

文章编号:1007-6735(2011)04-0362-05

工件排序的改进蚁群算法优化

王欣盛, 马良

(上海理工大学 管理学院)

摘要: 就工件排序问题中的一种类型设计了融合局部改进策略的蚁群算法进行求解,并用 Delphi 在计算机上实现了相应的算法软件.经大量算例测试,获得了较好的效果,验证了算法的可行性和有效性.

关键词: 排序问题; 蚁群算法; 组合优化

中图分类号: O 22 **文献标志码:** A

Improved ant colony optimization for scheduling problem

WANG Xin-sheng, MA Liang

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

Abstract: Combined with the local searching strategy, an ant colony optimization algorithm was proposed for a kind of scheduling problems. The algorithm is coded in Delphi and runs on microcomputer. By solving series of experimental examples, the modified algorithm was tested and validated with satisfactory results achieved. The computational results prove the feasibility and effectiveness of the algorithm for solving scheduling problems.

Key words: scheduling problem; ant colony algorithm; combinatorial optimization

工件排序问题^[1]是组合优化中的著名难题,由于其一般情形下的 NP(nondeterministic polynomial)难解性,人们一直在不断地对其进行研究.而排序问题在现实中的应用,如航班着陆排序、汽车组装线排序等,更是对人们的日常生活和工作有着重大的影响.

自 20 世纪 90 年代以后,来自自然界的仿生类算法开始引起人们的关注,尤其是在解决组合优化难题方面,得到了一系列的尝试和深入研究.其中,源自生物世界的蚁群算法在近十几年里获得了长足的发展^[2-9].

本文主要针对一类工件排序问题设计了一种改进的蚁群算法,并在计算机上实现了相应的软件程序,经大量算例测试,获得了较好的效果.

1 问题概述

考虑单工序、多品种、多机器工件加工排序问题,即有多个品种的工件,其中,每个工件只有一道加工工序,众多工件同时在多台加工机器上并行加工,具体描述为:

收稿日期: 2010-06-09

作者简介: 王欣盛(1988-),男,本科.研究方向: 系统科学. E-mail: wangxsh42@126.com

马良(联系人),男,教授.研究方向: 智能优化. E-mail: maliang@usst.edu.cn

有 x 件工件, y 台不同种类的机器, 每个工件可以在任何一台机器上加工, 第 i 件工件在第 j 台机器上的加工时间为 t_{ij} , $U = \{U_1, U_2, \dots, U_y\}$, U 是在第 i 台机器上所有加工工件的集合, 优化的目标是使各台机器上的最大加工时间值最小.

2 基于蚁群算法的改进

2.1 算法改进思想

现介绍有关概念的定义.

节点: 每个节点代表两个实体——加工机器和工件, 有 $M \times N$ 个节点, 其含义为第 i 个工件在第 j 个机器上加工.

边与边长: 蚁群算法在处理此类型工件排序问题上, 可忽略边的存在, 转而关注从节点到节点转移所花费的时间. 若要从某个节点转移到另一节点, 需要花费的时间可以在目标节点中获得, 即从节点 i 转移到节点 j 所需要的时间为节点 j (节点 j 是由第 x 个工件与第 y 台机器组合而成) 第 x 个工件在第 y 台机器的加工时间.

蚂蚁: 在众多的节点中转移, 通过其转移到的节点, 来确定工件排序的可行解, 并且通过对可行解的比较, 得出问题的最优解.

2.2 算法改进关键点

对基本蚁群算法作如下改进:

a. 将待加工工件与加工机器的组合作为蚁群算法中每个蚂蚁所要转移的节点.

b. 将工件加工时间作为蚂蚁在节点转移中所花费的时间, 该时间与所要转移的节点有关, 即节点为第 i 个工件在第 j 台机器加工, 则转移所需花费的时间为第 i 个工件在第 j 台机器加工的时间.

c. 转移概率结合考虑转移花费时间与其他蚂蚁留下的信息素浓度.

d. 解是由 M (机器台数) 个子解集构成, 每个解集代表在该台机器加工的工件号.

e. 每次循环后 (即同一时刻出发的蚂蚁全部完成觅食后), 对信息素进行更新.

2.3 改进蚁群算法描述

首先随机放置 m 个蚂蚁到 n 个节点上, 然后对每个蚂蚁进行节点转移. 在蚂蚁要选择一个节点作为移动目标前, 按转移概率选择可能转移的节点. 蚂蚁转移的可能性为轨迹强度的增函数、花费时间的减函数, 即转移花费时间越少, 可见度越大, 越容易被选择; 而走过的蚂蚁越多, 在边上留

下的轨迹信息素越多, 轨迹强度就越强, 也越容易被选择. 经多次循环后, 所有的蚂蚁所走的路线趋向于最佳路径.

