

基于遗传算法的车间作业调度

边庆彪, 李浙昆, 樊瑜瑾, 许平, 朱立金
(昆明理工大学机电工程学院, 云南昆明 650093)

摘要: 研究了遗传算法在车间作业调度问题中的应用, 以遗传算法作为优化工具, 给出了车间作业调度算法. 最后给出了计算机仿真结果, 并与启发式算法求得的结果作了比较, 证明了该算法解的有效性和优越性.

关键词: 遗传算法; 作业调度; 启发式算法

中图分类号: TP391.73 **文献标识码:** A **文章编号:** 1007-855X(2003)01-0042-04

A Job Shop Scheduling Method Based on Genetic Algorithms

BIAN Qing-biao, LI Zhe-kun, FAN Yu-jin, XU Ping, ZHU Li-jin

(Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650093, China)

Abstract: This paper focuses on the application of genetic algorithms in job shop scheduling system. It presents the computing method of job shop scheduling based on the best way of genetic algorithms, whose results are compared with those from heuristic algorithms. The findings prove the effectiveness of the genetic algorithms.

Key words: genetic algorithms; job shop scheduling; heuristic algorithms

0 引言

车间作业调度(Job Shop Scheduling, 简称JSS)的启发式算法^[2], 是用某一调度优先级规则在当前可调度工序(其紧前工序已完成且加工使用机床空闲)中选择一个工序进行加工, 最终形成一个由所有被加工零件各工序组成的序列, 即所谓调度结果. 由于调度规则通常只针对特定问题和特定环境, 它存在着难以克服的缺点, 如计算规模不可能较大, 寻优具有局部性等. 而基于遗传算法的车间作业调度的基本思想是, 预先排列出若干个由工序组成的序列, 然后对这些序列进行遗传进化操作, 从而达到优化调度结果性能指标的目的. 由于遗传算法只利用适应性信息, 它不要求目标函数可微、连通和凸性, 因而它是一种高效率的随机搜索与优化的方法, 具有搜索面广、算法速度快等优点.

1 基于遗传算法的车间作业调度的实现技术

1.1 直接工序的编码方法

由遗传算法的运行机理可知, 不论JSS染色体编码怎样设计, 它必须能够直接或间接地表达JSS的一个可行解. 如果JSS染色体编码直接代表JSS的可行解, 称为直接编码; 如果JSS染色体编码需要进行一定的转换才能表示JSS的可行解, 叫做间接编码. 对于JSS问题, 其可行解为所有被加工零件各工序组成的序列. 我们采用直接编码的方式, 这样消除了遗传算法应用于JSS受到成组技术的局限性.

1.2 目标函数的建立

我们把优化调度结果转化如下目标函数的优化

$$\min(T(JM)) = \min\{\max[C(1), C(2), \dots, C(i), \dots, C(m)]\}$$

其中 $C(i)$ 为全部 m 台机器中第 i 台机器的总加工时间, $T(JM)$ 为总的完工时间.

收稿日期: 2002-07-02; 基金项目: 云南省计算机技术应用重点实验室资助项目.

第一作者简介: 边庆彪(1977.12~), 男, 硕士; 主要研究方向: 车间作业调度, workflow 管理系统.

1.3 基本染色体的产生

JSS 染色体编码是一个有序的编码串,代表着工件的第几个工序在哪台机器上加工及加工时间,并且转换后 JSS 编码与 JSS 有效解应该一一对应,保证 JSS 解的有效性.基于染色体编码的要求,我们采用单一编码方式.将所有工件的总的工序统一编号.例如,对于 n 个工件 m 台机器,第 i 个工件的工序数为 $k_i, k = \max\{k_1, k_2, \dots, k_n\}$,这样的调度问题可以用如下方式表示:

$$J[nk] = \{(1, 1, M_1, t_1), (1, 2, M_2, t_2), \dots, (n, k, M_m, t_{nk})\}$$

其中 (n, k, M_m, t_{nk}) 表示第 n 个工件的第 k 个工序在 M_m 机床上加工,加工时间为 t_{nk} .其中小括号内的元素作为染色体的一个基因,它是一个整体,遗传运算过程中不可分割,避免了在其后的遗传进化操作中产生非法解的染色体.

1.4 选择初始种群

随机产生与群体规模一致的个数为 nk 的染色体串作为初始种群,其中每个染色体串用以上方式表示,保持各基因之间的相互独立.

