

含有运输小车的柔性作业车间调度研究

周 鑫

(湖北工业大学机械工程学院, 湖北 武汉 430068)

✉1205316902@qq.com



摘 要: 在实际的柔性作业车间调度中, 不但工件需要加工时间, 而且工件在各个机器之间利用AGV(自动导引小车)转移也需要占用一定的时间, 因此对柔性作业车间调度中考虑AGV运输时间的研究更具有实际意义。针对此问题, 本文建立含有AGV的柔性作业车间调度的数学模型, 针对问题自身特点对遗传算法进行改进, 引入局部搜索策略加强局部寻优能力, 将模拟退火算法作为局部搜索策略加入全局搜索中, 增强了算法的收敛性能。通过在仿真实验平台上的实验数据结果可以看出, 本算法有比较好的效果。

关键词: 遗传算法; 柔性作业车间调度; 模拟退火; 自动引导小车

中图分类号: TP301 **文献标识码:** A

Research on Flexible Job-shop Scheduling with Automated Guided Vehicle

ZHOU Xin

(School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China)

✉1205316902@qq.com

Abstract: In the actual flexible job-shop scheduling, it is more of practical significance to consider transportation time of Automatic Guided Vehicle (AGV) in flexible job-shop scheduling Research, because workpieces not only require processing time, but also take a certain amount of time to be transferred between machines using AGV. Aiming at this problem, this paper proposes to establish a mathematical model of flexible job-shop scheduling with AGV and improve genetic algorithm according to characteristics of the problem. Local search strategy is introduced to strengthen the local optimization ability, and simulated annealing algorithm is added as a local search strategy to global search in order to enhance algorithm convergence performance. Experimental data from simulation experiment platform show that this algorithm is comparatively effective.

Keywords: genetic algorithm; flexible job-shop scheduling; simulated annealing; automatic guided vehicle

1 引言(Introduction)

FJSP是公认的NP难问题, 所以FJSP一直是国内外学者研究的热点^[1]。这个问题从20世纪90年代Bruker等人提出后, 国内外的专家学者进行了深入广泛的研究。FJSP是JSP的扩展, JSP工件的工序所在的加工机器是固定的, FJSP更加符合实际车间中的生产环境。目前求解FJSP的主要智能算法有遗传算法^[2]、免疫算法^[3]、粒子群算法^[4]、蚁群算法^[5]等。张国辉等^[6]使用随机初始化种群优化方法。赵诗奎^[7]等基于设备均衡的策略并运用均匀设计的思想优化了种群的初始化方法。张立果^[8]等针对多目标的柔性作业车间调度问题提出新的交叉策略来进行求解。在实际的生产环境中, 工件在不同的加工机

器之间通过AGV小车进行搬运。

对于AGV参与集成车间调度问题的研究也得到一些学者的广泛研究, 付建林等^[9]对AGV和车间调度之间的融合调度方法——AGV调度优化问题做了一个分类, 为研究AGV融合车间调度的学者提供了学习的建议。戴敏等^[10]针对加工机器和AGV的集成调度提出了改进的分布估计算法进行优化研究。刘二辉等^[11]通过改进花授粉算法对AGV和机器的集成调度进行了研究。徐云琴等^[12]研究了AGV在柔性作业车间调度中的数量变化对加工时间的影响。贺长征等^[13]对AGV在柔性作业车间中的路径进行了优化。

通过阅读以上文献发现, 目前AGV在柔性作业车间调度

中的集成调度相关研究较少, 大部分的文献都没有研究车间调度和AGV相融合的问题。针对此问题, 本文将考虑AGV在实际生产环境中发挥调度作用这一因素, 研究柔性作业车间AGV的融合问题, 让本文的研究更加符合实际的生产环境。本文内容和以上文献的主要区别为: 本文在遗传算法中加入局部搜索增强局部寻优能力, 并研究了有AGV情况下的车间调度问题。

2 含有AGV的柔性作业车间调度模型(Flexible job-shop scheduling model with AGV)

2.1 问题描述

含有AGV的柔性车间调度问题可以描述为: n 个工件由多台AGV运送到多台不同的机器上加工, 工件的不同工序在不同的机器上进行加工的加工时间不同, 每台加工机器之间的距离也不相同。每个工件有 n_i 道不同的工序、 r 个AGV, 且 $r < m+2$ 。有如下假设:

未加工的工件和已加工的工件分别放置在未加工的区域 P_1 和已加工的区域 P_2 。

(1)AGV的搬运速度是不变的, 搬运时间和机器之间的距离有关, 设相邻的机器之间距离相等。

(2)一台机器一次处理一个零件。

(3)不计算工件在设备缓冲区中的时间。

(4)在最初的时刻, 所有的机器和工件都可以使用。

(5)AGV的运输路线是固定的, AGV的运输没有延迟, 并且不同AGV的运输不会相互干扰。

2.2 模型建立

根据以上描述的资源约束条件建立模型, 在车间调度中有多目标优化和单目标优化, 本文采用比较常见的单目标优化方案, 以所有工件加工完成的最小时间为目标。

$$\text{Makespan} = \min(\max(T_{me})) (1 \leq m \leq M) \quad (1)$$

$$T_{agve} - T_{agvs} = T_{mn} \quad (2)$$

$$T_{ijs} = \begin{cases} T_{agve} & T_{agve} > T_{me} \\ T_{me} & T_{agve} < T_{me} \end{cases} \quad (3)$$

$$T_{agvs} = T_{me} \quad (4)$$

$$T_{me} = T_{ms} + t_{ij} \quad (5)$$

$$\sum_1^m X_{ijm} = 0 \quad (6)$$

$$\sum_1^m x_{ijm} = 1 \quad (7)$$

$$T_{ije} = T_{ijs} + t_{ij} \quad (8)$$

$$N = \sum_{i=1}^h \sum_{j=1}^k O_{ij} \quad (9)$$

式中, i 为工件($i=1, 2, \dots, h$), h 为工件数; j 为工序($j=1, 2, \dots, k$), k 为工序数; m 为机器($m=1, 2, \dots, M$), M 为机器的数量; T_{ms} 和 T_{me} 表示加工开始和结束的时间; T_{agvs} 和 T_{agve} 为小车运输工件的过程所消耗的时间; T_{mn} 表示工件从机器 m 到机器 n 之间所花费的时间; T_{ijs} 和 T_{ije} 是第 i 个工件的第 j 道工序的加工开始和结束时间; t_{ij} 是第 i 个工件的第 j 道工序的加工

时间; $\sum_1^m X_{ijm} = 0$ 和 $\sum_1^m x_{ijm} = 1$ 分别表示工序 O_{ij} 是否在机器上加工, 0表示不加工, 1表示加工。

公式(1)为最小完工时间; 公式(2)表示AGV的运输时间由两台设备之间的距离决定; 公式(3)表示AGV运输工件是独立运行的; 公式(4)表示小车能使工件的每一道工序完成后瞬间被小车运往下一个机器; 公式(5)表示一道工序开始后不能停止; 公式(6)和公式(7)表示一个工件的一道工序在某一时刻只能被一台加工机器加工; 公式(8)表示相同工件的工序是有顺序进行加工的; 公式(9)表示将工序累加。

3 改进遗传算法设计(Improved genetic algorithm design)

3.1 基本原理

本文算法的基本原理是在原有GA的基础上加以改进, 延用GA的基本框架, 并用模拟退火作为局部搜索让算法的收敛性能更好, 并且在每次全局搜索过程中的交叉变异之后都要进行局部搜索, 从而能够达到获得较好质量的最优解的目的。改进GA的程序结构流程图, 如图1所示。

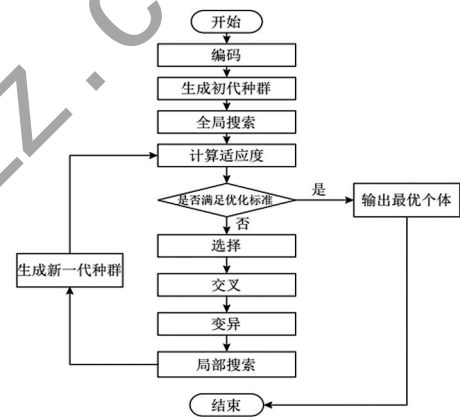


图1 改进GA程序结构流程

Fig.1 Improve GA program structure flow chart

3.2 全局搜索方法

本文研究的算法是基于传统的遗传算法的基本原理, 并结合车间调度的实际生产环境加以改进。在实际的车间调度中要考虑工件在不同机器之间的运输过程等因素, 因此本文在结合GA的基础上将AGV和机器进行融合调度研究。在全局搜索方法中加入局部搜索, 延用GA的基本框架, 并用模拟退火作为局部搜索让算法的收敛性能更好。GA的基本框架原理是由随机产生初始解, 经过评估、选择、交叉、变异等操作最终获得最优解的过程。

3.3 编码与解码

传统的遗传算法中, 一旦种群规模太大, 无法有效淘汰无用的个体, 容易出现局部最优的情况, 既影响了种群的进化速度, 又影响了计算结果的准确性, 因此根据第2节含有AGV的柔性车间调度数学模型, 本文有两种编码方式: 一种编码方式是对机器进行编码, 另一种编码方式是对工序进行编码。

