

基于BP神经网络的不确定性动态Job-shop调度研究

张喆,刘阶萍,张予昊

(北京交通大学 机械与电子控制工程学院,北京 100044)

摘要:针对作业车间调度的不确定和动态问题,围绕设备故障、订单追加、紧急订单插入3种典型的不确定性情况提出1种基于BP神经网络的重调度方法。当生产过程中发生不确定性事件对原先调度方案产生巨大扰动时,通过已构建且训练好的BP神经网络快速进行响应并生成1个重调度方案,保证整个生产过程高效、有序、稳定地运行。通过仿真实例验证了可行性。

关键词:Job-shop调度;不确定性;动态调度;BP神经网络

中图分类号:TP183 **文献标志码:**B **文章编号:**1671-5276(2019)05-0121-05

Research on Uncertainty Dynamic Job-shop Scheduling Based on BP Neural Networks

ZHANG Zhe, LIU Jieping, ZHANG Yuhao

(School of Mechanical and Electronic Control Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The uncertainty exists in job shop scheduling. It is mainly caused by machine fault, additional order and emergency order insertion. This paper proposes a method of the rescheduling based on BP neural network. When the uncertainty event appears in the production process and it has great influence on the original scheduling, the BP neural network already constructed and trained can be used to quickly generate a re-scheduling scheme which is used to make sure that the effective and smooth operation is guaranteed in the whole production process. The feasibility is verified by the simulation example.

Keywords: job-shop scheduling; uncertainty; dynamic scheduling; BP neural networks

0 引言

作业车间调度(Job-shop, JPS)是一类经典的组合优化问题,大部分都是NP-hard问题^[1]。在实际生产中的不确定性因素,会使车间状态处于动态变化的过程,如何在动态环境下对车间进行有效调度是解决实际调度问题的关键。

随着研究的发展与深入,逐渐形成了各种不同的研究方法。例如Suresh等^[2]通过分支定界法与启发式算法相结合来解决工件到达时间不确定的动态问题。Chong等^[3]通过模拟仿真干扰情况得到不同动态车间环境的调度方案,从而构建一个实时调度机制。纪树新等^[4]根据JSP的数学模型并利用连锁基因编码法解决了动态调度过程中工件到达不确定性的问题。

由于BP神经网络具有较强的非线性映射能力^[5]、自学习和自适应能力,国内外学者也有提出利用神经网络解决Job-shop动态调度问题。Liu H. J.等^[6]在调度系统处于不同的动态情况下,基于获取调度知识的神经网络构建调度决策模型。Min H. S.等^[7]通过神经网络来构建调度选择器,针对不同情况选择不同的调度算法,实现了自适应调度。Park Y.等^[8]提出了一种将BP神经网络与调度规则相结合方法,来求解并行机器的调度问题。这些研究大多是基于神经网络来构建调度决策模型,其对不确定性因素有较快的响应,但是启发式的调度规则并不能保证其

调度结果的近优性。

本文针对基于神经网络构建调度决策模型中的问题,提出了一套解决方案,使调度决策模型不仅能满足对于不确定性事件的快速响应,而且能针对不确定性事件生成一个接近最优的调度方案。

1 Job-shop调度问题模型

Job-shop调度问题可描述为将 n 个工件安排在 m 台设备上加工。

1) 设有 m 台可用设备,即设备集合 $M = \{M_1, M_2, \dots, M_m\}$, M_j 为第 j 台设备 $j=1, 2, \dots, m$;

2) 设有 n 个需要加工工件,即工件集合 $P = \{P_1, P_2, \dots, P_n\}$, P_i 为第 i 个工件;

3) 设 O_{ij} 表示为工件 i 的第 j 道加工工序,则工件 P_i 的加工工序序列为 $O_i = \{O_{i1}, O_{i2}, \dots, O_{im}\}$;

4) 设 s_{ij} 为工件 P_i 在设备 M_j 上开始加工的时间, c_{ij} 为工件 P_i 在设备 M_j 上完成加工的时间,那么 $c_{ij} = s_{ij} + t_{ij}$, t_{ij} 为工件 P_i 在设备 M_j 上完成加工的时间。

Job-shop调度目标需要考虑到各个方面的利益,尽量满足准时交货、缩短生产周期、降低成本等要求,本文选定最小化最大完工时间为调度目标。目标函数为 $\min \max(c_{ik})$ 。