1.5 适应度函数

适应度函数在子代的产生中起着重要的作用.由于是求问题的最小值,为了使遗传算法对适应度较高的个体有更多的生存机会,本文通过对目标函数 $T(JM)$ 进行变换得到适应度函数 $F(JM)$,即取 $F(JM) = \alpha + \beta T(JM)$, α, β 为常数,目标函数为: $T(JM) = \max[C(1), C(2), \dots, C(i), \dots, C(m)]$.

1.6 选择策略

每代中的每一个个体,按照适应度的大小决定它能够复制到下一代的概率.通过复制,使得群体中的优秀个体数目不断增加,整个优化过程朝着更优解的方向进行,反映了优胜劣汰的原则.各个体的选择概率和其适应度成正比例.令 $P(k)$ 为个体 k 的选择概率,则

$$P(k) = F(k) / \sum_{l=1}^{nk} F(l) \quad (k = 1, 2, \dots, nk)$$

令 $S(0) = 0, S(k) = P(1) + P(2) + \dots + P(k), k = 1, 2, \dots, nk$,其中 pop_size 为群体规模,共产生 nk 个 $0 \sim 1$ 均匀分布的随机数,若 $S(k-1) < \xi_s < S(k)$,则选个体 k 为下一代的父代,其中 ξ_s 为 $0 \sim 1$ 的随机数, $s = 1, 2, \dots, nk$.

1.7 交叉

由于本文所用的编码方式使基因之间完全相互独立,所以交叉方式采用 k 点杂乱交叉方式.此交叉方法是先用随机的方法选取 k 个交换点,在这 k 个交换点处进行交叉,从而将原有字符顺序打乱,以使增加各字符相互交换的概率,目的是使基因之间更充分交换信息,以便更快地达到最优解.

例1 设有父代个体 P^0 和 P^1 ,将各字符按顺序编号得:

$$P^0 = P_0^0 \ P_1^0 \ P_2^0 \ P_3^0 \ P_4^0 \ P_5^0 \ P_6^0 \ P_7^0 \ P_8^0 \ P_9^0$$

$$P^1 = P_0^1 \ P_1^1 \ P_2^1 \ P_3^1 \ P_4^1 \ P_5^1 \ P_6^1 \ P_7^1 \ P_8^1 \ P_9^1$$

用随机的方法打乱 P^0 和 P^1 的字符排列顺序,使第 i 个字符位于第 j 位,但两个父代的个体的 i, j 要相同,如

$$P^0 = P_2^0 \ P_3^0 \ P_6^0 \ P_9^0 \ P_0^0 \ P_7^0 \ P_8^0 \ P_4^0 \ P_3^0 \ P_1^0$$

$$P^1 = P_2^1 \ P_3^1 \ P_6^1 \ P_9^1 \ P_0^1 \ P_7^1 \ P_8^1 \ P_4^1 \ P_3^1 \ P_1^1$$

用随机的方法产生 k 个交叉点,并进行 k 点交叉(见表1),如:

表1 随机法交叉表

序号	交换号	1	2	3
1	父代	$P^0 = P_2^0 \ P_3^0 \ P_6^0 \ P_9^0 \ P_0^0 \ P_7^0 \ P_8^0 \ P_4^0 \ P_3^0 \ P_1^0$	$P^1 = P_2^1 \ P_3^1 \ P_6^1 \ P_9^1 \ P_0^1 \ P_7^1 \ P_8^1 \ P_4^1 \ P_3^1 \ P_1^1$	$P^0 = P_2^0 \ P_3^0 \ P_6^0 \ P_9^0 \ P_0^0 \ P_7^0 \ P_8^0 \ P_4^0 \ P_3^0 \ P_1^0$
2		$P^1 = P_2^1 \ P_3^1 \ P_6^1 \ P_9^1 \ P_0^1 \ P_7^1 \ P_8^1 \ P_4^1 \ P_3^1 \ P_1^1$	$P^0 = P_2^0 \ P_3^0 \ P_6^0 \ P_9^0 \ P_0^0 \ P_7^0 \ P_8^0 \ P_4^0 \ P_3^0 \ P_1^0$	
3	子代	$P^1 = P_2^1 \ P_3^1 \ P_6^1 \ P_9^1 \ P_0^1 \ P_7^1 \ P_8^1 \ P_4^1 \ P_3^1 \ P_1^1$	$P^0 = P_2^0 \ P_3^0 \ P_6^0 \ P_9^0 \ P_0^0 \ P_7^0 \ P_8^0 \ P_4^0 \ P_3^0 \ P_1^0$	$P^1 = P_2^1 \ P_3^1 \ P_6^1 \ P_9^1 \ P_0^1 \ P_7^1 \ P_8^1 \ P_4^1 \ P_3^1 \ P_1^1$
4		$P^0 = P_2^0 \ P_3^0 \ P_6^0 \ P_9^0 \ P_0^0 \ P_7^0 \ P_8^0 \ P_4^0 \ P_3^0 \ P_1^0$	$P^1 = P_2^1 \ P_3^1 \ P_6^1 \ P_9^1 \ P_0^1 \ P_7^1 \ P_8^1 \ P_4^1 \ P_3^1 \ P_1^1$	