由于这里研究的是单工序、多工件、多机器的加工工件排序问题, 其解的形式不同于双机 N 件工件排序问题的解形式, 最终所得的解是由 M 个 (加工机器数) 子解集的集合, 每个子解集的值代表在该加工机器上需要加工的工件号.

3 改进蚁群算法规则及步骤

在原始蚁群算法规则和步骤基础上, 即可实现具体的改进蚁群算法

3.1 相关参数

α 为信息素相对重要性, $\alpha \geq 0$; β 为转移时间相对重要性, $\beta \geq 0$; ρ 为轨迹衰减度, $0 \leq \rho \leq 1$; τ 为信息素, $\tau \geq 0$; m 为蚂蚁数, $m \geq 0$; U_i 为第 i 台机器加工的工件集合; S_i 为集合的 U_i 未来总加工时间, $S_i \geq 0$; Q 为蚂蚁一次所留信息素的量为一个常数; d 为节点转移所需时间; η 为节点能见度, 等于 $1/d$; θ 为某机器未来加工总时间的相对重要度, $\theta \geq 0$; P 为转移概率; p_{se}^k 为第 k 只蚂蚁从节点 s 转移到节点 e 的转移概率; s, e 分别为所在节点, 转移目标节点; $\Delta\tau_{ij}$ 为节点 i 到 j 的信息素增量; d_{ij} 为第 i 个工件在第 j 个机器加工时间.

3.2 目标函数

$$\min \max \left\{ \sum_{i \in U_1} d_{i1}, \dots, \sum_{i \in U_i} d_{ij}, \dots, \sum_{i \in U_m} d_{im} \right\}$$

其中, $\sum_{i \in U_j} d_{ij}$ 为 j 机器的总加工时间, 目标函数使得最大总加工时间最少.

3.3 规则

3.3.1 状态转移规则

$$p_{se}^k = \begin{cases} \frac{\tau_{se} \eta_{se}^\beta}{\sum_{e \notin T} (\tau_{se} \eta_{se}^\beta S_i^\theta)} & e \notin T \\ 0 & e \in T \end{cases}$$

当转移目标节点 e 不在禁忌表 T 中, 计算转移概率; 如果该点在 T 中, 则不考虑转移到该节点. 当 $p_{se}^k = 0$ 时, 直接将其视为最佳转移概率. 如果有相同转移概率的节点存在, 则从中随机选取一个作为转移节点. 其中, i 与 e 节点的加工机器号相等.

3.3.2 轨迹强度更新规则

在蚁群算法中, 第 k 只蚂蚁从节点 s 到节点 e 的转移路径所留下的信息素, 可由有该蚂蚁觅食的

时间来决定,故该蚂蚁在从 s 节点转移到 e 节点所留下的信息素为

$$\Delta\tau_{se}^k = \begin{cases} Q/\sum_{i=1}^m S_i & \text{蚂蚁从 } s \text{ 节点转移到 } e \text{ 节点} \\ 0 & \text{其他} \end{cases}$$

式中, $\sum_{i=1}^m S_i$ 为蚂蚁节点转移花费的总时间.

同一时刻出发的蚂蚁全部完成觅食(找到可行解)后,轨迹强度更新规则为

$$\Delta\tau_{se} = \sum_{k=1}^m \Delta\tau_{se}^k$$
$$\Delta\tau_{se} = (1 - \rho)\tau_{se} + \Delta\tau_{se}$$

3.4 算法步骤

步骤 1 初始化.初始化各个节点的信息素,赋值为 0; T 表(禁忌表)初始化,置空;随机将 m 个蚂蚁放置于 n 个节点上;将路径信息素赋初值,这里使用 $Q/(\min \text{ 加工时间} \times \text{零件数})$.

步骤 2 将 T 表与 U 相对应,蚂蚁的初始出发点置于 T 表中,同时给相对应的 U_i 集合添加该工件号.

步骤 3 对每个蚂蚁计算转移概率 p_{rs}^k ,按最大转移概率将该蚂蚁从现节点转移至 s 节点,并将 s 节点放 T 表中,在相对应的 U_i 集合添加该工件号.如所对应的 U_i 为空,则将该工件号直接放入该子解集中;如不为空,则计算每个 U_i 的所含有的元素的加工时间总和.选取加工时间总和最小的 U_i 集合,放入所选节点的工件号,若有多个 U_i 集合加工时间总和最小,则随机选取一个集合.

步骤 4 在蚂蚁获得可行解后,计算每个蚂蚁的目标函数值,在同一时刻出发的所有蚂蚁得到可行解后,作邻域搜索局部改进,并更新轨迹强度.