约束条件:

$$c_{jk} - c_{ik} + M(1 - x_{ijk}) \geq t_{jk} \quad (1)$$

$$c_{ik} - c_{jk} + Mx_{ijk} \geq t_{ik} \quad (2)$$

$$c_{ik} - t_{ik} + M(1 - a_{ihk}) \geq c_{ih} \quad (3)$$

式中: $x_{ijk}, a_{ihk} = 0, 1$; $c_{ik} \geq 0$; $i, j = 1, 2, \dots, n; h, k = 1, 2, \dots, m$ 。 M 为一个很大且不为 0 的数。 x_{ijk} 决策变量, 其意义为在设备 M_k 上工件 P_i 早于工件 P_j 到达且先加工, x_{ijk} 为 1, 反之则为 0。 a_{ihk} 也为决策变量, 其意义为工件 P_i 先在设备 M_h 上加工再到设备 M_k 上加工, a_{ihk} 为 1, 反之则为 0。

式(1)、式(2)表示设备上加工工序前后顺序的约束, 即同一时刻内一台设备只能加工一个工件; 式(3)表示工件工序前后顺序的约束。

2 典型不确定问题的重调度机制

1) 设备故障时的重调度

在制造车间原调度方案执行过程中, 某时刻(t), 制造车间的设备(M)发生故障, 在时刻(t)对系统各设备的状态进行检测, 判断故障设备和非故障设备的状态, 执行设备故障时的重调度机制。具体设备故障时的重调度模型如图 1 所示。

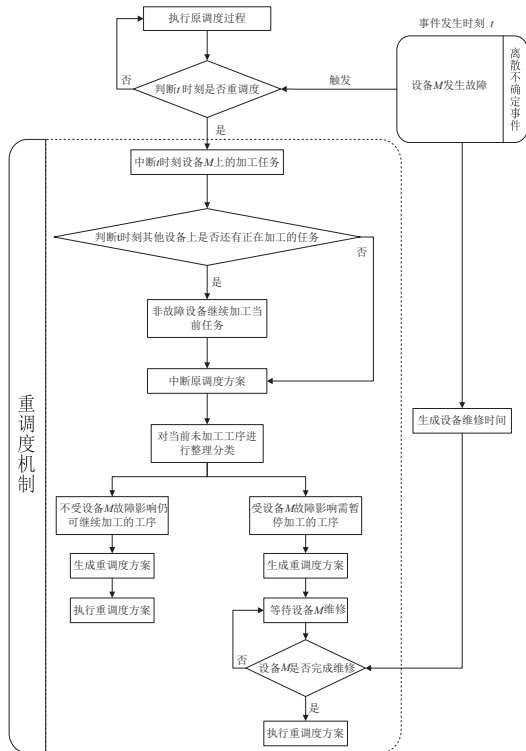


图 1 设备故障时的重调度流程

2) 订单追加时的重调度

在制造车间原调度方案执行过程中, 某时刻(t), 制造车间的工件(P)的订单增加产量, 在时刻(t)对系统的状态进行检测, 判断系统在当前状态下是否需要重调度, 如果不需要则执行原调度。如果需要则执行订单追加情况下的重调度机制。具体订单追加时的重调度模型如图 2 所示。

3) 紧急订单时的重调度

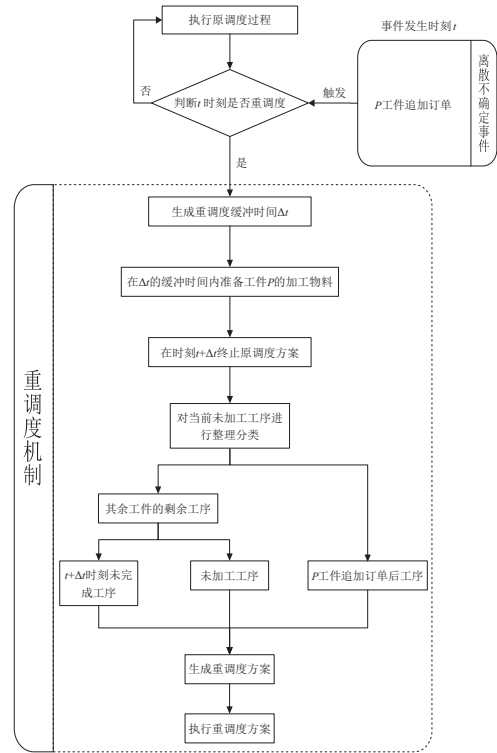


图 2 订单追加时的重调度流程

在制造车间原调度方案执行过程中, 某时刻(t), 制造车间接收到了工件(P')的紧急订单, 在时刻(t)对系统的状态进行检测, 判断系统在当前状态下是否需要重调度, 如果不需要则执行原调度, 如果需要则执行订单追加情况下的重调度机制。具体紧急订单追加时的重调度模型如图 3 所示。

