1} = \max(ct_{i-1,\pi_j,1}, ct_{i,\pi_{j-1},n_{\pi_{j-1}}}) + p_{i,\pi_j} + s_{i,j} \quad j = 2, \dots, n \quad i = 2, \dots, m \quad (6)$$

$$ct_{i,\pi_j,k} = \max(ct_{i-1,\pi_j,k}, ct_{i,\pi_j,k-1}) + p_{i,\pi_j} \quad k = 2, \dots, n_{\pi_j} \quad (7)$$

$$c_j = \max(ct_{m-1,\pi_j,n_{\pi_j}}, ct_{m,\pi_j,n_{\pi_j}-1}) + p_{m,\pi_j} \quad j = 1, \dots, n \quad (8)$$

该序列的最大的完工时间为:

$c_n = \max(ct_{m-1,\pi_n,n_{\pi_n}}, ct_{m,\pi_n,n_{\pi_n}-1}) + p_{m,\pi_n}$ ，以最大完工时间为调度目标的批量流水线调度问题就是从所有的工件序列中找到一个工件序列 π^* ，使得 $c_{\max}(\pi^*) \leq c_{\max}(\pi)$ ， $\forall \pi \in \Pi$ 。

2 基本的布谷鸟搜索算法

在自然界中，布谷鸟寻找适合自己产卵的鸟巢位置是随机的或类似随机的方式，为了模拟布谷鸟寻窝的方式，首先，需要设定以下 3 个理想的状态^[6]：

- 1) 布谷鸟每一次只能产一个卵，并随机选择鸟巢位置来孵化它；
- 2) 在随机选择的一组鸟巢中，最好的鸟巢将会被保留到下一代；
- 3) 可利用的鸟巢数量 n 是固定的，一个鸟巢的主人能发现一个外来鸟蛋的概率 $p_a \in [0,1]$ ，在这 3 个理想状态的基础上，布谷鸟寻窝的路径和位置更新公式如下：

$$x_i^{(t+1)} = x_i^{(t)} + \partial \oplus L(\lambda), i = 1, 2, 3, \dots, n \quad (9)$$

式中 $x_i^{(t)}$ 表示第 i 个鸟巢在第 t 代的鸟巢位置； \oplus 为点对点乘法； ∂ 表示步长控制量； $L(\lambda)$ 为 Levy 随机搜索路径，并且 $L \sim u = t^{-\lambda}$ ， $(1 < \lambda \leq 3)$ 。通过位置更新后，用随机数 $r \in [0,1]$ 与 p_a 对比，若 $r > p_a$ ，则对 $x_i^{(t+1)}$ 进行随机改变，反之不变。最后保留测试值较好的一组鸟巢位置 $y_i^{(t+1)}$ ，此时仍把 $y_i^{(t+1)}$ 记为 $x_i^{(t+1)}$ 。可

见基本的布谷鸟搜索算法是一种连续型的智能算法，并不能直接用于离散问题的求解，如表 1 鸟巢的向量表示。

3 改进的布谷鸟搜索算法(ICS)

3.1 解的表示

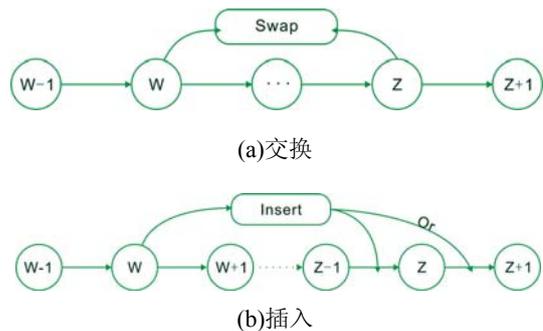
基于第 2 节的分析，标准的布谷鸟搜索算法所具有连续编码不能直接地被用于求解批量流水线调度问题，因此，构造从鸟巢位置矢量到工件排序的合适映射是应用布谷鸟搜索算法求解批量流水线调度问题首当其冲的工作。 n 个工件表示 n 维，每一维代表一个工件，而鸟巢 $X_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{in}^t]$ 表示 n 维的连续值，它本身并不能直接表示工件排序，因此，对鸟巢 X_i^t 使用降序排列规则，可以将连续值所在的位置作为离散的调度方案。表 1 阐述了这个规则，从表 1 中可以看出 3.89 是最大值，所以维数为 6 对应的调度排序为 1，稍小的是 3.14，则维数为 6 对应的调度排序为 2，以此类推，可以得到它的调度方案为 $\pi = (1, 4, 3, 6, 5, 2)$ 。通过一个这样的简单转换，则弥补了基本的布谷鸟搜索算法不能直接应用于离散问题的缺陷。

