基于正交实验的蚁群算法在车间调度问题中的应用①

张晓玲　杨　健　杜英国 (大理学院 数学与计算机学院 云南 大理 671003)

摘　要： 提出用正交实验的方法来设置蚁群算法在求解车间调度问题的参数。蚁群算法在求解车间调度问题时的性能大部分依赖于参数的设置，各参数的值不同，则蚁群算法的收敛速度和得到的解也不同，使用正交实验的方法来测试各个参数对蚁群算法性能的影响，通过对实验结果的分析可得出参数的最佳组合方案。用经典的 JSP 的样例对这种组合方案进行了测试，实验结果表明用正交实验法得到的蚁群算法的参数设置方案可以加快算法的收敛速度，使算法能够得到问题的最优解或近似最优解。

关键词： 蚁群算法(Ant Colony Algorithms)；车间调度问题(J0b Shop Scheduling Problem)；正交实验；正交表

An Ant Colony Algorithm Based on Orthogonal Experimental Method for Job-Shop Scheduling

ZHANG Xiao-Ling, YANG Jian, DU Ying-Guo

(Department of Computer Science and Mathematics, Dali University, Dali 671003, China)

Abstract: In this paper, an orthogonal experimental method is used to set the parameters of Ant Colony Algorithm for solving Job-Shop Scheduling Problem. The performance of Ant Colony Algorithm for Job-Shop Scheduling mostly depends on the parameters settings, the different values of each parameter, the different convergence rates and the solution of Ant Colony Algorithm. The orthogonal experimental method is used to test the parameters on the performance of Ant Colony Algorithm. The best combination of parameters can be drawn by analyzing the experimental results. This best combination of parameters has been tested based on the cases from the JSP classic. The results demonstrate the parameters setting can speed up the convergence rate and the optimal solution or the approximate optimal solution can be achieved.

Keywords: ant colony algorithm(ASA); job-shop scheduling problem (JSP); orthogonal experiment; orthogonal table

1　引言

1.1 文章安排

本文讨论的是用蚁群算法(Ant Colony Algorithms)求解 JSP(Job Shop Scheduling Problem)，然而 ACA 中参数的设置是很重要的，参数的设置直接决定了求解结果的好坏。因此，采用正交实验的方法来确定蚁群算法在解 JSP 时候的最优参数组合，以使蚁群算法求解 JSP 得到最好的效率。本文的以下内容安排如下，第 2 部分给出 JSP 的描述，第 3 部分给出了蚁群算法求解 JSP 的整体框架，第 4 部分讨论用正交实验法确定试验指标的最优参数组合，并讨论其有效性。第 5 部分通过经典的 JSP 问题验证算法的优势。第 6 部分进行总结。

1.2 基本介绍

车间调度问题(Job-Shop Scheduling Problem)是在经典的调度理论[1]中所描述的最难的组合问题中的一种，是典型的 NP 难(非确定性多项式时间难)问题。已经有许多方法用来解决 JSP，包括线性规划、分支界定法、约束满足、局部搜索、神经网络[2]、专家系统、遗传算法[3,4]、禁忌搜索[5]，模拟退火[6]和蚂蚁系统[7]等等。

蚁群算法(Ant Colony Algorithm)是由意大利学者 M.Dorigo 在 20 世纪 90 年代提出的一种模仿自然界蚂蚁群体的觅食行为的一种分布式进化计算方法[8]，具有并行性、算法简单和求解效率高等特点，并且已

① 收稿时间:2009-07-20;收到修改稿时间:2009-08-22

经被广泛的应用。蚁群算法模拟了自然界蚂蚁寻找从蚁巢到食物源的最短路径并找到回蚁巢路径的机制，蚂蚁之间通过一种信息素(“ 激发工作 ”:stigmergy)[9]来相互交流信息。蚂蚁在移动的过程中，会在经过的路径上散发信息素，蚁群通过感知路径上的信息素进行通信，个体之间并不直接进行交互，而是通过改变他们共同存在的环境进行交互，个体又通过对环境的改变去影响其他个体的行为，从而形成了一种正反馈机制。长度越短的路上，经过的蚂蚁越多，聚集的信息素也越多，蚂蚁趋向于选择信息素多的路径移动。通过这种机制，经过一段时间后蚂蚁最终能够发现最短路径。蚁群算法已经被成功的应用于解决组合优化问题，如：TSP(Traveling Salesman Problem), QAP (Quadratic Assignment Problem)，JSP(Job_shop Scheduling Problem)等。

2 车间调度问题(Job shop scheduling problem)

