

doi: 10.3969/j.issn.1006-1576.2010.06.011

蚁群算法求解工件尺寸不同的单机批处理问题

李端明¹, 程八一²

(1. 西南科技大学 经济管理学院, 四川 绵阳 621010; 2. 合肥工业大学 管理学院, 安徽 合肥 230009)

摘要: 针对工件尺寸不同的单机批处理调度问题, 提出以制造跨度为优化目标的蚁群算法。通过人工蚁对图进行遍历, 形成工件序列; 同时采用启发式规则对所得结果进行分批处理, 得到问题的解; 最后对不同复杂度的算例进行仿真。结果表明, 蚁群算法在解决工件尺寸不同的单机批处理问题时, 具备优异的性能。但对于不同工件尺寸的多机批处理问题, 还有待进一步研究。

关键词: 生产调度; 批处理机; 蚁群算法; 组合优化

中图分类号: TP301.6 **文献标识码:** A

Single Batch-Processing Machine with Non-Identical Job Sizes of Ant Colony Algorithm Solution Workpiece

LI Duan-ming¹, CHENG Ba-yi²(1. School of Economics & Management, Southwest University of Science & Technology, Mianyang 621010, China;
2. School of Management, Hefei University of Technology, Hefei 230009, China)

Abstract: Aiming at the single batch-processing machine with non-identical job sizes, an ant colony algorithm is proposed to make span. First, artificial ants are used to search the paths in the model to get a sequence of jobs and then heuristics are adopted to assign the jobs into batches, which form solutions of the problem. In the simulation part, different levels of instances are taken and the results show that the proposed algorithm is efficient in solving the problem. The scheduling of multi-machine with non-identical job sizes deserves further research in the future.

Keywords: Production scheduling; Batch-processing machine; Ant colony algorithm; Combinatorial optimization

0 引言

调度问题是工业领域重要的组合优化问题。古典调度问题包括单机、flowshop、openshop、jobshop等具体模型, 均要求在满足加工约束的前提下, 通过优化调度方案, 将待加工的工件按照时间顺序合理地分配到相应的机器上, 实现具体的调度目标。但古典调度问题没有涉及到工件尺寸与机器容量的约束, 但在实际的经济活动中却需要考虑这个空间上的约束条件, 例如半导体芯片的预烧、电路测试、港口货物装卸、金属加工等。

Uzsoy^[1]提出了工件尺寸不同的单机批调度问题 (Single Batch-processing Machine with Non-identical Job Sizes, NSBM), 证明了NSBM问题是NP完备的, 并设计了优化制造跨度和加权完工时间的启发式算法; 在启发式算法方面, Dupont和Jolai^[2]对算法进行了改进; Dupont和Flipo^[3]采用了分枝定界法优化制造跨度, 但对于规模较大的问题, 启发式算法的运算时间较长; Sevaux和Peres^[4]引入了遗传算法(Genetic Algorithm, GA), 对加权延迟工件数进行最小化; Kashan^[5]考虑了不同的编码

方式, 提出2种混合的GA: SGA和BHGA; Damodaran^[6]等利用分批规则产生初始种群, 然后重新设计了交叉算子和变异算子, 在制造跨度方面取得了较大的改进; Melouk^[7]则运用了模拟退火算法解决NSBM问题, 采用随机产生的方式获得不同规模的算例。故利用蚁群算法的分布式计算特点, 结合启发式规则来解决NSBM问题。

1 问题描述

NSBM问题的具体描述如下:

- 1) 工件集合为 $J = \{1, 2, \dots, n\}$, 其中, 工件 j 的加工时间为 t_j , 尺寸为 s_j ;
- 2) 机器容量为 B , J 中的工件分批加工, 任一批工件 b_k 中的所有工件的总尺寸不大于 B ;
- 3) 批的加工不允许中断, b_k 的加工时间 T_k 等于 b_k 中最后完工工件的加工时间, NSBM问题的制造跨度为所有批的加工时间总和。

