

# 批处理机调度问题的蚁群算法求解

夏欣

(南京农业大学信息科技学院 南京 210095)

**摘要** 为了求解工件具有不同尺寸的批处理机调度问题,将蚁群算法调整为工件直接成批的调度算法,并提出了一个新的局部优化算法对蚁群算法进行改进。最后通过仿真实验将本算法与其它算法对本问题的求解进行了比较,表明该算法在求解批调度问题上有较好的性能。

**关键词** 调度,蚁群算法,制造跨度,组合优化

**中图分类号** TP301 **文献标识码** A

## Scheduling Single Batch Processing Machine Using Ant Colony Algorithm

XIA Xin

(Information and Technology Department, Nanjing Agriculture University, Nanjing 210095, China)

**Abstract** An ant colony algorithm(ACO) was proposed to study the problem of minimizing makespan on a single batch processing machine with non-identical jobs sizes. ACO was used by grouping several jobs into a batch directly and the parameter settings were also discussed. A new local search algorithm was provided to improve the performance of the ACO. The computational experiment was conducted to show the improvement of ACO compared to some other algorithms.

**Keywords** Scheduling, Ant colony algorithm, Makespan, Combinatorial optimization

## 1 引言

批调度问题可以描述为将一组工件使用批处理机分批进行加工的问题,是现实生活中典型的组合优化问题,在社会生产的许多领域有着广泛的应用。如金属加工业中的热处理操作、集成电路板生产时进行的环境应力筛选(Environmental stress-screening)等。在这类问题中,如何有效地利用批处理机将显著影响批的完工时间。因此,对该类调度问题进行优化具有很强的现实意义。

## 2 常见算法回顾

批调度问题的解决主要包括:对工件集合进行排序、对工件序列分批以及最后按一定顺序进行批的加工。对于工件有尺寸的单机批调度问题,R. Uzsoy<sup>[1]</sup>研究了目标函数为最小化 $C_{max}$ 和 $\sum C_i$ 的情况,并结合装箱问题证明二者均为NP-hard,对该类问题的求解给出了若干启发式算法。

启发式算法属于贪婪算法,求解速度快,但对全局解搜索能力差。Melouk<sup>[2]</sup>和Damodaran<sup>[3]</sup>等分别采用模拟退火和遗传算法求解了本类问题。其求解思路是先对工件进行排序,然后采用启发式算法对其进行分批以求解最大完工时间。程八一等<sup>[4]</sup>引入了Metropolis准则的概率选择机制作为路径激励策略,以避免蚁群算法过早收敛的问题。Chen<sup>[5]</sup>等从聚类视角,通过定义批的浪费比提出CACB(constrained agglomerative clustering of batches)算法对该类问题进行了求解。本

文将采用改进的蚁群算法直接对工件分批,将排序和分批有机地结合在一起求解最小化最大完工时间 $C_{max}$ 的问题。对于批调度更为细致的文献回顾可以参见Potts<sup>[6]</sup>和Mathirajan<sup>[7]</sup>。

蚁群优化算法(Ant Colony Optimization, ACO)是由Colormi等人<sup>[8]</sup>于1991年提出的一种基于种群的元启发式算法(meta-heuristic),算法最初用于解决旅行商问题,在求解TSP以及工件排序等组合优化问题上均显示出很好的效果。蚁群优化算法将全局搜索与避免早熟收敛的机制结合在一起,有效提高了算法的性能。在基本蚁群算法的基础上,出现了多种改进形式<sup>[9]</sup>,算法参数设置非常灵活,不同参数对算法效果也将产生根本的影响<sup>[6]</sup>,这也是蚁群算法应用的困难之处。

## 3 问题描述

本文研究单机环境下,工件具有差异尺寸的最小化最大完工时间 $C_{max}$ 的批调度问题。该问题中,在不超过批容量约束下,多个工件可以作为一批同时进行加工。任一工件必须且只能属于一个批。同一批的工件,具有相同的开始、结束加工时间。从一批开始加工直到该批加工完成,该过程不允许中断,也不允许有工件中途退出或加入该批。批加工时间由批中加工时间最长的工件决定。

本文所研究批处理机调度问题的相关符号说明如下:

$i$ : 工件编号,  $i=1, 2, \dots, n$ ;

$N$ : 工件集合,  $N=\{1, 2, \dots, n\}$ ;

