

Dynamic Scheduling Strategy Based on Dynamic Priority Algorithm

Ying Zhuansun

School of Information and Control Engineering, Qingdao University of Technology, Qingdao Shandong
Email: 2967647088@qq.com

Received: Jun. 6th, 2019; accepted: Jun. 20th, 2019; published: Jun. 28th, 2019

Abstract

The core of intelligent RGV dynamic scheduling strategy is to maximize the system efficiency, and priority determination of CNC machining is a powerful guarantee for maximizing efficiency. Based on the dynamic scheduling priority algorithm, this paper sets the priority selection criteria for the workbench by integrating the nearby principle, the shortest waiting time principle and the shortest feeding time principle. And taking the shortest task completion time as the goal, the paper gives a good answer to the path selection and the arrangement of the occurrence of fault in dynamic scheduling. Finally, Matlab is used to verify the maximum number of materials and the maximum working efficiency that can be processed within 8 hours under different cases of instance data. After verification, this strategy has good effectiveness.

Keywords

Dynamic Priority Scheduling Algorithm, Divide-and-Conquer Algorithm, Adaptive Genetic Algorithm

基于动态优先级算法的RGV调度策略

颀孙盈

青岛理工大学信息与控制工程学院, 山东 青岛
Email: 2967647088@qq.com

收稿日期: 2019年6月6日; 录用日期: 2019年6月20日; 发布日期: 2019年6月28日

摘要

针对自动化仓库中直线型轨道RGV调度问题, 以任务完成最多为目标, 结合分治算法, 提出基于CNC动态优先级的调度算法。该算法综合了就近原则、最短等待时间原则、最短上料时间原则, 就RGV的路径选择

与故障情况的解决办法给出了比较好的答案。并利用Matlab验证在实例情形下计算8小时内所能加工的最大物料数和最大工作效率，在实例中，通过与自适应遗传算法对比可验证本算法具有良好的有效性。

关键词

动态优先级调度算法, 分治算法, 自适应遗传算法

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

车间调度是生产加工的关键环节，基本任务之一是合理安排各机器上的待加工工件的顺序，使得最大完工时间最短。轨道式导引小车(Rail-Guided Vehicle, RGV)由于其经济、容量大和灵活的特点，被广泛应用于车间调度[1] [2]。

现有的研究多采用启发式算法解决车间作业的调度问题，然而启发式算法容易陷入局部最优解的困境，不具有优良的全局寻优能力。本文在提出一种基于动态规划策略的 RGV 调度来解决直线往复式轨道装配调度问题，主要采用了分治算法的思想，将整体的大问题分而化之，成为若干个小问题，通过将小问题实现最优解以达到整体的优解[3] [4] [5]。

2. RGV 调度问题描述

智能加工系统由 M 台计算机数控机床 CNC、N 道工序、1 辆轨道式自动引导车 RGV、1 条 RGV 直线轨道、1 条上料传送带、1 条下料传送带等附属设备组成。不同的物料处理时间，移动时间，清洗时间均不同，同一物料不同工序的处理时间不同且不能用同一台 CNC 处理，RGV 两边的 CNC 上下料时间也是不同的(具体如下图 1)，CNC 在工作过程中有一定的概率发生故障，需要人工排除。已知各个步骤的相应时间，求一个比较好的动态调度模型。