表 1 鸟巢解的表示

	维数, j					
	1	2	3	4	5	6
鸟巢	3.89	2.93	3.07	-0.87	-0.21	3.14
调度序列	1	4	3	6	5	2

3.2 局部搜索

为了提高算法的性能，本文采用交换、插入和逆序 3 种操作算子，交换即在 1 到 n 之间随机产生两个位置 w 和 z ，然后交换它们的位置值；插入即在 1 到 n 之间随机产生两个位置 w 和 z ，然后将 w 所在位置的删除，将其随机插入到 z 的左边或右边；逆序即在 1 到 n 之间随机产生两个位置 w 和 z ，将 w 和 z 之间的序列倒置后替换原来的序列。它们的搜索结构如图 1 所示。同时对最好的解使用交换和插入操作。



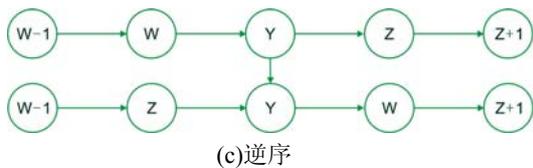


图 1 局部搜索的领域结构

3.3 算法流程

步骤 1: 设置初始的参数: 种群规模为 30, $p_a=0.25$ (p_a 为布谷鸟发现外来蛋的概率), 随机初始化解群.

步骤 2: 使用降序排序规则找出每一个鸟巢所对应的工件序列.

步骤 3: 利用目标函数评估每一个工件排序, 并找出最优解和最佳工件序列.

步骤 4: 通过 Lévy flight 获得新的布谷鸟.

步骤 5: 应用降序排序规则找出每一个鸟巢所对应的工件序列. 并评估它的适应度函数 F_i , 选择一个鸟巢(如 j), 如果 $F_i > F_j$, 则用 F_j 代替 F_i , 以 p_a 的概率抛弃差的鸟巢并建立新的鸟巢, 最后保留最好的解.

步骤 6: 对鸟巢应用 3 种操作算子(交换、插入和逆序, 如图 1 所示).

步骤 7: 对最好的解应用交换和插入算子.

步骤 8: 停止标准, 如果迭代次数超过最大迭代次数, 则停止, 否则转至步骤 4.

对基本的布谷鸟搜索算法做了 3.1 和 3.2 节的处理后, 即可将改进的算法直接应用于离散的优化问题, 而且寻优能力也大大增强.

4 数值仿真

为了检测 ICS 算法的性能, 所有的实例均运行在处理器为 Celeron(R)双核 CPU T3100, 1.90GHZ、内存为 2G 的 PC 上, 以 Matlab R2010a 编写代码, 采用与文献[3]相同的实验数据分别进行 30 次仿真. 参数设置为: 种群规模 30, $p_a=0.25$, 对最好的解搜索次数为 50, 总迭代次数为 200. 机器数分别为 3、5、7, 工件数 n 为 30, 以最大完工时间(Makespan)作为评价各个算法的标准.

表 2 给出了 6 种调度算法最大完工时间的比较数据, 图 2 给出 ICS 算法在 7×30 情况下 5 个问题以最大完工时间为调度目标的收敛进化曲线图.

表 2 6 种算法的最大完工时间的比较

m	问题	最大完工时间					ICS	
		TA		NIHS	HSTA			
		TAP	TAI		HST	HS		
1	1	815	834	801	796	796	796	
	2	691	692	691	691	691	691	
	3	3	632	632	625	625	625	625
		4	626	627	623	623	623	623
		5	775	785	770	770	770	770
2	1	872	867	853	834	838	827	
	2	778	764	747	747	747	747	
	5	3	655	648	648	627	627	627
		4	686	688	677	677	677	677
		5	797	806	792	779	776	774
3	1	894	915	886	865	864	855	
	2	792	781	764	758	757	751	
	7	3	732	717	696	688	691	686
		4	706	692	690	685	685	685
		5	849	831	806	794	793	786