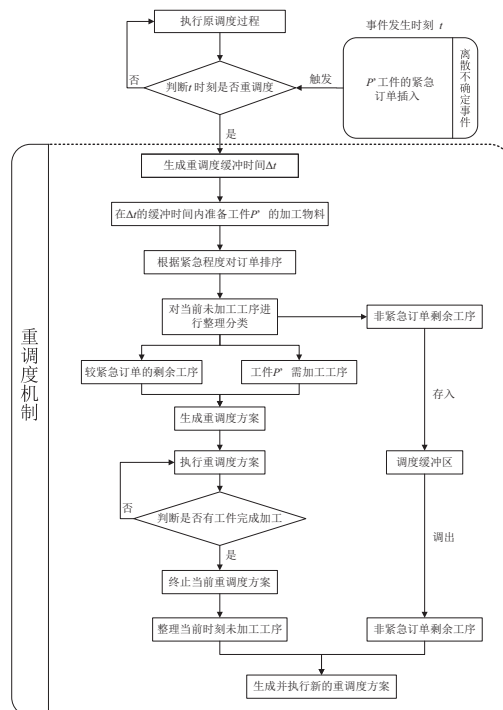


图 3 紧急订单时的重调度流程

3 基于BP神经网络的Job-shop动态重调度

由于离散不确定性因素的重调度策略一般为事件驱动型,以突发事件的发生作为启动信号立即驱动重调度机制,则在这个过程中就需要一个能快速反应的重调度方案产生,否则生产系统长时间停滞会带来很大程度上成本浪费^[9]。利用神经网络的特性、能力来解决这一问题,利用事先训练好的神经网络在重调度时刻快速生成重调度方案。

神经网络的训练过程是提取训练样本集中所蕴含的规律,使网络记忆并运用这些规律来求解类似问题,因此训练样本集的构建要具有代表性。考虑到故障发生的不确定性,假设存在正整数 x ,将整个调度过程平均分为 x 段,则就会产生 $x-1$ 个时间点。假设从时间点1至时间点 $x-1$ 分别发生故障,分别在每个时间点系统将进行重调度,所以针对每个时间点发生故障的情况进行样本构建。

1) 设备故障情况下的样本构建

考虑到故障发生设备的不确定性,每台设备都有发生故障的可能,所以需要针对每台设备发生故障的情况进行样本构建。其训练样本数可由式(4)得:

$$K_1 = 2(x-1)m \quad (4)$$

式中: $x-1$ 表示均分调度过程后时间点的数量, m 表示系统内可用的加工设备数量,由于设备故障发生后需要进行2次重调度所以整个算式需乘2。

2) 订单追加情况下的样本构建

考虑到工件进行订单追加的不确定性,每批工件都有进行订单追加的可能,所以需要针对每批工件进行订单追加的情况进行样本构建。其训练样本数可由式(5)得:

$$K_2 = (x-1)n \quad (5)$$

式中: $x-1$ 表示均分调度过程后时间点的数量, n 表示系统内所需加工的工件批数。

3) 紧急订单插入情况下的样本构建

考虑到系统内工件被紧急工件替换的不确定性,每批

工件都有被紧急订单替换的可能,所以需要针对每批工件被紧急订单替换的情况进行样本构建。其训练样本数可由式(6)得:

$$K_3 = 2(x-1)n \quad (6)$$

式中: $x-1$ 表示均分调度过程后时间点的数量, n 表示系统内所需加工的工件批数,由于紧急订单插入后需要进行2次重调度所以整个算式需乘2。

4 案例仿真与分析

本文采用一个 5×5 车间调度问题来仿真,如表1所示。在Job-shop生产制造系统中有5批不同的需加工件和5台可用设备,可用设备集合为 $M = \{M_1, M_2, M_3, M_4, M_5\}$;需要加工工件集合 $P = \{P_1, P_2, P_3, P_4, P_5\}$ 。