根据上述约束条件, NSBM问题制造跨度的数学模型如下^[8]:

$$\text{Min } C_{\max} = \sum_{k=1}^m T_k \quad (1)$$

收稿日期: 2010-01-25; 修回日期: 2010-03-17

基金项目: 受国家自然科学基金项目“二维多机调度若干问题研究”(项目编号: 70671096)的资助。

作者简介: 李端明(1964-), 男, 广东人, 硕士, 副教授, 从事管理信息系统与商务智能研究。

$$s.t. \sum_{k=1}^m Y_{jk} = 1 \quad j=1,2,\dots,n \quad (2)$$

$$\sum_{j=1}^n s_j Y_{jk} \leq B_0 \quad j=1,2,\dots,n; k=1,2,\dots,m \quad (3)$$

$$T_k \geq t_j Y_{jk} \quad j=1,2,\dots,n; k=1,2,\dots,m \quad (4)$$

$$Tk \geq 0 \quad (5)$$

$$\sum_{j=1}^n \frac{S_j}{B_0} \leq m \leq n \quad (6)$$

其中, m 为分批之后的总批数; Y_{jk} 为 $0 \sim 1$ 变量, 当工件 $j \in b_k$ 时, $Y_{jk}=1$; 否则 $Y_{jk}=0$ 。式(1)表明模型的优化目标为制造跨度; 式(2)限定一个工件只能分配到一个批中; 式(3)即为机器容量的约束; 式(4)给出了批的加工时间的定义; 式(5)、式(6)为问题的基本约束。

2 蚁群算法

作为分布式算法, 蚁群算法在古典调度问题中已经有了较多的运用^[9-12]。与上述算法相比, 蚁群算法具备更好的鲁棒性和并行性^[13], 充分利用了群智能的信息交互优势。在搜索解的过程中, 人工蚂蚁会在经过的路径上设置一定浓度的信息素, 后面的人工蚂蚁在选择路径时, 将结合路径上的信息素和启发式信息 2 种因素进行判断, 从而有效地避免了非优秀解的重复, 提高了搜索效率。将引入蚁群算法, 对 NSBM 问题的制造跨度进行优化。

为了直观描述蚁群算法, 采用 NSBM 问题的图模型, 如图 1。在其他调度问题中, 图模型也被较多采用^[14-15]。在 NSBM 问题中, 顶点集 $V=(O \cup \sigma_0, \sigma_{n+1})$, 其中 $O=\{\sigma_1, \sigma_2, \dots, \sigma_n\}$, 对应工件集 J , 在此基础上, 增加 2 个虚拟结点 σ_0 、 σ_{n+1} , 加工过程的起点和终点, 加工时间均为零。从 σ_0 开始, 对全部结点 $\sigma_i (1 \leq i \leq n)$ 进行遍历, 到达 σ_{n+1} 之后, 形成有向非循环图。将工件序列进行分批, 得到问题的一个可行解。例如, 图 1 问题中, 工件数为 10, 按图示的访问结点的先后顺序得到解 $\pi=(\sigma_8, \sigma_4, \sigma_5, \dots, \sigma_{10}, \sigma_7)$ 。若分批之后形成 3 个批: $b_1=\{\sigma_8, \sigma_5, \sigma_3, \sigma_6\}$; $b_2=\{\sigma_4, \sigma_2, \sigma_1\}$; $b_3=\{\sigma_9, \sigma_{10}, \sigma_7\}$ 。则 π 的制造跨度为最后一个批 b_3 的完成时间, 即 $f(\pi)=T_1+T_2+T_3$ 。

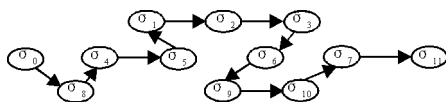


图 1 NSBM 问题的图模型示例

3 求解 NSBM 问题的蚁群算法

在算法执行之前, 设置人工蚁数量 m_0 , 初始位置在结点 σ_0 处。对人工蚁 $l (1 \leq l \leq m_0)$, 设置 $G(\sigma_l)$ 和 $S'(\sigma_l)$, σ_l 为 l 访问的当前结点, $G'(\sigma_l)$ 尚未访问的结点集, $S'(\sigma_l)$ 为下一步允许访问的结点集, 在访问 σ_l 之后, σ_l 从 $G'(\sigma_l)$ 和 $S'(\sigma_l)$ 中删除。当 $G'(\sigma_l) = S'(\sigma_l) = \emptyset$ 时, 人工蚁 l 该次搜索过程结束。人工蚁在经过的路径上, 留下一定浓度的信息素, 在所有人工蚁均完成该次搜索后, 各路径上的信息素将进行更新, 然后进入下一次迭代。当迭代次数达到设定值 I_m 时, 执行过程结束。

3.1 状态转移规则

在对结点的遍历过程中, 人工蚁从当前结点 σ_i 移动至后续结点 $(0 \leq i, j \leq n+1)$, σ_j 的选择依据下述的伪随机比例状态迁移规则 σ_j :

$$\sigma_j = \begin{cases} \arg \max_{\sigma \in S^l(\sigma_i)} q < q_0 \\ s & q \geq q_0 \end{cases} \quad (7)$$