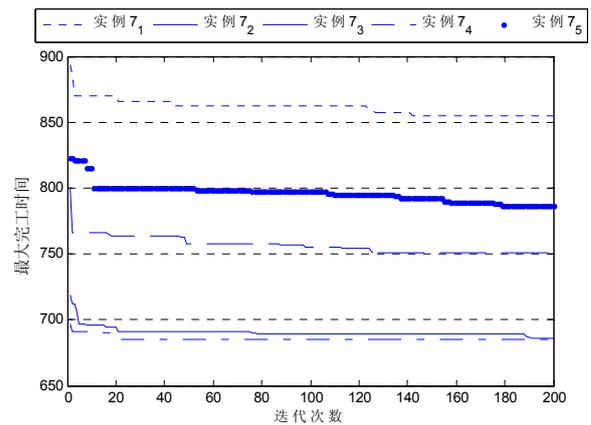


图 2 以最大完工时间为调度目标的收敛进化曲线

由表 2 可以看出, ICS 与 TAP、TAI、NIHS、HSTAP 和 HSTAI 调度算法相比, ICS 算法对不同问题、不同规模得到的最大完工时间都小于或等于其他算法得到的结果, 说明 ICS 具有更好地求解质量. 对小规模的问题, 效果不是很明显; 随着问题规模的增加, 效果越为越明显; 尤其对 7 台机器时, 几乎每次结果都比其他算法要好, 改进算法所求的结果达到了目前已知的最好解. 图 2 是 7 台机器数的最大完工时间为调度目标的收敛进化曲线, 与表 2 的 ICS 求得的数据相吻合. 但迭代次数还有点多. 图 2 中的实例 7_1 表示 7 台机器的第 1 个问题, 其他的以此类推.

5 结论

针对批量流水线调度问题的特点,应用排序规则,将改进的离散布谷鸟搜索算法成功应用于调度问题.为了提高算法的性能,融合交换、逆序和插入算子.仿真实验表明所提出的算法具有较高的优化性能.将该算法应该到其他组合优化问题中是下一步研究的方向.

参考文献

- 1 Yoon SH, Ventura JA. An application of genetic algorithm to lot-streaming flow shop scheduling. *IIE Transaction*, 2002, 34(9): 779–787.
- 2 Marimuthu S, Ponnambalam SG, Jawahar N. Evolutionary algorithms for scheduling m-machine flow shops with lot-streaming. *Robotics Computer-Integrated Manufacturing*, 2008, 24(1): 125–139.
- 3 Marimuthu S, Ponnambalam SG, Jawahar N. Threshold accepting and ant-colony optimization algorithms for scheduling m-machine flow shops with lot streaming. *Journal of Materials Processing Technology*, 2009, 209(2): 1026–1041.
- 4 Tseng CT, Liao CJ. A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. *European Journal of Operation Research*, 2008, 191(2): 360–373.
- 5 韩红燕,潘全科.求解批量流水线调度问题的改进和声搜索算法. *计算机工程*, 2011, 37(6): 178–181.
- 6 王凡,贺兴时,王燕.基于高斯扰动的布谷鸟搜索算法. *西安工程大学学报*, 2011, 25(4): 566–569.
- 7 Yang XS, Deb S. Engineering optimization by cuckoo search. *Int J Math Modeling & Num Optimization*, 2010(4): 330–343.
- 8 Zheng HQ, Zhou YQ. A discrete binary version of cuckoo search for knapsack problems. *Advances in Information Sciences and Service Sciences*, Oct 2012, 4(18): 331–339.
- 9 Zhou YQ, Zheng HQ. An improved cuckoo search algorithm for solving planar graph coloring problem. *Applied Mathematics & Information Sciences*, *Appl. Math. Inf. Sci.* 2013, 7(2): 785–792.
- 10 Zheng HQ, Zhou YQ. A hybrid cuckoo search algorithm-GRASP for vehicle routing problem. *Journal of Convergence Information Technology*, 2013, 8(3).
- 11 Zheng HQ, Zhou YQ. A hybrid cuckoo search algorithm-GA for the runway dependent aircraft landing problem. *Research J. of Applied Sciences, Engineering and Technology*, 2013, 6(12): 2136–2140.