表1 5×5 Job-Shop 车间调度问题

工件	工序数	工序序列
P_1	5	$O_{11}, O_{12}, O_{13}, O_{14}, O_{15}$ $\{1, 30\}, \{3, 40\}, \{5, 45\}, \{2, 25\}, \{4, 30\}$
P_2	5	$O_{21}, O_{22}, O_{23}, O_{24}, O_{25}$ $\{3, 50\}, \{5, 35\}, \{1, 30\}, \{2, 25\}, \{4, 50\}$
P_3	5	$O_{31}, O_{32}, O_{33}, O_{34}, O_{35}$ $\{5, 40\}, \{1, 25\}, \{2, 45\}, \{3, 35\}, \{2, 40\}$
P_4	5	$O_{41}, O_{42}, O_{43}, O_{44}, O_{45}$ $\{2, 25\}, \{1, 50\}, \{3, 20\}, \{4, 35\}, \{5, 30\}$
P_5	5	$O_{51}, O_{52}, O_{53}, O_{54}, O_{55}$ $\{5, 25\}, \{4, 45\}, \{1, 30\}, \{5, 45\}, \{3, 35\}$

在MATLAB环境下,用遗传算法对模型求解,参数设置为:种群规模设定为200、交叉概率设定为0.8、变异概率设定为0.1、最大迭代次数设定为250次。遗传算法求解结果如图4所示。

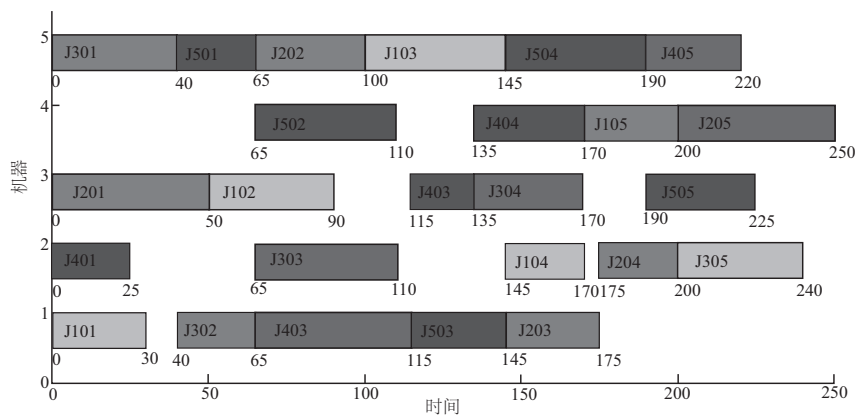


图4 5×5 Job-Shop 调度方案甘特图

对上述车间调度问题分别在设备故障情况、订单追加情况和紧急订单插入3种情况进行仿真。每种情况下随机选定一个时间发生突发事件,用训练好的神经网络在此

时间点生成重调度方案。将神经网络根据重调度机制生成的方案与同一时刻换作遗传算法生成的重调度方案进行对比,验证神经网络生成调度方案的有效性。

1) 设备故障情况仿真

假设在 $t=90$ 时刻,设备 M_3 发生故障,设备维修所需要的时间 $t_B=95$,则 $t=90$ 时刻之后的工序 O_{43} 将无法进行加工,对工序 O_{43} 的后续相关联工序分析可以得表 2。

表 2 设备 M_3 故障后的剩余工序

剩余工序种类	工序
需等待 M_3 维修	$O_{34}, O_{35}, O_{43}, O_{44}, O_{45}, O_{55}$
可立即重调度	$O_{13}, O_{14}, O_{15}, O_{23}, O_{24}, O_{25}, O_{53}, O_{54}$

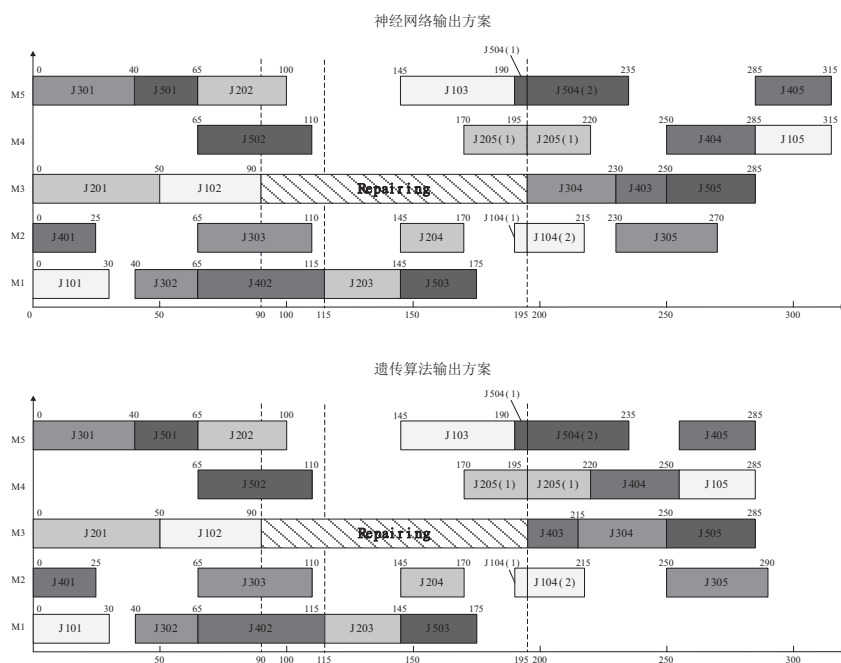
由于设备 M_3 维修所需要的时间 $t_B=95$,则在 170 时刻设备 M_3 重新可用,对此刻剩余的进行整理分类,如表 3