1.8 变异

复制和交叉只能在现有的基因型的排列和组合内寻找最优,而不能产生新的基因型.变异算子对每个个体的每一位按变异概率产生新的变化,产生新的基因型,扩大了寻优范围,从而有利于保证算法的全局最优性.

2 车间作业调度的整体结构流程(如图1)

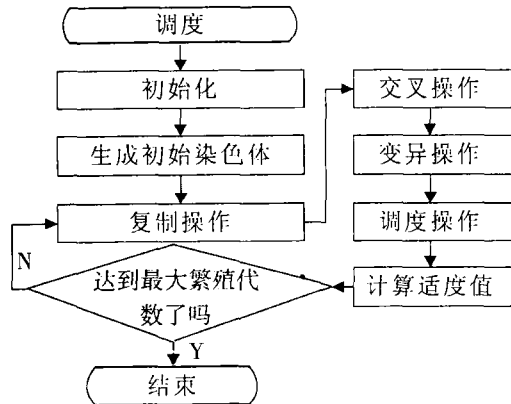


图1 车间作业调度的整体结构流程图

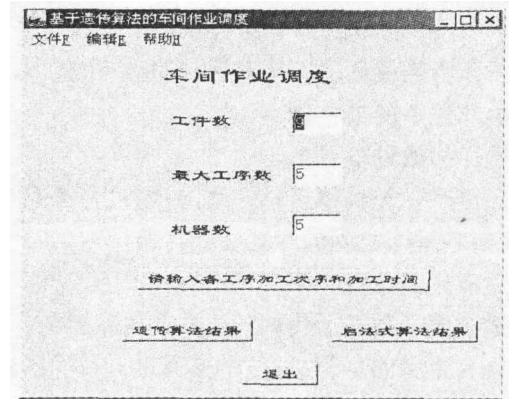


图2 车间作业调度主界面

3 仿真实验及结果

为了验证采用遗传算法解决车间作业调度问题的效果,笔者用 C++ Builder 5.5 开发了“基于遗传算法的车间作业调度”软件,对不同规模的车间作业调度问题进行了仿真,并将其结果与启发式算法所得的结果进行了比较.算例包括 30 个工件 /10 台机床 /20 道工序 (30/10/10) 以及 30/10/15,50/20/10,15/10/10,20/5/10,10/10/10 等许多情况,都取得了较好的结果.限于篇幅只将其中一个算例的仿真列于其后,该调度问题的规模为:工件数 $n = 9$,机床数 $m = 5$,最大工序 $k = 5$,交叉概率 $P_c = 0.5$,变异概率 $P_m = 0.1$,终止代数 $T = 1000$,由于篇幅的限制,本文只给出车间作业调度的主界面(见图2).

初始种群如下: $J[5 \times 9] = \{ \{1,1,1,21\}, \{1,2,0,53\}, \{1,3,4,95\}, \{1,4,3,55\}, \{1,5,2,34\}, \{2,1,0,21\}, \{2,2,3,52\}, \{2,3,4,16\}, \{2,4,2,26\}, \{2,5,1,71\}, \{3,1,3,39\}, \{3,2,4,98\}, \{3,3,1,42\}, \{3,4,2,31\}, \{3,4,0,12\}, \{4,1,1,77\}, \{4,2,0,55\}, \{4,3,4,99\}, \{4,4,2,66\}, \{4,5,3,77\}, \{5,1,0,83\}, \{5,2,3,34\}, \{5,3,2,64\}, \{5,4,1,19\}, \{5,5,4,37\}, \{6,1,1,54\}, \{6,2,2,43\}, \{6,3,4,79\}, \{6,4,0,92\}, \{6,5,3,62\}, \{7,1,3,69\}, \{7,2,4,77\}, \{7,3,1,87\}, \{7,4,2,87\}, \{7,5,0,93\}, \{8,1,2,38\}, \{8,2,0,60\}, \{8,3,1,41\}, \{8,4,3,24\}, \{8,5,4,84\}, \{9,1,3,17\}, \{9,2,1,49\}, \{9,3,4,25\}, \{9,4,0,44\}, \{9,5,2,98\} \}$