$p_i$ : 工件加工时间,  $i \in N$ ;  
 $s_i$ : 工件尺寸,  $i \in N$ ;  
 $S$ : 批处理机容量;  
 $B$ : 批集合;  
 $b$ : 组成一批的工件集合,  $b \in B$ , 且满足  $\sum_{i \in b} s_i \leq S$ ;  
 $|X|$ : 集合  $X$  中的元素个数;  
 $P^b$ : 批  $b$  的加工时间,  $P^b = \max\{p_i | i \in b\}$ 。

## 4 蚁群算法设计

### 4.1 解的构建

批调度问题中, 由于最终的解是形成批, 且同批中的工件无先后顺序, 因此在批调度问题中信息素的含义是指多个工件放在同一批中的期望度, 相应信息素的表达式如下:

$$\theta_b = \frac{1}{|b|} \sum_{j \in b} \tau_{ij} \quad (1)$$

式中,  $\tau_{ij}$  表示工件  $i$  与  $b$  中已有工件  $j$  之间的信息素。  $\theta_b$  表示工件  $i$  放入批  $b$  的期望度, 这个期望度定义为工件  $i$  与已安排在批  $b$  中的所有工件间信息素的平均值。在之后的转移概率计算中, 将使用此期望度来表示蚂蚁算法中解元素之间的信息素。

为了减少批数目, 显然应该提高批的利用效率。不考虑加工时间, 则本问题退化为装箱问题, 根据装箱问题的 FFD 算法, 在工件尺寸方面, 将工件  $i$  加入批  $b$  的启发性信息  $\mu_b$  定义如下:

$$\mu_b = s_i \quad (2)$$

由于批的加工时间只取决于批中加工时间最大的工件, 直观上应该将加工时间大的工件集中在较少的批, 即提高同一批中工件加工时间的一致性。因而在工件加工时间方面, 将工件  $i$  加入批  $b$  的启发性信息  $\eta_b$  定义如下:

$$\eta_b = \frac{1}{1 + |P^b - p_i|} \quad (3)$$

结合信息素以及启发式信息, 蚂蚁  $a$  从批  $b$  选择下一工件  $j$  的转移概率如式(4)所示:

$$P_{ij}^a = \begin{cases} \frac{\theta_{ij}^a \cdot \eta_{ij}^a \cdot \mu_{ij}^a}{\sum_{j \in V} \theta_{ij}^a \cdot \eta_{ij}^a \cdot \mu_{ij}^a}, & j \in V \\ 0, & \text{Else} \end{cases} \quad (4)$$

### 4.2 局部优化策略

对于任一个可行解  $S = \{b_1, b_2, \dots, b_k\}$ , 有  $p^{b_i} \geq p^{b_{i+1}}, i = 1, 2, \dots, k-1$ 。在  $b_{i+1}$  中取工件集合  $J' = \{j | \arg \max\{p_j\}\}$ , 在  $b_i$  中取工件集合  $J$ , 满足  $\sum_{j \in J} s_j \leq C - \sum_{j \in b_{i+1}} s_j - \sum_{j \in J'} s_j$ , 且  $\max\{p_j | j \in J\} \leq \max\{p_j | j \in J'\}$ 。交换  $J$  及  $J'$  得到邻域解  $S'$ , 从而有,  $p^{b_i} = p^{b'_i}$ , 且  $p^{b_{i+1}} \leq p^{b'_{i+1}}$ 。由于  $P^b = \max\{p_j | j \in b\}$ , 从而有  $C_{\max}^S = \sum_{i=1}^k P^{b_i} \geq C_{\max}^{S'}$ , 即  $S'$  相对于  $S$  更优。

依此得到局部优化算法的具体过程描述如下:

对于任一只蚂蚁得到的迭代解  $S^i$ ,  $S^i$  中批个数为  $n$ 。

①按批加工时间非增序排列  $S^i$  中各批, 对  $S^i$  中各批的工件按加工时间非增序排列;

②令  $k=0$ ;

③如果  $k=n$ , 则完成局部优化, 退出; 否则令  $k=k+1$ ;

④在满足机器容量约束的条件下, 将第  $k$  批中的  $b(p)$  工件与  $m(m \geq 0)$  个  $k-1$  批中加工时间小于  $b(p)$  的工件进行交

换, 如果  $m=0$ , 则直接将  $b(p)$  工件加入  $k-1$  批;

⑤如果第  $k$  批中工件为空, 则删除批  $k$ 。转至③。

### 4.3 算法流程

首先给出集合表示:  $A$  表示未被当前蚂蚁访问到的工件集合;  $B$  记录当前蚂蚁生成的批序列;  $V$  表示当前蚂蚁根据当前批序列的剩余空间从  $A$  中查找到的满足机器容量约束的可选工件集合。结合蚂蚁算法的一般框架, 本文中蚁群算法流程如下所示。