所示。

表 3 M_3 维修完成后的剩余工序

剩余工序种类	工序
第 1 次重调度未完成工序	$O_{14}(2), O_{15}, O_{25}(2), O_{54}(2), O_{55}$
需等待 M_3 维修	$O_{34}, O_{35}, O_{43}, O_{44}, O_{45}, O_{55}$

如图 5 所示为分别通过遗传算法和神经网络在 $t=90$ 时刻设备 M_3 发生故障情况的整个调度过程对比,可知,神经网络输出结果已经十分接近最优结果,其最大完工时间($c_{\max}$)比遗传算法的结果多 8.6%。

图 5 $t=90$ 时刻神经网络和遗传算法的调度过程对比

2) 订单追加情况仿真

假设在 $t=120$ 时刻,工件 P_2 订单需要追加,追加为原先的 1 倍,则 $t=120$ 时刻工件 P_2 所有的加工工序时长翻倍,已完成加工的工序以相同时间再加工一次,此时刻对剩余工序进行整理分类可得表 4。

表 4 $t=120$ 时刻的剩余工序

剩余工序种类	工序
工件 P_2 剩余工序	$O_{13}(2), O_{14}, O_{15}, O_{34}, O_{35}, O_{43}(2), O_{44}, O_{45}, O_{53}(2), O_{54}, O_{55}$
其余工件剩余工序	$2O_{23}, 2O_{24}, 2O_{25}, O_{21}, O_{22}$

如图 6 所示为通过遗传算法和神经网络在 $t=120$ 时刻工件 P_2 订单进行追加的整个调度过程对比,可得:神经网络输出结果已经十分接近最优结果,其最大完工时间 $c_{\max}$ 比遗传算法的结果多 4.8%。