2.1 问题的描述

JSP 问题可以描述为：有 n 个加工顺序不同的工件要在 m(m>2)台机器上完成加工。

已知：工件集 $J=\{J_1，J_2，\cdots，J_n\}$，J_i为第 i 个工件，$i=1,2,\cdots,n$；机器集 $M=\{M_1，M_2，\cdots，M_m\}$，M_j为第 j 号机器，j=1，2，…，m；每个工件有 k 道工序，每道工序要求在不同的机器上加工，并且预先已经定义了每个工件在 m 台机器上的加工顺序，u_{ij}表示工件 i 在机器 j 上加工，工序序列集 $O=\{u_{ij}\,|\,i\in[1,n],\ j\in[1,m]\}$，其中 n 为工件的数目，m 为机器的数目。

每个工件使用每台机器的时间矩阵 T，$t_{ij}\in T$为第 i 个工件 J_i使用第 j 台机器的时间。$t_{ij}=0$表示工件 J_i不使用机器 M_j。

$c_{ij}=c_{ik}+T_{ij}$是工序 u_{ij}相对于关系 $u_{ik}\rightarrow u_{ij}$ 的完整的执行时间，调度目标：求出可行的工序的加工序列，使其最后一道工序的最大完工时间 $c_{\max}$ 最小。

$$C_{max}=\max_{all\,u_{ij\in O}}(c_{ij})=\max_{u_{ik}\rightarrow u_{ij}}\left(c_{ik}+T_{ij}\right) \tag{1}$$

2.2 JSP 的表示

可以用析取图 $D=(N,A,B)$来表示 JSP 问题，N 是所有结点的集合，每个结点对应一个工件的一个工序 u_{ij}，并且 N=m*n。A 是工件使用机器的加工顺序的有向弧的集合。B 是连接在同一台机器上处理的不同的工件的工序的无向弧的集合。为了确定先对哪个工件进行加工和什么时候结束，需要设置一个初始结点 O_0和调度结束的结点 O_{10}。以 3 个工件 3 台机器的问题作为例子，见图 1。

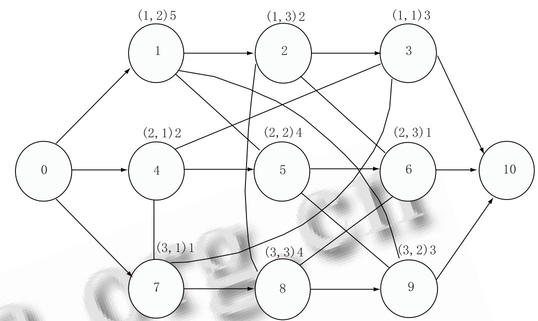


图 1　3/3/G/CmaxJSP 的表示图，其中 $0=O_0$, $10=O_{10}$,1=u12,2=u13,3=u11,4=u21,5=u22,6=u23,7=u31,8=u33,9=u32

在图 1 中，结点 1 旁边的(1,2)5 表示工件 1 的第一道工序在机器 2 上处理，处理时间为 5，图中除与 O_0 和 O_{10} 相连的有向弧，其他的有向弧表示同一个工件的加工顺序，工序必须按照该顺序进行加工，其他的为无向弧。每一行加工一个工件，1,2,3 加工工件 1，4,5,6 加工工件 2，7,8,9 加工工件 3。每个工件只能使用每台机器一次，有 m 台机器，则每个工件就有 m 道工序。因此求解 JSP 就变成在满足约束的前提下，求得上图中结点的一个序列，例如{4,1,7,8, 2,3,5,9,6},求出在每台机器上加工完毕所有的工件的最后一道工序时的最大完工时间，求解目标：求出最大完工时间最小的序列，即为调度的最佳序列。

3 蚁群算法求解JSP

根据图 1，蚁群算法(ACA)求解 JSP 的基本工作过程描述如下：

由于在整个工件调度的过程中添加了一个开始的结点 O_0 和结束的结点 O_{n+1}，因此所有的蚂蚁从结点 0(工序 0)出发，最后到达的结点是结点 10。设集合 Gk 是没有访问过的结点的集合，Sk 表示根据约束规则下一步允许访问的结点的集合，还需要一个禁忌表 Jk，存放已经访问过的结点。在上面的例子中，Gk={1,2,3,4,5,6,7,8,9}，Sk={1,4,7}。首先蚂蚁分布在 0 结点，然后每个蚂蚁根据状态转