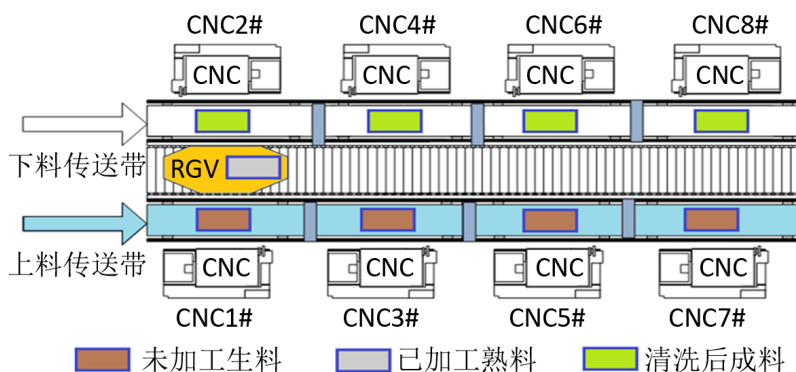


Figure 1. Schematic diagram of intelligent machining system

图 1. 智能加工系统示意图

2.1. 工序过程

1) 智能加工系统通电启动后，RGV 在 CNC1#和 CNC2#正中间的初始位置，所有 CNC 都处于空闲

状态。

2) 在工作正常情况下, 如果某 CNC 处于空闲状态, 则向 RGV 发出上料需求信号; 否则, CNC 处于加工作业状态, 在加工作业完成即刻向 RGV 发出需求信号。

3) RGV 在收到某 CNC 的需求信号后, 它会自行确定该 CNC 的上下料作业次序, 并依次按顺序为其上下料作业, 若此时完成作业, 已加工出熟料, 则进行清洗作业。

4) RGV 在完成一项作业任务后, 立即判别执行下一个作业指令。此时, 如果没有接到其他的作业指令, 则 RGV 就在原地等待直到下一个作业指令。

某 CNC 完成一个物料的加工作业任务后, 即刻向 RGV 发出需求信号。如果 RGV 没能即刻到达为其上下料, 该 CNC 就会出现等待。

5) 系统周而复始地重复 3)至 4), 直到系统停止作业, RGV 回到初始位置。

3. 约束条件

3.1. 变量描述

首先假设调度队列为 $W = \{W_1, W_2, W_3, \dots, W_i, \dots, W_{n-1}, W_n\}$, 其中 n 表示需要加工的工件数。为了加工出最大的工件数, 使机器效率最大化, 我们应使 n 尽可能的大, 即在一定的时间内 RGV 的调度时间尽可能的小[6] [7] [8]。

为了方便说明, 在此对相关变量进行属性描述, 设第 i 台机器的调度时间为 l , RGV 移动时间为 m , 上下料时间为 t , 清洗时间为 t_c , 离 CNC 工作结束还需 t_r (包括物料工作或修理工作), 物料需加工时间为 t_w , 物料已加工时间为 t_b , 此刻时间为 T , 上料开始时间为 t_u , 下料开始时间为 t_d , 故障开始时间为 r , 故障结束时间为 r_e , 人工修理时间为 r_i , 下一个物料上料开始时间为 t'_u , 下一个物料的 RGV 移动时间为 m' , 上下料时间为 t' , 清洗时间为 t'_c 。

接下来, 设置两个决策变量:

1) 决策变量 x

$$x = \begin{cases} 1, & \text{机器正常工作} \\ 0, & \text{机器故障} \end{cases} \quad (1)$$

2) 决策变量 y

$$y = \begin{cases} 1, & \text{机器故障} \\ 0, & \text{机器正常工作} \end{cases} \quad (2)$$

3.2. 约束模型的建立

可建立约束模型如下:

$$o.p \quad \min l$$

$$\text{s.t} \begin{cases} l = m + t + t_c + t_r \\ t_b = T - t_u - t \\ t_r = x(t_w - t_b) + y(r + r_i - T) \\ T \geq t_u \\ t_d - t_u \geq m + t + t_w \\ t'_u - t_u \geq m' + t + t'_c \end{cases} \quad (3)$$

根据上述约束, 严格执行求得最优解。

4. 基于动态优先级算法的模型求解

4.1. 模型算法的建立

该模型算法，建立于多道工序，且需故障排除的情形下。在初始化时用一个数组记录各个步骤的相应时间，同时利用随机函数初始化故障开始时间，需要修理的时间，以及物料号，具体可以阐述为：在处理第 i 个物料时的 t 时刻起，机床 CNC 开始故障，需要人工修理时间为 r 。对于此类问题我们可以增设一个标志位，判断故障与否，当物料处理过程中发生故障，立刻停止处理，开始故障排除，排除故障后复位机床 CNC 状态位，以示 RGV 可以开始工作。在机器的工作时间下：