ACO 算法对问题求解流程如下:

```

Procedure ACO()
  初始化算法参数; //  $\alpha, \beta, \tau(0), \Delta, N_c, N_{cmax}$ 
  While( $N_c \leq N_{cmax}$ ) {
    For each(Ant a in  $\Delta$ ) {
      If( $A! = \Phi$ ) {
        If( $V! = \Phi$ ) {
          蚂蚁 a 构建新批;
          在 A 集合中随机选择一工件;
          在 V 集合计算转移概率, 选择下一工件加入当前批;
          更新蚂蚁 a 的 V, A, B 集合;
        }
      }
    }
    对迭代最优解进行局部优化;
    信息素更新;
  }
  输出最终调度方案;
}
  
```

## 5 仿真结果与分析

### 5.1 算例

本文根据 Melouk<sup>[2]</sup> 所使用的分类生成随机测试实例的方法进行仿真, 并与其他算法进行比较。实例的生成主要依赖 3 个因素: 工件数量 ( $J$ )、工件加工时间 ( $P$ ) 和工件尺寸 ( $S$ ), 并使用  $J_i, P_j, S_k (i=1, 2, 3, 4; j=1, 2, k=1, 2, 3)$  来表示各类实例。各因素具体取值如表 1 所列。

表 1 算例生成的分类因素及取值

因素	分类及取值
J	$J_1=10; J_2=20; J_3=50; J_4=100$
P	$P_1: U[1, 10]; P_2: U[1, 20]$
S	$S_1: U[1, 10]; S_2: U[2, 4]; S_3: U[4, 8]$

注: U 表示数据服从离散均匀分布。

### 5.2 参数设置

从实验可以得出如下结论: 在运行时间一定的情况下, 适量减少蚂蚁数而增加迭代次数, 可以增强信息素对算法的引导作用。显然, 算法的运行时间和蚂蚁数及迭代次数成正比。同时, 增大算法的搜索空间以及提高信息素的引导作用又可以提高算法的求解质量。综合考虑运行时间以及解的质量, 本文中设置蚂蚁数为 30, 迭代次数为 80,  $\rho=0.6$ 。

根据式(4)所示,  $\alpha, \beta, \gamma$  的绝对大小将影响选择概率值的集中程度。取  $\alpha=1$  为一个参数设置的基准, 对  $\beta, \gamma$  在  $[1, 10]$  之内进行讨论, 通过预实验得出最终的参数组合, 如表 2 所列。

表 2 启发信息系数设定

工件尺寸类型	$\beta$	$\gamma$
S1	4	5
S2	6	3
S3	3	6

### 5.3 实验结果与分析

本文在 Damodaran<sup>[3]</sup>研究的基础上将蚁群算法与该文中的 GA 相比较,以验证本算法的有效性。另外,由于本文采用的是蚂蚁算法的改进算法,因此也将其与基本的蚂蚁算法——蚁周算法 AC(ant-cycle)对本问题的求解做了比较。

表 3 是蚁群算法与其它算法运行结果的比较,包括了全部  $P_1$  类和  $P_2$  类工件在不同工件规模以及工件尺寸时的情况,表中的数字表示蚁群算法相对于其它算法在优于、等于和劣于这 3 方面所占的比例。整体而言,蚁群算法均优于其它算法,在  $J_3S_2$  类工件和  $J_4S_2$  类工件较大解空间的情况下,蚁

群算法出现个别解劣于其它算法。

我们通过计算蚁群算法相对于下界的接近程度  $K_L$ ,来说明蚁群算法相对于其它算法的改进程度。

$$K_L = \frac{C_{\max}(A) - LB}{LB} \times 100 \quad (5)$$

式中,  $A$  表示算法蚁群、AC、GA、BFLPT、FFLPT。  $C_{\max}(A)$  表示使用  $A$  算法得到的平均值。算法的改进度与  $K_L$  成反比,  $K_L$  值越小,则表明  $A$  算法求解出的  $C_{\max}$  越接近下界,改进度越大,算法效果越好。