移规则选择下一个结点，蚂蚁倾向于选择那些加工时间较短且信息素强度较高的路径，图中的每个弧表示节点间信息素的量和启发式距离的一对值{ α_{ij},T_{ij}}，T_{ij} 通常为对工件 j 的第 i 步工序的加工时间。每选择一个结点，该结点就被追加到禁忌表 Jk 中并从 Gk 中删除：如果被选的结点不是工件的最后一步，那工件中紧邻的下一个结点就会被加入到 Sk 中。单个的蚂蚁在遍历过程中根据信息素局部更新规则在路径上释放一定数量的信息素，同时蚂蚁经过的路径上的信息素随着时间的推移而蒸发。当所有的蚂蚁都完成了它们的遍历过程之后，Gk＝φ。最后禁忌表中得到的结点的排序序列就是蚂蚁 k 找到的解。此时可以计算出此次迭代的最小的最大完工时间，再应用信息素全局更新规则对获得最小的最大完工时间的蚂蚁所经历的边上的信息素进行更新。此后算法反复迭代直至满足终止条件后结束。可见，蚁群算法的求解过程主要由三个规则控制，即状态转移规则，信息素全局更新规则和信息素局部更新规则。下面将对这三个规则进行说明。

3.1 状态转移规则

ACS 中的的状态转移规则(又叫做“ 伪随机比例规则 ”),“ 伪随机比例规则 ”为在保持探索(exploration)一条新的边和利用蚂蚁所积累的搜索经验开发(exploitation)一条边之间的平衡提供了一种直接的方式，状态转移规则如下：位于结点 r 的蚂蚁 k 使用以下的规则选择结点 s 为下一个要访问的结点：

$$s=\begin{cases} \arg\max_{u\in S_k}\{[\tau(r,u)]\times[\eta(u)]^{\beta}\}, & if\ q\le q_0 \\ S, & otherwise \end{cases} \quad (2)$$

在式(2)中，τ 是信息素，η 是启发函数，$\eta(s)=1/T(s)$ 是在结点 s 的工序在机器上的处理时间的倒数，S_K 是蚂蚁 k 还没有访问过的结点的集合，而 $\beta>0$ 是一个参数，用来定义信息素和时间之间的相对重要性。q 是一个均匀分布的随机数，$q\in[0,1]$，$0\le q_0\le 1$ 是自定义的一个参数。S 是根据以下的概率公式(3)计算得来的随机变量。

$$p_k(r,s)=\begin{cases} [\tau(r,s)]\times[\eta(s)]^{\beta} \Big/ \sum_{u\in S_k}[\tau(r,u)]\times[\eta(u)]^{\beta} & if\ s\in S_k \\ 0 & otherwise \end{cases} \quad (3)$$

伪随机比例规则使得蚂蚁倾向于选择加工时间短且信息素强度浓的路径。参数 q_0 决定了“ 探索 ”和“ 开发 ”之间的相对重要性。当位于结点 r 的蚂蚁 k 要选择下一个将要访问的结点时，它选择一个随机数 $0\le q\le 1$，如果 $q\le q_0$ 则按照式(2)根据启发信息和信息素强度取最好的边，否则，按照式(3)伪随机比例规则选择下一条要移动的边。

3.2 信息素全局更新规则

在蚁群算法中，当所有的蚂蚁都访问完所有的结点以后，只有那些属于最小的最大完工时间的边上的信息素才被得到增强，信息素全局更新规则的使用使算法的寻优过程更具有指导性：最小的最大完工时间寻找始终是在当前找到的最小完工时间的附近进行。按以下的公式(4)对最小的最大完工时间的边上的信息素进行更新。

$$\tau(r,s)\leftarrow(1-\alpha)\times\tau(r,s)+\alpha\times\Delta\tau(r,s) \quad (4)$$

其中

$$\Delta\tau(r,s)=\begin{cases} 1/T_{gb} & if\ (r,s)\in global\ best\ tour \\ 0 & otherwise \end{cases},$$

$0<\alpha<1$，是信息素衰减系数，而是蚂蚁 k 在本次循环中找到的最小的最大完工时间。

3.3 信息素局部更新规则

单个的蚂蚁在遍历过程中按照公式(5)信息素局部更新规则对它所经过的边上的信息素进行更新。

$$\tau(r,s)\leftarrow(1-\rho)\times\tau(r,s)+\rho\times\tau_0 \quad (5)$$

$0<\rho<1$ 是信息素挥发系数，是预先设置的信息素的初始值，信息素局部更新规则使得被蚂蚁经过的路径减少了一部分信息素，使得这些边被后来的蚂蚁经过的可能性减少，增强了算法的“ 勘探 ”能力，从而有效的避免算法进入停滞状态(所有的蚂蚁收敛到同一个解)。