步骤 5 如循环次数达到或超过给定的最大循环次数,结束排序,输出当前最佳结果;否则,转步骤 2.

4 改进蚁群算法的 Delphi 实现

整个算法用 Delphi 10 实现,并在 Windows XP 环境下运行通过.相应的算法软件界面如图 1 和图 2 所示.

算法软件可通过导入数据文件、手工输入或通过测试数据生成器来形成原始数据,并在软件的输入框中显示这些数据.在软件求解完毕后,会在右输出框中显示相应的结果.



图 1 测试用例生成器
Fig.1 Test data generator

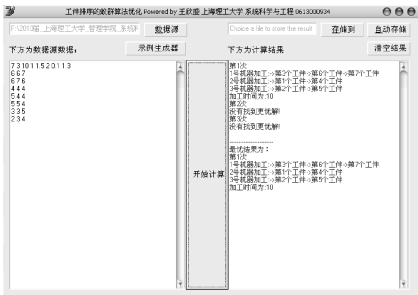


图 2 软件计算示意图
Fig.2 Software computation

5 算例测试

例 1 数据如表 1 所示.

表 1 算例数据 1
Tab.1 Sample data 1

工件	加工机器号		
	1	2	3
1	6	6	7
2	6	7	6
3	4	4	4
4	5	4	4
5	5	5	4
6	3	3	5
7	2	3	4

在工件数为 7,加工机器数为 3,蚂蚁数量为 10,能见度重要性为 1.5,已经过路径长度重要性为 2,轨迹衰减度为 0.1,循环次数为 50 的情况下,所得计算结果如表 2 所示.

优化结果为:

1 号机器加工:→第 5 个工件→第 6 个工件→第 7 个工件(实际加工时间为 10);

2 号机器加工:→第 1 个工件→第 4 个工件(实际加工时间为 10);

3 号机器加工:→第 2 个工件→第 3 个工件(实际加工时间为 10);
机器最大加工时间为 10.

例 2 数据略.
在工件数为 7,加工机器数为 3,蚂蚁数量为 20,能见度重要性为 1.5,已经过路径长度重要性为 2,轨迹衰减度为 0.1,循环次数为 80 的情况下,所得计算结果如表 3 所示.

表 2 计算结果 1
Tab.2 Computational results 1

工件	加工机器号			加工机器
	1	2	3	
1	6	6	7	2
2	6	7	6	3
3	4	4	4	3
4	5	4	4	2
5	5	5	4	1
6	3	3	5	1
7	2	3	4	1

注: 表中带方框数据为软件求得的解.

表 3 计算结果 2
Tab.3 Computational results 2

工件	加工机器号									
	1	2	3	4	5	6	7	8	9	10
1	8	6.3	7.5	5.9	7	5.6	6.5	5.5	6.5	6
2	5.5	3.8	5.4	3.8	5.3	3.5	4.5	3.5	4	3.9
3	5.1	4.2	4.8	4	4.8	4	4.5	4	4.5	4.3
4	9	6	8	6	7.3	5.7	7	5.6	6.4	6.1
5	11	8.5	10.4	8.2	9.1	8	9	8	8.8	8.5
6	13.9	13.3	13.8	13.2	13.7	13	13.6	13	13.5	13.4
7	4.5	3.8	4.4	3.7	4.4	3.5	4.3	3.5	4.3	4.3
8	14.2	11.5	13.5	11.3	13	11	13	11	12.5	12.2
9	12.2	9.7	12	9.5	11.5	9.3	11	9.2	10.5	9.8
10	2.5	1.8	2.4	1.8	2.3	1.7	2.2	1.7	2	2
11	4.5	3.2	4.2	3.1	4	3	3.8	2.8	3.5	3.4
12	9	7	8.5	6.8	8	6.6	7.8	6.5	7.5	7.3
13	15	13.5	14.5	13.2	14.3	13	14	13	14	13.8
14	17.8	13.5	17	13	16	12.5	15	12	14.5	14
15	22	18.5	21.5	18	21	18	20	18	19.5	19
16	17	13.5	16.5	13	16	12.5	15.1	12	14.5	14
17	24.4	20.5	23	20	22.2	19.5	22	19	21.5	21
18	8.5	7	8.2	6.9	8	6.8	7.8	6.8	7.5	7.2
19	19	15.5	18.5	15	18	14.5	17.5	14	16.5	16
20	30	26.5	29.3	26	28.5	26	28	25	27.5	27
21	18.5	15.5	17.7	15.2	17.1	15.1	16.5	15	16.2	16
22	12	10	11.5	9.8	11.2	9.5	11	9.5	10.5	10.2
23	29.5	26.8	29	26.5	28.5	26	28	25.5	27.5	27
24	17.2	16	17	15.8	16.8	15.5	16.5	15.2	16.2	16.1
25	7.5	6	7.2	5.8	7	5.5	6.8	5.2	6.5	6.2
26	16.2	14	16	13.5	15.5	13.4	15.2	13.2	15.1	14.5
27	28.3	25	28	24.5	27.6	24	27	24	26	25.5
28	8	6.8	7.8	6.5	7.7	6.3	7.4	6.1	7.2	7.1
29	12.5	10.5	12.2	10.2	12	10.1	11.5	9.9	11	10.9
30	24.8	22	24.5	21.5	24	21	23.5	20.5	23	22.5
31	6.1	5.6	6	5.5	5.9	5.3	5.8	5.2	5.7	5.7
32	2.8	2.2	2.7	2.1	2.6	2	2.5	2	2.4	2.3
33	6.3	5.2	6.1	5	6	4.8	5.8	4.7	5.7	5.5
34	8.2	7.1	7.9	7	7.8	6.8	7.6	6.5	7.5	7.3
35	8.8	7.8	8.5	7.5	8.4	7.4	8.2	7.2	8.1	8
36	6	5	5.8	4.8	5.7	4.7	5.5	4.5	5.3	5.1
37	5.2	4.4	5	4.3	4.9	4.1	4.8	4	4.7	4.5
38	6.4	5.3	6.2	5.1	6	4.9	5.8	5.7	5.6	5.5
39	2.5	2	2.4	1.9	2.3	1.8	2.2	1.8	2.1	2.1
40	3.5	3	3.4	3	3.3	3	3.2	3	3.1	3.1
Total	42	42.5	41.1	42.5	42.5	42.5	41.2	42.1	36.8	41.8