图 1 显示了各算法相对于下界的平均改进度,从图中可以直观地看出蚁群算法相对其它算法的优劣效果。

表 3 蚁群与其它算法的运行结果优劣比例

		S1				S2				S3			
		AC	GA	BFLPT	FFLPT	AC	GA	BFLPT	FFLPT	AC	GA	BFLPT	FFLPT
J1P1	优	0.00	0.02	0.16	0.17	0.00	0.10	0.16	0.16	0.00	0.01	0.07	0.07
	等	1.00	0.98	0.84	0.83	1.00	0.90	0.84	0.84	1.00	0.99	0.93	0.93
	劣	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J2P1	优	0.04	0.45	0.47	0.59	0.01	0.31	0.32	0.32	0.00	0.07	0.14	0.20
	等	0.96	0.55	0.53	0.41	0.99	0.69	0.68	0.68	1.00	0.93	0.86	0.80
	劣	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J3P1	优	0.43	0.84	0.83	0.93	0.23	0.89	0.91	0.91	0.00	0.30	0.30	0.43
	等	0.57	0.16	0.17	0.07	0.75	0.11	0.09	0.09	1.00	0.70	0.70	0.57
	劣	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J4P1	优	0.78	0.93	0.90	1.00	0.49	0.99	0.99	0.99	0.05	0.18	0.18	0.30
	等	0.21	0.06	0.09	0.00	0.49	0.01	0.01	0.01	0.95	0.82	0.82	0.70
	劣	0.01	0.01	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J1P2	优	0.02	0.02	0.10	0.13	0.01	0.14	0.19	0.19	0.00	0.03	0.08	0.11
	等	0.98	0.98	0.90	0.87	0.99	0.86	0.81	0.81	1.00	0.97	0.92	0.89
	劣	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J2P2	优	0.13	0.42	0.41	0.57	0.06	0.32	0.33	0.34	0.00	0.14	0.19	0.29
	等	0.87	0.58	0.59	0.43	0.94	0.68	0.67	0.66	1.00	0.86	0.81	0.71
	劣	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J3P2	优	0.60	0.90	0.85	0.99	0.39	0.87	0.87	0.87	0.12	0.27	0.27	0.39
	等	0.40	0.10	0.15	0.01	0.60	0.13	0.13	0.13	0.88	0.73	0.73	0.61
	劣	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
J4P2	优	0.92	0.95	0.92	0.98	0.51	0.99	0.99	0.99	0.35	0.22	0.22	0.30
	等	0.05	0.04	0.06	0.02	0.43	0.00	0.00	0.00	0.65	0.78	0.78	0.70
	劣	0.03	0.01	0.02	0.00	0.06	0.01	0.01	0.01	0.00	0.00	0.00	0.00

从图 1 可以看出,工件数少时,算法改进度小,这是因为对于小数目工件,解空间小,各算法均易找到近优解,从算例输出的下界来看,各算法得出的结果均与下界接近。随着工件数增多,算法改进幅度变小。但从表 3 中的优劣工件的比例仍然可以看出,蚁群算法相对于其它算法有很大改进,图 1 中蚁群数据线均最接近横轴,也直观地说明了蚁群相对其它算法有较好的改进度。

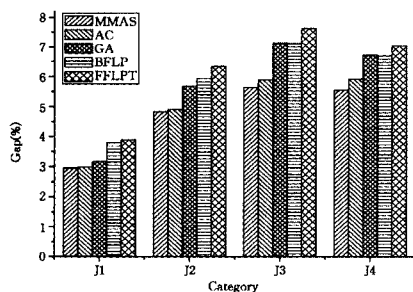


图 1 各算法相对于下界的平均改进度

**结束语** 针对工件有尺寸的单机批调度问题,本文使用蚁群算法对最小化最大完工时间进行了求解,对算法的参数

进行了测试和讨论,并将得出的结论应用于之后的仿真实验。文章对蚁群算法进行了改进,根据本文所研究问题的特征,设计了基于工件加工时间和工件尺寸的两类启发式信息,同时在算法中加入局部优化过程来改进迭代解的质量。实验中将该算法同蚂蚁算法(蚁周模型)、遗传算法、BFLPT 和 FFLPT 对本问题的求解结果结合下界进行了比较,结果表明该算法在不同工件规模的问题上均取得了较优的结果。

该算法也可调整、扩展应用到调度领域中的其它问题,例如对有工件交货期、工件动态到达的问题以及平行机等问题进行求解。

### 参考文献

- [1] Uzsoy R. Scheduling a single batch processing machine with non-identical job sizes[J]. International Journal of Production Research, 1994, 32(7): 1615-1635
- [2] Melouk S, Damodaran P, Chang P Y. Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing[J]. International Journal of Production