为了展示本文算法所提出的编码, 用三台加工机器和三

个加工工件为实验例子模拟实际加工环境。表1和表2表示三台加工机器上三个工件的不同工艺的加工程序，不同加工机器之间工件的运输时间不同，以及不同加工机器之间的距离不同。其中， O_{ij} 表示工件*i*的第*j*道工序。

表1 三个工件在三台机器上的加工工艺

Tab.1 The process of 3 workpieces on 3 machines

工件	工序	M_1	M_2	M_3
J_1	O_{11}	2	4	5
	O_{12}	3	6	2
	O_{13}	3	3	2
J_2	O_{21}	4	4	2
	O_{22}	5	3	2
	O_{23}	4	3	2
J_3	O_{31}	3	2	4
	O_{32}	7	3	5
	O_{33}	4	3	2

表2 AGV小车在机器间的转移时间

Tab.2 Transfer time of AGV between machines

机器	M_1	M_2	M_3
M_1	0	3	5
M_2	3	0	2
M_3	5	2	0

假设工序 O_{ij} 在机器 M_i 上加工，各工件的加工顺序是按照同一个工件有先后顺序的约束，以选取32231321为机器的编码、112233123为工序的编码为例，其对应的解码过程如图2(a)和图2(b)所示。



图2 解码过程
Fig.2 Decoding process

3.4 选择操作

选择操作是算法流程中关键的一步，通过选择操作这一步骤获得质量较高的解，选择的过程中确定选择的策略和选择优质解的比例至关重要。选择的目标性过强可能会导致进入早熟的状态，导致结果局部最优的情况；选择优秀质量解的比例太小会导致淘汰的速度变慢，增加算法的运行时长，降低算法的性能。

选择策略是新种群的4/5用轮盘赌选择，选择的概率符合

大自然的规律随机产生。每次选择大于随机概率的个体到新种群。新种群剩下的1/5的组成部分通过选择策略找到当前解中最优个体，然后复制该个体，复制数量是新种群的1/5。

3.5 交叉操作

交叉操作在遗传算法中是必不可少的，是模拟生物进化过程中两条染色体之间互相交叉重组新的染色体的过程。根据所采用的编码的特点，采用IPOX^[14]交叉操作，基于工序编码。

IPOX操作是在POX操作基础上经改进而形成的。IPOX的具体操作如图3所示。P1、P2和C1、C2表示一对染色体经过交叉后又重新组合成的一对新的染色体。IPOX交叉操作过程为：

- (1)将全部工件分为B1和B2两个集合。
- (2)将P1染色体中有B1集合中的工件序列加入C1，P2染色体中有B2集合中的工件序列加入C2。
- (3)将P2染色体中有B2集合中的工件序列加入C1，P1染色体中有B1集合中的工件序列加入C2。

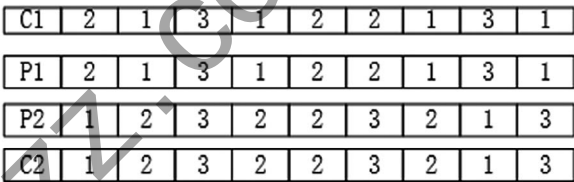


图3 IPOX交叉过程示意图

Fig.3 Schematic diagram of IPOX crossover process

3.6 变异操作

变异操作是模拟生物的基因突变过程，可以防止进入结果过早收敛，在一定程度上增加了种群的多样性，增加了跳出局部极小的可能性。本文采用两种变异的操作方式：第一种在染色体的*n*个基因中，随机产生位置*i*和*j*(*i*<*n*,*j*<*n*)，将基因片段*i*到*j*进行反转操作。为了让搬运小车的数量不变，AGV链同样采用随机反转基因片段的方式。第二种变异操作的方式是随机在机器编码基因串上选择*m*个基因，然后将其位置随机交换。

3.7 局部搜索策略

局部搜索是为了解决非常复杂的问题而出现的一种解决最优问题的算法，最优解的时间有可能是很长的，局部搜索策略为了防止整个算法的寻优时间过长，通过局部搜索寻找近似最优解。局部搜索算法是从爬山法改进而来的，其过程和原理同贪心搜索算法类似，总是从当前解的领域解空间中选择一个最好质量解作为下次迭代过程中的当前解，直到达到一个局部最优解(Local Optimal Solution)。局部搜索算法的基本过程可以描述为：选取一个初始的解，然后通过某种领域动作产生初始解的邻居解，再选择更优的邻居解。一直重复以上过程，直到达到终止条件。

以下是模拟退火算法的基本原理解释，*T*为温度，降温过程具体的表示为：

$$P(dE)=\exp(dE / kT)$$
 (10)