1. 若 CNC 空闲

1-1 若机器故障

a. 物料数减 1，不删除数据

1-2 若 CNC 上无物料

a. 更新上料开始时间，即总时间；

b. 更新总时间：总时间 += 上料开始时间；

c. CNC 状态置位加工，有物料；

1-3 若 CNC 有物料，则根据 CNC 物料工序道数：

a. 更新总时间：总时间 += 清洗时间；此时总时间即为下料开始时间；

b. 更新此台 CNC 生产总时间：生产总时间 += 需生产加工时间

c. 若为最后一道工序，则生产物料总数加 1

d. 更新新物料上料开始时间，即总时间；

e. 更新总时间：总时间 += 需生产加工时间；

2. 计算各 CNC 剩余加工时间

遍历 CNC：

2-1 若 CNC 故障

故障结束时间 = 故障开始时间 + 人工修理时间；

a. 若总时间 < 故障结束时间

则剩余时间 = 故障结束时间 - 总时间；

b. 否则，剩余时间 = 0，CNC 状态复位；

2-2 若 CNC 空闲，则剩余时间为 0；

2-3 若 CNC 工作

计算已加工时间：已加工时间 = 总时间 - 上料开始时间 - 上下料时间；

a. 若已加工时间 < 生产加工时间

则剩余时间 = 需生产加工时间 - 已加工时间；

b. 否则，剩余时间 = 0，CNC 状态复位；

3. 动态调度，计算最优选择

遍历 CNC：

3-1 若 CNC 无物料

a. 若 CNC 正常

调度时间 = 路径时间 + 上料时间；

b. 若 CNC 故障

调度时间 = 路径时间 + 上料时间 + 剩余加工时间;

3-2 若 CNC 有物料

调度时间 = 路径时间 + 上料时间 + 清洗时间 + 剩余加工时间;

4. 求出最短调度时间及索引位置;

5. RGV 到索引位置等待, 更新时间, 若此时总时间大于机床故障开始时间, 则置位该机床的故障标志位, 回到步骤 1;

该算法综合了就近原则、最短等待时间原则、最短上料时间原则, 得出最短调度时间, 通过此算法, 可以得出较为良好的模型解。

4.2. 模型算法的验证

在进行验证说明前, 为了方便, 首先设置以下变量: 在第 n 台机床 CNC 加工第 i 个物料的第 j 道工序的上料开始时间为 u_0 、上料结束时间为 u_1 、下料开始时间为 d_0 、下料结束时间为 d_1 、故障开始时间为 s_0 、故障结束时间为 s_1 、RGV 移动时间长度为 m_i , 接下来, 建立模型约束条件如下:

$$\text{s.t} \begin{cases} d_0 - u_0 \geq m_i + y_n + t_{ij} \\ s_0 - u_0 = y_0 + (u_1 - s_0) \\ s_1 - u_1 = (u_1 - s_0) + (s_1 - s_0) \end{cases} \quad (4)$$

除了以上公式用于验证, 还有逻辑衔接部分, 即第 n 台机床 CNC 加工完成第 i 个物料的第 j 道工序时(同时上料), RGV 移动至第 n' 台机床 CNC 加工完成第 i 个物料的第 $j+1$ 道工序(同时下料), 此时若下料的物料已完成作业, 则进行清洗熟料, 然后开始在新的机床 CNC 开始加工第 $i+1$ 块物料的第一道工序。在机床 CNC 进行这些操作步骤过程中, 应该是连续衔接的。

5. 自适应遗传算法

为了对比分析本文算法的有效性, 我们引入了自适应遗传算法。

自适应算法采用自适应的交叉变异参数, 当个体差异大时, 缩小差距, 是优秀个体充分发挥, 同时给次等个体一定机会进行进化。当个体差异大时, 它则会扩大差距。自适应遗传算法在一定程度上缓解了陷入局部最优解的困境[1] [9]。

6. 实例分析

以 2018 年全国数学建模比赛为例, 其中故障率为 1%, 故障排除时间介于 10~20 分钟之间, 故障排除后即刻加入作业序列, 其他基本数据如表 1:

Table 1. Instance data

表 1. 实例数据

系统作业参数	时间
RGV 移动 1 个单位所需时间	20
RGV 移动 2 个单位所需时间	33
RGV 移动 3 个单位所需时间	46
CNC 加工完成一个两道工序物料的第一道工序所需时间	400
CNC 加工完成一个两道工序物料的第二道工序所需时间	378
RGV 为 CNC1#, 3#, 5#, 7#一次上下料所需时间	28
RGV 为 CNC2#, 4#, 6#, 8#一次上下料所需时间	31
RGV 完成一个物料的清洗作业所需时间	25

将算法转化为程序，经过 MATLAB 仿真可得出结果，如表 2:

Table 2. Example results
表 2. 实例结果

物料号	CNC 号	工序 1 上料开始时间	工序 1 下料开始时间	CNC 号	工序 2 上料开始时间	工序 2 下料开始时间
1	1	0	428	2	459	868
2	2	28	459	3	510	916
3	3	79	510	4	538	947
4	4	107	538	5	589	995
5	5	158	589	6	617	1026
6	6	186	617	7	668	1074
7	7	237	668	8	696	1105
8	8	265	696	2	868	1277
9	1	428		(此行为故障数据)		
9	3	916	1344	1	1392	1798
10	4	947	1422	5	1473	1879

在 8 小时内，机器作业效率结果如表 3:

Table 3. Results of operational efficiency (unit time: seconds)
表 3. 作业效率结果(单位时间: 秒)

加工总物料数	空闲时间	作业时间	作业效率
237	45,540	184,860	80.2%

根据表 3 可看到此题实例的部分结果，其中，在第一次加工物料 9 时，机床 1 在第 470 s 时发生故障，经人工修理了 13 分钟(1250 s)。经过验证，可知实例结果为正确无误的。通过表 3 可看出，此模型算法的作业效率。

为了验证本模型的良好性，采用了自适应遗传算法，通过多次循环求得局部最优解，得出了 RGV 运送物料以及 CNC 加工物料的顺序，为了方便说明，特意将第 9 块物料加工时设为故障，如表 4。

Table 4. Optimal solution based on genetic algorithm
表 4. 基于遗传算法最优解

RGV 调度路径(CNC 号)	工序道数	物料号
1→2→3→4→5→6→7→8	1	1→2→3→4→5→6→7→8
→2→3→4→5→6→7→8→2	2	1→2→3→4→5→6→7→8

为了更直观的说明，做出了甘特图以及两种算法对比图，如图 2，图 3。其中图 3 中，横坐标表示代码执行次数，纵坐标代表算法效率。可知自适应遗传算法所求局部最优解，往往需要多次执行算法以求得最优解。这不仅仅说明了本模型的解是最优解，具有良好性以及相对较高的作业效率，还说明了本算法的直接，简便。不需要多次执行就可以求得一个比较稳定的优解。