- [3] Damodaran P, Manjeshwar P K, Srihari K. Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms[J]. International Journal of Production Economics, 2006, 103(2): 882-891
- [4] 程八一, 陈华平, 王栓狮. 优化差异工件单机批调度问题的改进蚁群算法[J]. 系统仿真学报, 2009(9): 2687-2695
- [5] Chen H, Du B, Huang G Q. Scheduling a batch processing machine with non-identical job sizes: a clustering perspective[J]. International Journal of Production Research, 2011, 49(19): 5755-5778
- [6] Potts C N, Kovalyov M Y. Scheduling with batching: A review [J]. European Journal of Operational Research, 2000, 120(2): 228-249
- [7] Mathirajan M, Sivakumar A I. A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor[J]. International Journal of Advanced Manufacturing Technology, 2006, 29(9/10): 990-1001
- [8] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[C]//Proceedings of the First European Conference on Artificial Life. 1991
- [9] Dorigo M, Blum C. Ant colony optimization theory: A survey [J]. Theoretical Computer Science, 2005, 344(2/3): 243-278

(上接第 232 页)

由表 1 可见,当环境发生不同类型的动态变化时,本文改进型 QEA 在所有 30 次规划中,均能搜索到最优路径。未加交叉和变异算子的 QEA 及 ACO 在第 3 种环境中,均未能有效地搜索到最优路径,说明交叉和变异算子有助于增强 QEA 的局部搜索能力,同时,ACO 在复杂环境中的全局寻优能力有所欠缺,且 ACO 的整体搜索时间长于本文改进型 QEA。GA 的运行时间在 3 种算法中最短,但它的稳定性较差,在 30 次规划中,每次搜索到的路径差异较大,且当环境发生变化时,GA 难以有效地求得最佳路径。上述结果表明,与其它智能路径规划方法相比,本文改进型 QEA 具有更高的搜索效率、更强的稳定性和鲁棒性。

**结束语** 本文提出一种改进型的量子进化算法以解决机器人动态路径规划问题,利用 Matlab 图形用户界 GUI 建立了一个实时路径规划的监控界面,其与所提算法共同组成一个完整的路径规划系统,使系统的各项操作简单、直观且实用性强。对该系统的多次仿真结果表明,与其它路径规划方法相比,所提算法具有稳定性高、鲁棒性强的特点。当环境中的障碍物发生不同类型的动态变化时,它均能及时反应,重新规划出一条新的最优路径。该算法结合路径规划问题的特点,给出了新型的个体解码方式和路径修复算子,同时引入交叉和变异算子,与量子旋转门共同更新量子种群,这些改进措施均有效地提高了算法的搜索效率。综上所述,本文所设计的系统对于动态环境下的机器人实时路径规划问题具有一定的实用性和有效性。

### 参 考 文 献

- [1] Tsai C C, Huang H C, Chan C K. Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation[J]. IEEE Transactions on Industrial Electronics, 2011, 58(10): 4813-4821
- [2] Sun Yu, Zhang Ru-bo. Research on global path planning for AUV based on GA[J]. Advances in Intelligent and Soft Computing, 2012, 125: 311-318
- [3] 朱庆保, 张玉兰. 基于栅格法的机器人路径规划蚁群算法[J]. 机器人, 2005, 27(2): 132-136
- [4] Mohammad A K J, Mohammad H G, Eyad A F. Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field[J]. Soft Computing-A Fusion of Foundations, Methodologies and Applications, 2012, 16(1): 153-164
- [5] Chen Xu. Fast patrol route planning in dynamic environments [J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2012, 42(4): 894-904
- [6] 杨姗姗, 戴学丰, 唱江华. 实现机器人动态路径规划的仿真系统 [J]. 计算机工程与应用, 2009, 45(32): 237-239
- [7] 刘传领, 杨静宇. 一种基于量子染色体变异的移动机器人路径规划融合算法[J]. 信息与控制, 2011, 40(5): 594-598
- [8] Fu Yang-guang. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV[J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2012, 42(2): 511-526
- [9] Kim Y-H. Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot evolutionary computation[C]// Proceedings of IEEE congress on evolutionary computation. Trondheim, Norway: IEEE, 2009: 1185-1192
- [10] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 2002
- [11] 李士勇, 李盼池. 量子计算与量子优化算法[M]. 哈尔滨: 哈尔滨工业大学出版社, 2009
- [12] Han K-H, Kim J-H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization[J]. IEEE Transactions on Evolutionary Computation. 2002, 12(6): 582-584
- [13] 刘砚菊, 杨青川, 辜吟吟. 蚁群算法在机器人路径规划中的应用研究[J]. 计算机科学, 2008, 35(5): 263-265