3) 紧急订单情况仿真

假设在 $t=75$ 时刻,工件 P_e 的紧急订单突然到达,假设原先 5 个工件中 P_3 的紧急程度最小被替换,则此时刻对剩余工序进行整理分类可得表 5。

表 5 $t=75$ 时刻的剩余工序

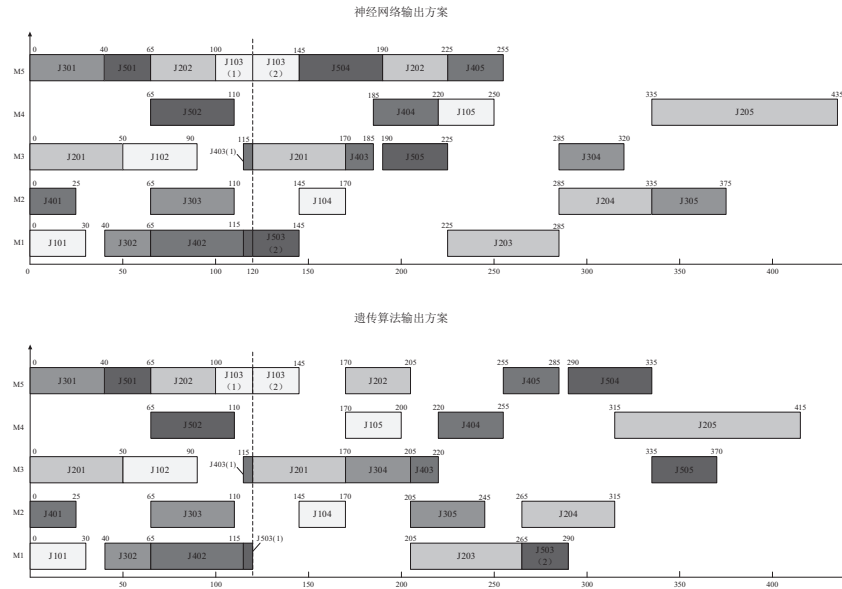
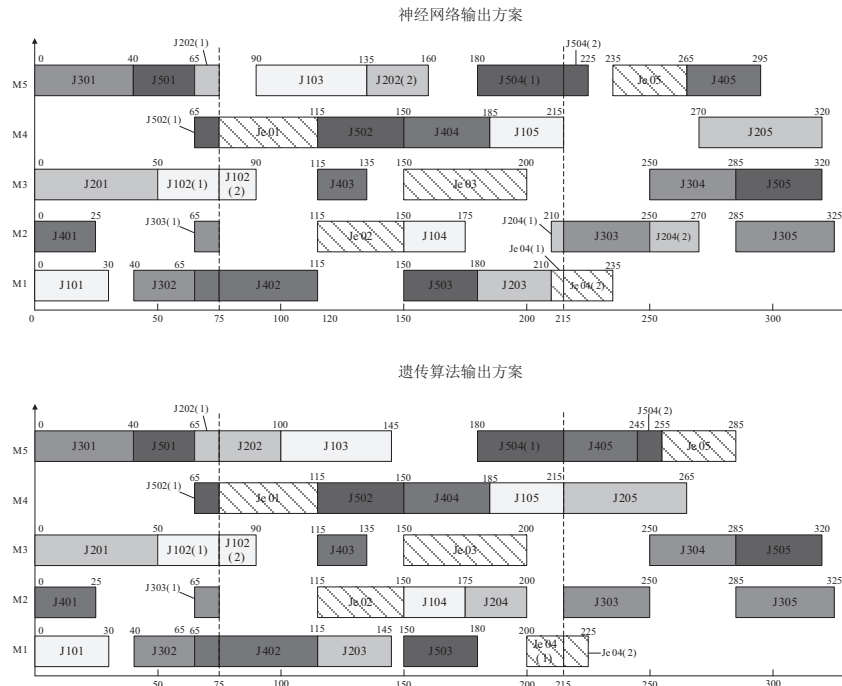
剩余工序种类	工序
剩余工序	$O_{12}(2), O_{13}, O_{14}, O_{15}, O_{22}(2), O_{23}, O_{24}, O_{25}, O_{42}(2), O_{43}, O_{44}, O_{45}, O_{52}(2), O_{53}, O_{54}, O_{55}$
工件 P_e 的工序	$O_{e1}, O_{e2}, O_{e3}, O_{e4}, O_{e5}$

在重调度执行过程中, $t=215$ 时刻,工件 P_1 的所有工序完成加工,此时将工件 P_3 的剩余工序和此时其余工件的剩余工序一起进行整理得表 6。

表 6 $t=215$ 时刻的剩余工序

剩余工序种类	工序
其余剩余工序	$O_{24}(2), O_{25}, O_{45}, O_{54}(2), O_{55}, O_{e4}(2), O_{e5}$
P_3 的剩余工序	$O_{33}(2), O_{34}, O_{35}$

如图 7 所示为通过遗传算法和神经网络在 $t=75$ 时刻工件 P_e 的紧急订单突然插入的整个调度过程对比,神经网络输出的调度方案在具体工序序列的顺序和遗传算法稍有不同,但其最大完工时间($c_{\max}$)与遗传算法的结果相同。

图6 $t=120$ 时刻神经网络和遗传算法的调度过程对比图7 $t=75$ 时刻神经网络和遗传算法的调度过程对比

4) 结果分析

3种情况仿真分析对比结果如表7~表9所示。结果表明神经网络输出的调度结果有效,其接近遗传算法所得的最优值。

表7 设备故障案例的结果对比

设备故障案例情况			遗传算法结果($c_{\max}$)	神经网络结果($c_{\max}$)	误差/%
发生时刻	故障设备	维修时间			
35	M_1	110	340	365	7.3
90	M_3	95	290	315	8.6
115	M_5	80	305	305	0

表8 订单追加案例的结果对比

订单追加案例情况		遗传算法结果($c_{\max}$)	神经网络结果($c_{\max}$)	误差/(%)
发生时刻	追加订单			
75	P_3	425	450	5.9
120	P_2	415	435	4.8
160	P_5	510	510	0

通过表7~表9所示的结果可知,神经网络输出的重调度方案与遗传算法求解得到的重调度方案之间的误差较小。在工程实际运用中,可根据实际案例的工程指标对

(下转第139页)