公式(10)中 k 是一个自然常数,并且 $dE < 0$,表示温度退火的过程。从该公式中可以得出:

$\exp(dE/kT)$ 取值是 $(0,1)$, dE 代表下降的温度,那么 $P(dE)$ 的函数取值范围是 $(0,1)$ 。随着温度 T 的降低, $P(dE)$ 会逐渐降低。我们认为向更差解的转变是温度跃迁过程,我们以概率 $P(dE)$ 接受它,当算法结束时所得到的解即为所得到的最佳结果。

4 实验结果与分析(Experimental results and analysis)

本文实验的环境为Windows 10系统,在VS开发环境下求得程序的最优解,并将最后结果与文献中的算法结果相比较。在仿真实验中,设置种群的数目为5,初始交叉概率为0.4,种群的大小为1,000,初始突变概率是0.05,分段函数中的片段数为 $n/5$ 。

表3表示五个工件在五台机器上的加工工艺,每个工件有三道不同的工序,用 O_{ij} 表示,机器用 M 表示。小车的运输时间表如表4所示,机器之间的距离不同,导致不同机器之间的运输时间不同。

表3 五个工件在五台机器上的加工工艺

Tab.3 The process of 5 workpieces on 5 machines

工序	M_1	M_2	M_3	M_4	M_5
O_{11}	—	4	6	2	3
O_{12}	3	—	8	5	7
O_{13}	4	3	—	6	—
O_{21}	—	5	6	4	6
O_{22}	4	5	3	6	4
O_{23}	5	3	7	—	4
O_{31}	5	4	5	6	3
O_{32}	4	—	6	5	7
O_{33}	5	4	3	—	7
O_{41}	4	6	—	5	4
O_{43}	5	—	4	6	3
O_{52}	8	—	6	6	—
O_{53}	5	5	4	—	6

表4 小车运输时间表

Tab.4 AGV transportation schedule

机器	M_1	M_2	M_3	M_4	M_5
M_1	0	2	4	4	3
M_2	2	0	3	3	4
M_3	4	3	0	1	3
M_4	4	3	1	0	2
M_5	3	4	3	2	0

本文测试了实验数据,设置机器的数量为10。从表5的实验结果中的数据可以看出,本文改进的算法性能有一定的提升。图4为改进遗传算法和平均值的对比结果。图5是本次实例中的调度的甘特图。综合图表的数据可以得出本文的算法比平均值的进化速度更快。

表5 机器数量为10的结果对比

Tab.5 Results contrast of machines with a total number of 10

基本遗传算法			本文改进遗传算法		
最佳值	平均值	运输速度	最佳值	平均值	运输速度
18	20	10	16	22	12
20	22	11	18	24	12
22	24	9	22	26	11
21	24	8	19	21	10
20	23	10	17	23	11
19	22	8	16	22	10
22	24	9	20	21	12

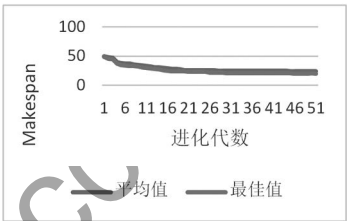


图4 进化代数与Makespan折线图

Fig.4 Evolutionary algebra and Makespan line graph

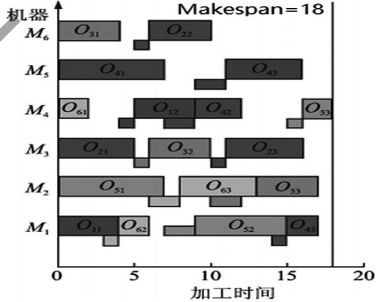


图5 调度甘特图

Fig.5 Gantt chart of schedule

5 结论(Conclusion)

本文考虑AGV的运输时长这一实际生产环境中的因素,让本文的研究更加符合实际生产环境。建立了含有AGV的柔性作业车间调度的数学模型,针对问题自身特点对遗传算法进行改进,引入局部搜索策略加强局部寻优能力,将模拟退火算法作为局部搜索策略加入全局搜索中,增强了算法的收敛性能。遗传算法是进行全局范围的搜索求解,加入局部搜索是为了更快地找到最优近似解或者次优解的个体。本文中的实验测试数据充分体现了本文算法的有效性和可行性。

参考文献(References)

[1] S. Karthikeyan, P. Asokan, S. Nickolas, et al. A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems[J]. Int. J. of Bio-Inspired Computation, 2015, 7(6):386-401.

[2] ZHANG G H, GAO L, SHI Y. An effective genetic algorithm for the flexible job-shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(4):3563-3573.

[3] SHAO X Y, LIU W Q, LIU Q, et al. Hybrid discrete particle