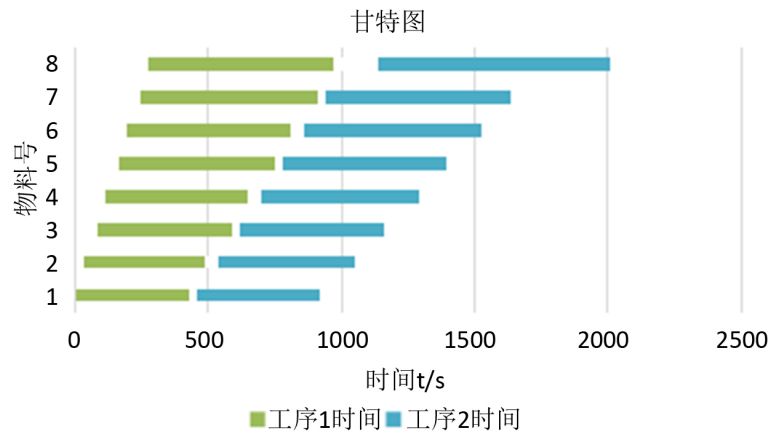


Figure 2. Gantt diagram of genetic algorithm
图 2. 遗传算法甘特图

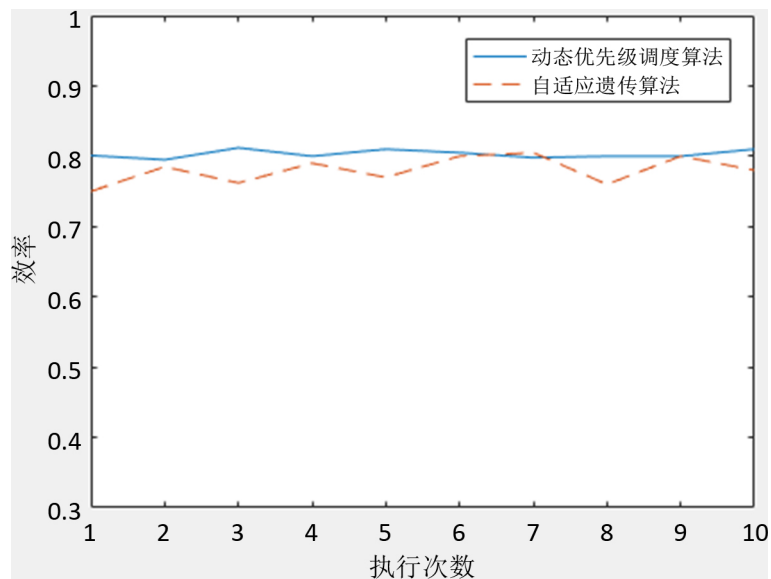


Figure 3. Efficiency comparison of the two algorithms
图 3. 两种算法效率对比图

7. 模型的评价

7.1. 优点

- 1) 本文通过动态调度、分治算法的方式，使 RGV 在每执行完一个步骤之后，计算下一个最优步骤，可达到全局处于最优的状态，计算出最优调度模型，得到一个比较好的模型解。
- 2) 本模型考虑了多道工序的复杂情况，借鉴性强。
- 3) 本文所阐述的模型逻辑严谨，工序之间状况良好，具有较高的作业效率，且与实际关联密切，为实际生产提供了一个良好的可借鉴模型。
- 4) 本模型可用 MATLAB 软件加以验证，使我们的模型具有较高的可信度。

7.2. 缺点

- 1) 本模型参数与变量较多，算法较为繁琐。

2) 本模型算法偏向于暴力解决方式, 对于大型工厂有较高的平台要求。

参考文献

- [1] 龙锋. 基于自适应遗传算法的 W 公司仓库货位分配与优化研究[D]: [硕士学位论文]. 广州: 华南理工大学, 2015.
- [2] 聂峰, 程珩. 多功能穿梭车优化调度研究[J]. 物流技术, 2008, 27(10): 251-253.
- [3] Gao, L., Zhang, G.H., Zhang, L.P., *et al.* (2011) An efficient Memetic Algorithm for Solving the Job Shop Scheduling Problem. *Computers & Industrial Engineering*, **60**, 699-705. <https://doi.org/10.1016/j.cie.2011.01.003>
- [4] Xia, W. and Wu, Z. (2005) An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems. *Computers & Industrial Engineering*, **48**, 409-425. <https://doi.org/10.1016/j.cie.2005.01.018>
- [5] Wu, L.H., Mok, P.Y. and Zhang, J. (2010) An Adaptive Multi-Parameter Based Dispatching Strategy for Single-Loop Interbay Material Handling Systems. *Computers in Industry*, **62**, 175-186. <https://doi.org/10.1016/j.compind.2010.10.010>
- [6] 陈华. 基于分区法的 2-RGV 调度问题的模型和算法[J]. 工业工程与管理, 2014, 19(6): 70-77.
- [7] 林佳良. 基于现实的自动化立体仓库轨道式循环搬运系统构建与调度优化[D]: [硕士学位论文]. 北京: 北京物资学院, 2014.
- [8] 来学伟. 动态规划法在 TSP 问题中的应用[J]. 吉林化工学院学报, 2017, 34(3): 65-67.
- [9] 江唯, 何非, 董一飞, 李东波. 基于混合算法的环形轨道 RGV 系统调度优化研究[J]. 计算机工程与应用, 2016, 52(22): 242-247.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org