其中, $q \in (0, 1)$, 为随机数, $q_0 \in (0, 1)$ 为预设的数, α 、 β 分别反映人工蚁在搜索时利用经验和独立探索的选择权重, $\tau(\sigma_i, \sigma)$ 为弧 (σ_i, σ) 上信息素浓度, $\eta(\sigma)$ 为 σ 上的启发式信息值, 等于对应工件加工时间的倒数。 $\alpha=0$ 时, 人工蚁完全按照工序的加工时间选择, 即加工时间最短的工序; $\beta=0$, 则人工蚁完全根据路径上的信息素浓度选择, 即其中浓度最高的路径。同时, 人工蚁以概率 $(1-q_0)$ 进行有偏向的搜索, s 为按一定概率进行选择的随机结点, 人工蚁选择路径 (σ_i, s) 的概率为:

$$P^l(\sigma_i, s) = \begin{cases} \frac{(\tau(\sigma_i, s))^\alpha (\eta(s))^\beta}{\sum_{u \in S^l(\sigma_i)} ((\tau(\sigma_i, u))^\alpha (\eta(u))^\beta)} & s \in S^l(\sigma_i) \\ 0 & \text{else} \end{cases} \quad (8)$$

3.2 分批规则

在人工蚁遍历所有结点后, 得解 $\pi=(\sigma_0 \dots \sigma_{n+1})$, 其中结点 σ_i 对应工件 J_i , π 的分批根据下述规则:

步骤 1: 若已经产生了 k 个批 b_1 、 $b_2 \dots b_k$, 其中没有分配完成的批(即批中的工件总尺寸小于 B) 为 b_1 、 $b_2 \dots$, 当前待分批的结点为 σ_i 。则将 σ_i 按升序将纳入 b_j , 若纳入之后该批的总尺寸仍然不超过 B , 则 $\sigma_i \in b_j$; 若所有 b_j 均不能容纳 J_i , 则新建批 b_{k+1} , $\sigma_i \in b_{k+1}$;

步骤 2: 对所有 $\sigma_i(1 \leq i \leq n)$, 依次按步骤 1 分批, 从而得出最终的分批方案。

例如, 表 1 中 $B=10$, $|J|=10$, 根据解 π , J_i 的排列顺序如第一行, 第二、三行为工件的加工时间和尺寸。根据上述规则, 分批结果见表 2。

表 1 10 工件问题示例

工件号	5	1	2	10	8	4	9	6	3	7
t_j	15	12	9	6	3	10	18	6	4	9
s_j	5	7	2	9	6	3	4	3	2	4

表 2 工件的分批结果

bk	j(sj)	T_j	批的实际尺寸
b1	5(5),2(2),4(3)	15	10
b2	1(7),3(2)	12	9
b3	10(9)	6	9
b4	8(6),9(4)	18	10
b5	6(3),7(4)	9	7

根据表中结果, 可得出制造跨度 $f(\pi) = \sum_{k=1}^4 T_k = 26$ 。

3.3 信息素更新

人工蚁在搜索的过程中, 路径上的信息素发生局部更新, 更新规则为:

$$\tau(\sigma_i, \sigma_j) = (1 - \rho)\tau(\sigma_i, \sigma_j) \quad (9)$$

其中, ρ 为信息素挥发系数, 笔者采用动态 ρ 值的设置: $\rho = k\tau(\sigma_i, \sigma_j)$, ($k \geq 0$, 为常数)。在所有人工蚁到达 σ_{n+1} 之后, 首先更新全局最优解 π^* , π^* 为目前所有解中制造跨度最小的解, 然后对信息素进行全局更新:

$$\tau(\sigma_i, \sigma_j) = \tau(\sigma_i, \sigma_j) + \Delta\tau(\sigma_i, \sigma_j) \quad (10)$$

其中:

$$\Delta\tau(\sigma_i, \sigma_j) = \begin{cases} \frac{Q}{f(\pi^*)} & (\sigma_i, \sigma_j) \in \pi^* \\ 0 & else \end{cases} \quad (11)$$

Q 为正的常数。在上述的信息素更新规则下, 人工蚁会逐渐向优秀解的路径上集中, 直至迭代过程结束。

4 实验

NSBM 问题的算例, 目前较多根据 Melouk 等提出随机产生方法获得。根据问题规模的大小, 按 3 个标准进行划分: 工件数量 n 、工件处理时间 t_i 、工件尺寸 s_i , 其中 t_i 、 s_i 取整数, 取值均服从离散的平均分布, 划分依据为取值的分布区间, 见表 3。