注: 表中带方框数据为软件求得的解.

机器最大加工时间为 42.5.
例 3 数据如表 4 所示.

表 4 算例数据 3
Tab.4 Sample data 3

工件	加工机器号		
	3	1	2
1	1	6	7
2	1	7	6
3	1	4	4
4	1	4	4
5	1	5	4
6	1	3	5
7	1	3	4

在工件数为 7,加工机器数为 3,蚂蚁数量为 10,能见度重要性为 1.5,已经过路径长度重要性为 2,轨迹衰减度为 0.1,循环次数为 50 的情况下,所得计算结果如表 5 所示.

表 5 计算结果 3
Tab.5 Computational results 3

工件	加工机器号			加工机器
	3	1	2	
1	1	6	7	1
2	1	7	6	1
3	1	4	4	3
4	1	4	4	2
5	1	5	4	1
6	1	3	5	1
7	1	3	4	1

注:表中带方框数据为软件所求得解.

最终结果为:

1 号机器加工:→第 1 个工件→第 2 个工件→第 5 个工件→第 6 个工件→第 7 个工件(实际加工时间为 5);

2 号机器加工:→第 4 个工件(实际加工时间为 4);
3 号机器加工:→第 3 个工件(实际加工时间为 4);
机器最大加工时间为 5.

6 结束语

蚁群算法是一种仿生智能优化算法,在求解许多 NP-难题中表现出了良好的性能.这种随机寻优方法具有很强的发展潜力,同时,还常常能与其他优化算法相结合.本文仅对单工序、多工件、多加工机器的工件排序问题设计了相应的改进蚁群算法,并在计算机上予以实现,为实际问题的求解提供了方便的软件工具.

参考文献:

[1] 陈义保,姚建初,钟毅芳,等.基于蚁群系统的工件排序问题的一种新算法[J].系统工程学报,2002,10(5):476-480.
[2] 马良,王波.基础运筹学教程[M].北京:高等教育出版社,2006.
[3] 马良,朱刚,宁爱兵.蚁群优化算法[M].北京:科学出版社,2008.
[4] 雷德明,严新平.多目标智能优化算法及其应用[M].北京:科学出版社,2009.
[5] 王洪刚,马良.TSP 的量子蚂蚁算法求解[J].运筹与管理,2009,18(6):11-13.
[6] 朱刚,马良,高岩.离散元胞蚂蚁算法及其收敛性[J].科学技术与工程,2009,9(5):1115-1119.
[7] 朱刚,马良.生长竞争型函数优化的蚁群算法[J].数学的实践与认识,2010,40(2):108-112.
[8] 朱刚,马良.多目标优化的生长竞争蚁群算法[J].系统工程,2010,28(12):91-95.
[9] 刘勇,马良.非线性 0-1 规划的元胞蚁群算法[J].系统管理学报,2010,19(3):351-355.