3.4 蚁群算法求解 JSP 的算法描述

假设以下是求 n 个工件 m 个机器的 JSP 问题，蚂蚁的个数为工件的个数，则下图 2 为求解 JSP 的蚁群算法的流程图。

4 基于正交试验法的参数设置

在蚁群算法中，参数α和β决定了算法的收敛速度并对解的性能好坏有重要的影响，同时信息素的挥发系数 ρ 也需要进行适当的设置以使搜索能在好的搜索空间中进行，并防止陷入局部最优的邻域中。

正交试验法，就是指运用正交表来安排试验方案和进行结果分析的一种试验设计方法。正交表是根据均衡分布思想，运用组合数学理论构造的一种数学表

格，具有正交性、典型性以及综合可比性等优点，它适用于多因素、多指标具有随机误差的试验。通过正交试验，可以分析各因素对试验指标的影响，按照其重要程度找出主次关系，并确定对试验指标的最优参数组合。

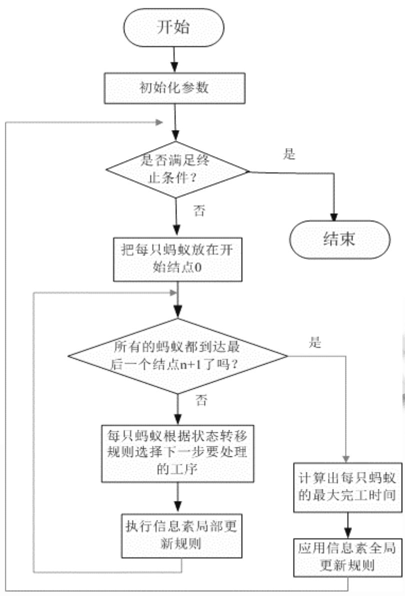


图 2　蚁群算法求解 JSP 的流程图

本文考虑影响试验结果的 4 个因素：信息素衰减系数 α、决定信息素和处理时间的相对重要性的因素 β、信息素挥发系数 ρ 和阀值 q_0，以 FT03 问题为例，采用四因素、三水平的正交表 $L_9(3^4)$，具体数据如表 1 所示。

表 1　因素水平正交表

因素 / 水平	衰减系数α A	相对参数β B	挥发系数ρ C	阀值 q_0 D
1	0.1	0.0	0.01	1.0
2	0.2	1.0	0.02	0.9
3	0.3	2.0	0.03	0.8

采用以下信噪比的公式(6)，对试验数据进行计算分析，

$$S/N = -10\log\left[\frac{1}{n}\sum_{i=1}^{n} y_i^2\right] \qquad (6)$$

其中 y_i 为经过 n 次参数的组合得到的最小的最大完工时间，各参数如{A1,B1,C1,D1},{A1,B1,C1,D2},{A1,B1,C1,D3}，…，下一次则变化 C 的值，每个值要对应 9 次实验。如此下去，从而得到各如表 2 的试验数据

表 2　正交试验设计数据分析表

因素 / 水平	A	B	C	D
1	−91.8205	−91.2353	−91.1782	−91.3366
2	−91.9753	−91.2137	−91.2009	−91.2008
3	−91.9096	−91.1890	−91.2589	−91.1006
极差	0.1629	0.0463	0.0807	0.2360
最优方案	A1	B3	C1	D3

通过各因素对试验的最后结果(最小的最大完工时间)的影响进行分析，得到最好的方案为 α＝0.1，β＝2.0，ρ＝0.01 和 q0=0.8，对此方案进行试验，表明该方案比其他的方案的结果都要好。

5　实验仿真与结果分析

5.1　实验仿真

本文采用了 4 个经典的 JSP 问题：FT03，FT06，LA06 和 LA07 的测试用例来说明用正交试验的结果来进行蚁群算法的参数设置的优势。

由于样例的规模不同,本文通过正交实验得到的蚁群算法的最优参数的组合也不同，具体得到的最优参数设置见下表 3。

表 3　通过正交实验得到的最优参数设置

Test cases	β	α	ρ	q_0
FT03	2.0	0.1	0.01	0.8
FT06	0.0	0.1	0.1	0.8
LA06	2.0	0.1	0.01	0.8
LA07	2.0	0.1	0.01	0.8