对各个规模的问题均进行实现, 运行环境为 Visual C++ 6.0。将表 4 中蚁群算法和 Melouk 的模拟退火算法的运行结果进行对比, 表 4 中数值为制造跨度, 平均解采用 10 次实验的平均值。蚁群算法的参数设置为 $m_0 = 200, q = 0.8, Q = 1, k = 0.1, I_m = 300$ 。

表 3 仿真算例的划分

参数	规模			
	J1	J2	J3	J4
n 取值	10	20	50	100
t_j	t1		t2	
t_j 分布区间	[1,10]		[1,20]	
s_j	s1	s2	s3	
s_j 分布区间	[1,10]	[2,4]	[4,8]	

表 4 J1 类问题的结果比较

问题分类	模拟退火算法			蚁群算法			改进 (%)	
	最好	平均	最差	最好	平均	最差	最好	平均
J1t1s1	30	33	37	30	30	30	0	10
J1t1s2	20	21.6	24	20	20	20	0	8
J1t1s3	52	52	52	52	52	52	0	0
J1t2s1	46	47.7	57	46	46	46	0	3.7
J1t2s2	34	36.4	47	34	34	34	0	6.6
J1t1s1	86	86	86	86	86	86	0	0

表 4 为 J1 类问题的实验结果, 其中最后一列数据为蚁群算法的解在模拟退火算法基础上的改进。可见, 与后者相比, 蚁群算法在 J1 类问题中, 均能在 300 次迭代之内获得最优解。

在 J1 以上的问题中, 其实验结果如表 5, 蚁群算法的制造跨度均优于模拟退火算法。与所得的最优解相比, 蚁群算法在平均解上的改进程度更大, 表明解的分布相对集中, 收敛性能更好。算法的时间性能如表 6, 可见, 在所有算例的运行中, ACO 比 SA 的求解时间更短。

表 5 J1 以上规模问题的实验结果比较

问题分类	模拟退火算法			蚁群算法			改进 (%)	
	最好	平均	最差	最好	平均	最差	最好	平均
J2t1s1	71	72	77	71	71	71	0	1.4
J2t1s2	42	45.8	50	42	42	42	0	9
J2t1s3	73	74.5	79	72	73.1	75	1.4	3.4
J2t2s1	107	115.7	122	107	109.3	112	0	5.5
J2t2s2	72	76.8	85	72	73.5	78	0	4.3
J2t2s3	149	149	149	149	149	149	0	0
J3t1s1	169	178.6	198	165	168.2	171	2.4	5.9
J3t1s2	97	102.6	111	93	96.4	101	4.1	6
J3t1s3	182	183.4	184	179	180.3	183	1.6	1.7
J3t2s1	324	346	414	322	335	356	0.6	3.2
J3t2s2	204	215.7	245	201	207.4	215	1.5	3.8
J3t2s3	378	386.6	428	378	379.3	342	0	1.9
J4t1s1	281	287.7	300	277	279.1	284	1.4	3
J4t1s2	207	212.1	221	194	197.5	205	6.3	6.9
J4t1s3	394	402.2	426	387	390.1	395	1.8	3
J4t2s1	610	637.4	708	608	613.2	621	0.33	3.8
J4t2s2	350	359.9	368	344	346.2	348	1.7	3.8
J4t2s3	804	832.3	858	796	803.1	809	1	3.5

表 6 算法运行时间对比

算例	平均运行时间(s)		算例	平均运行时间(s)	
	SA	ACO		SA	ACO
J1t1s1	1.84	1.27	J2t1s2	13.76	3.19
J1t1s2	2.14	1.15	J2t1s3	11.25	3.17
J1t1s3	1.36	1.29	J2t2s1	10.24	3.09
J1t2s1	3.22	1.4	J2t2s2	6.71	3.13
J1t2s2	2.25	1.49	J2t2s3	11.74	4.07
J1t2s3	1.33	1.29	J3t1s1	173.74	13.24
J2t1s1	11.42	3.32	J3t1s2	114.09	11.09

(下转第 40 页)

态来设定系统充电的主电源或满充还是半充（半电流充电最高电流为满电流充电的一半），如表 2。

表 2 充电电流设置

PESL	ISET2	充电电流
高	高	满充
高	低	半充
低	高	最高充电电流为 500 mA
低	低	最高充电电流为 100 mA

为更好地管理和控制锂电池与系统负载的关系，该电路同时具备充电定时设定功能、睡眠模式和备用模式、自动充电电源选择功能和电池温度的监视调整功能。

5 结束语

根据 BQ24032 芯片的功能特点，结合锂电池的充电要求，对基于 BQ24032 的锂电池充电管理电路进行了研究设计，分析了 BQ24032 充电和电源管理

(上接第 36 页)

5 结论

仿真结果表明，蚁群算法相对模拟退火算法有着较大优势。但由于随着机器数的增加以及加工约束的改变，在多机环境下，不同工件的批处理问题复杂度将加大，要对大量可行解进行搜索，故需进一步改善搜索的质量，减少路径的重复，同时避免非可行解的出现。分批规则也必须针对多台机器的总体加工情况，做出有效的改进。因此，对于不同工件尺寸的多机批处理问题，还将做进一步研究。

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] Dupont, L., Jolai Ghazvini, F., Minimizing makespan on a single batch processing machine with non identical job sizes[J]. European journal of Automation Systems, 1998, 32(4): 431-440.