机器加工过程仿真结果如下(见表2):($m - w - n - t$ 所表示的意义为 m - 机器号; w - 工件号; n - 工序号; t - 加工完该工序的时间,单位为 s)

表2 加工过程仿真结果

0-2-1-21	0-5-1-104	0-4-2-186	0-1-2-239	0-8-2-299	0-9-4-343	0-6-4-559	0-3-5-571	0-7-5-664
1-6-1-54	1-4-1-131	1-1-1-152	1-9-2-201	1-7-3-301	1-8-3-342	1-3-3-384	1-5-4-403	1-2-5-599
2-8-1-38	2-6-2-97	2-5-3-275	2-4-4-359	2-3-4-415	2-7-4-502	2-2-4-528	2-9-5-626	2-1-5-660
3-3-1-39	3-9-1-56	3-7-1-125	3-2-2-177	3-5-2-211	3-1-4-443	3-8-4-467	3-4-5-544	3-6-5-621
4-3-2-137	4-7-2-214	4-4-3-293	4-1-3-388	4-6-3-467	4-2-3-483	4-9-3-508	4-8-5-591	4-5-5-628

(下转第52页)

3 结束语

利用各配电室、开闭所的出线开关每个季度的负荷记录;安装于公用变压器低压测的配变综合测试仪的实测数据及调查用户的建筑面积、用电设备及用电等资料,用数理统计的原理与方法对目前昆明城市电力网内高层住宅居民、多层住宅居民及电炊户居民这三类用电负荷进行了分类调查及负荷计算方法的研究,提出了适于昆明城网实际的科学、真实并具有弹性的居民用电指标,指出了城区居民用电负荷构成情况,为在规划设计阶段正确计算其负荷,以确定符合城市经济发展的经济合理的供电方案提供了科学依据。

参考文献:

- [1] 陈章潮等. 城市电网规划与改造[M]. 北京:中国电力出版社,1999.
- [2] 龚国芳等. 居民用电负荷的调查分析[J]. 供用电,1995(2):20.
- [3] 万载扬. 重庆高层建筑的供电问题[J]. 供用电,1998,39(4):112.
- [4] 潘承毅等. 数理统计的原理与方法[M]. 上海:同济大学出版社出版,1993.20~70.
- [5] 温伯银. 超高层建筑供用电[J]. 供用电,1996,(2):80~120.

(上接第44页)

比较各台机器的最后完工时间,该时间即为所求问题的最后完成时间.得到结果为664,按照此排序方法得到的完工时间为664s(见表3).

4 结论

仿真实验结果表明:采用遗传算法调度结果的评价指

标明显优于启发式算法的调度结果,而且采用遗传算法解决车间作业调度问题,不仅可以优化调度结果,而且计算简单易于实现,在运算速度上基本可以满足实际调度的要求.充分说明了遗传算法解决JSS问题的有效性和优越性.笔者用的是面向对象编程,工件数、机床数、最大工序和时间都是可以根据实际情况而定,人机界面友好,可直接应用于生产.“基于遗传算法的车间作业调度”软件的开发,必将对CIMS的研究与应用具有重大意义。

参考文献:

- [1] 纪树新,钱积新,孙优贤. 遗传算法在车间作业调度中的应用[J]. 系统工程理论与实践,1998,(5):34~39,81.
- [2] 姜迪刚,叶尚辉. 基于遗传算法的车间作业调度[J]. 西安电子科技大学学报(自然科学版),2001,28(2):207~210.
- [3] 邢文训,谢金星. 现代优化计算方法[M]. 北京:清华大学出版社,1999.
- [4] DavisL. Handbook of Genetic Algorithms[M]. Van Nonstranded Reinhold, New York, 1991.

表3 9个工件5台机床的调度结果(加工时间)

样本编号	遗传算法	启发式算法(FCFS)
1	664s	724s