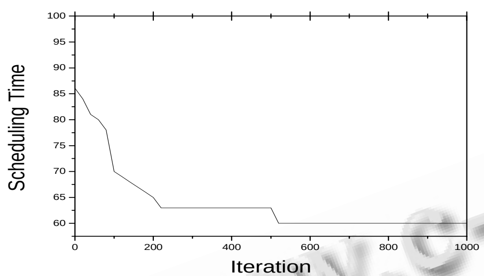
为了降低由于系统和统计的误差,本文对求解的每个用例都进行了 10 次的仿真,实验结果是 10 次仿真的统计值(平均最优值和方差等)。

5.2　结果比较

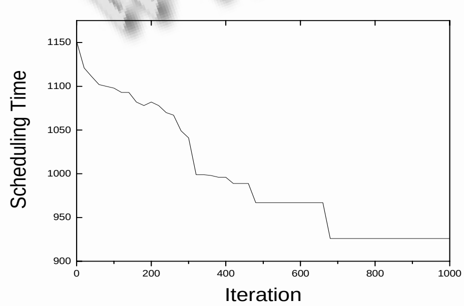
表 4 给出了使用以上表 3 的正交实验的结果进行参数设置的蚁群算法求解四个样例的结果，表中 n 是工件的数目，m 是机器的数目，“ OPT ”一列是样例的已知的最优解.为了能更加直观地反映这种参数组合的优势，图 3 给出了 FT06，LA06 和 ABZ6 在优化问题的时候的进化特征曲线。

表 4　ACA 求解不同 JSP 的平均结果

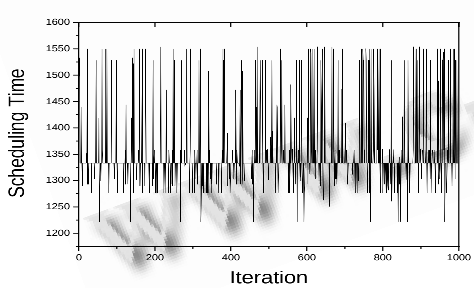
Test cases	n	m	Average	ACA_best	OPT
FT03	3	3	12	12	12
FT06	6	6	59.5	55	55
LA06	15	5	1034	926	926
ABZ6	10	10	1245	1154	—



(a)　FT06



(b)　LA06



(c)　ABZ6

图 3　ACA 求解不同 JSP 的进化特征曲线图

在图 3 中可以看出 ACA 在求解 JSP(FT06,LA06 和 ABZ6)时用了 1000 次迭代的收敛速度，一般都在 500 次迭代左右就可以收敛(ABZ6 问题的最优解出现在每次迭代中)。虽然表 4 和图 3 都体现出了正确的参数设置对提高蚁群算法搜索能力的高效性,但是本文提出的算法也存在工件数目越多，算法的时间复杂度越大的问题。对于蚁群算法解复杂的 JSP 的时间消耗，我们还需要对蚁群算法进行改进或融入其他的一些算法，以使算法能够在求得最优解的基础上时间消耗最小。

6　结语

本文针对蚁群算法(ACA)求解 JSP 的特征而提出了一种用正交实验的方法来进行蚁群算法的参数设置。该算法通过正交实验来取得蚁群算法在求解 JSP 时候的最优参数组合，从而加快算法的收敛速度，使算法能尽快找到最优解或近似最优解。本文对该算法的基本原理进行了阐述并给出了具体的实现方案，最后通过典型的 JSP 测试样例进行实验仿真，结果证明了该策略不但在收敛速度上而且在求解精度上都有很大的优势。

参考文献

1 Garey MR, Johnson DS, Sethi R. The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research, 1976, 1:117－29.

2 Huang DS, Ma SD. Linear and nonlinear feedforward neural network classifiers: a comprehensive understanding. Journal of Intelligent Systems, 1999,9:1－38.

3 Della CF, Tadei R, Volta, G. A genetic algorithm for the job shop problem. Computers & Operations Re- search, 1995,22:15－24.

4 Wang L. Shop scheduling with genetic algorithm, TsingHua University Press, 2003. 59－67.

5 Bames JW, Chambers. Solving the job-shop scheduling problem using tabu search. IEEE Transactions. 1995, 27:257－63.

6 van Laarhoven PJM, Aarts EHL, Lenstra JK. Job shop scheduling by simulated annealing. Operations Research, 1992,40:113－25.

7 Colorni A, Dorigo M, Maniezzo V, Trubian M. Ant system for Job-Shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science, 1993,34:39－54.

8 Dorigo M, Gambardella LM. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans on Evolutionary Computation, 1997,1(1):53－66.

9 Dorigo M, Stitzle T. Ant Colony Optimization. MA: MIT Press, 2004.