[3] Dupont, L., Dhaenens-Flipo, C.: Minimizing the makespan on a batch machine with nonidentical job sizes: an exact procedure[J]. Computers & Operations Research.2002, 29(7): 807-819.

[4] Sevaux, M., Peres, S.D. Genetic algorithms to minimize the weighted number of late jobs on a single machine[J]. European Journal of Operational Research, 2003, 151(2): 296-306.

[5] Koksalan, M., Keha, A.B. Using genetic algorithms for single machine bicriteria scheduling problems[J]. European Journal of Operational Research, 2003, 145(3): 543-556.

[6] AH Kashan, B Karimi, F Jolai. Minimizing Makespan on a Single Batch Processing Machine with Non-identical Job Sizes: A Hybrid Genetic Approach[C]. Evolutionary Computation in Combinatorial Optimization-6th European Conference, 2006, Springer Berlin: Proceedings, Lecture

的主要过程，给出了典型应用电路的设计方案。实验表明，该充电电路在给一组锂电池充电的同时还能给系统负载供电，并且可在充电电流和系统供电电流之间进行动态的调整，确保了系统工作的稳定性。

参考文献：

[1] 贾英江, 傅孝忠, 王耀济, 等. 锂电池充电方法分析[J]. 科技资讯, 2009(2):123.

[2] 张庆, 李莘臣. 锂离子电池充放电特性的研究[J]. 自动化技术与应用, 2008, 27(12): 107-109.

[3] 郑如定. 锂离子电池和锂聚合物电池概述[J]. 通信电源技术, 2002(5): 18-21.

[4] 贾颖, 黎火林. 锂离子电池失效率模型的探讨[J]. 北京航空航天大学学报, 2008, 34(8): 973-975.

[5] TEXAX INSTRUMENTS. SINGLE-CHIP CHARGE AND SYSTEM POWER-PATH MANAGEMENT IC (BQTINY™-III). <http://www.ti.com>, 2005.

Notes in Computer Science 3906: 135-146.

[7] Purushothaman Damodaran, Paraveen Kumar Manjeshwar, Krishnaswami Srihari.. 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.

[8] Melouk, S., Damodaran, P., Chang, P.-Y., 2004. Minimizing makespan for single machine batch processing with nonidentical job sizes using simulated annealing[J]. International Journal of Production Economics 2004, 87(2): 141-147.

[9] Blum C, Sampels M. An ant colony optimization algorithm for shop scheduling problems[J]. Journal of Mathematical Modelling and Algorithms, 2004, 3(3): 285-308.

[10] J. Heinonen, F. Pettersson. Hybrid ant colony optimization and visibility studies applied for job shop scheduling problem[J]. Applied Mathematics and Computation. In Press. Available online 18 October 2007, 187(2): 989-998.

[11] Kuo-Ling Huang, Ching-Jong Liao. Ant colony optimization combined with taboo search for the job shop scheduling problem[J]. Computers & Operations Research, 2008, 35(4): 1030-1046.

[12] Zhang J, Hu X, Zhong J H, Huang Q. Implementation of an Ant Colony Optimization technique for job shop scheduling problem[J]. Transactions of the Institute of Measurement and control, 2006, 28(1): 93-108.

[13] Colorni A, Dorigo M, Maniezzo V, Trubian M. Ant system for job shop scheduling[J]. Belgian Journal of Operations Research, 1994, 34(1): 39-53.

[14] Blazewicz J, Presch E, Sterna M.B. Disjunctive graph machine representation of the jobshop scheduling problem[J]. European Journal of Operational Research, 2000, 127(2): 317-331.

[15] Felix T. S. Chan and H. K. Chan. A Comprehensive Survey and Future Trend of Simulation Study on FMS Scheduling[J]. Journal of Intelligent Manufacturing, 2004, 15(1): 87-102.

[16] 高桂清, 王汉坤, 武建, 等. 基于模糊理论及蚁群算法打击任务量化限[J]. 四川兵工学报, 2009(10): 1-3.