

基于多 Agent 的柔性作业车间调度研究

潘 颖^{*1,2}, 孙 伟¹, 马 跃¹, 马沁怡¹

(1. 大连理工大学 机械工程学院, 辽宁 大连 116024;

2. 大连海洋大学 机械学院, 辽宁 大连 116023)

摘要: 针对多目标柔性作业车间调度问题, 构造了基于多 Agent 的车间调度模型, 研究了多 Agent 之间的交换协调机制. 提出一种改进遗传算法并封装在策略 Agent 中, 通过对编码规则、解码算法与交叉、变异算子进行改进, 提高了调度优化算法的实用性和优化效果, 实现了多目标动态调度, 提高了系统的适应性和健壮性. 某企业车间应用实例证明其可以满足车间调度高效、稳定的要求.

关键词: 代理; 多 Agent 系统; 柔性作业车间调度; 多目标; 遗传算法

中图分类号: TP18 **文献标志码:** A

0 引言

近年来, 具有多目标优化的柔性作业车间调度问题(flexible job-shop scheduling problem, 简称 FJSP)成为研究的热点^[1]. 在实际调度过程中, 需要综合考虑生产的各个环节和因素, 对多个目标进行分析决策, 而通常这些目标之间又是相互冲突的. 与传统的作业车间调度比较, FJSP 是传统作业车间调度问题(job-shop scheduling problem, JSP)的扩展, 是更复杂的 NP-Hard 问题^[2], 其任务中的每道工序可在多台机床上进行, 并且在不同的机床上所需时间不同. FJSP 减少了机器约束, 扩大了可行解的搜索范围, 增加了问题的难度. 因此, FJSP 除了要解决 JSP 中的确定加工顺序外, 还要解决某工序由哪台机床进行的问题, 因而求解更加复杂.

近年来, 多目标 FJSP 的研究出现了很多新的智能优化方法^[3~7], 使得多目标 FJSP 的研究方法向多元化方向发展. 其中, 遗传算法利用生物进化机制, 在一个较大的初始解空间中通过优胜劣汰的方法进行优化求解, 具有寻优能力强、计算速度快、原理简单、鲁棒性好、通用性强、不受限制性

条件的约束, 具有隐含并行性和全局解空间搜索能力的特点, 因此, 在生产调度领域得到广泛应用, 用遗传算法解决 FJSP 近年也取得了丰硕的成果^[8].

同时, 多 Agent 系统(MAS)由于其本身的分散自治性、网络合作性、结构开放性、智能性等特点, 也已成为调度研究领域的前沿和热点^[9].

针对多目标 FJSP 的特点, 本文提出一种基于多 Agent 的多目标柔性作业车间调度方法, 运用 MAS 对柔性车间生产调度过程建模, 其中特别构建策略 Agent 来实现调度及生产上的实时调控, 并在策略 Agent 中封装改进的遗传算法, 最后通过实例对所提的调度方法进行验证.

1 基于 MAS 的调度模型

1.1 问题描述

设生产系统有 n 个工件, 每个工件有 p 个工序. 各工序均可在 m 台设备中的一台或多台上进行. 由于各设备的性能、特点不同, 某设备 k 对各个工序 O_{ij} 的加工时间为 T_{ij}^k (其中工件号 $i \in N = \{1, 2, \dots, n\}$, 工序号 $j \in P = \{1, 2, \dots, p\}$, 设备号 $k \in M = \{1, 2, \dots, m\}$).

取设计变量

$$F_{ij}^k \in \{0,1\}; i \in N, j \in P, k \in M \quad (1)$$

$$ORD_{ij} \in \{1,2,\dots,pm\}; i \in N, j \in P \quad (2)$$

其中0-1变量 F_{ij}^k 表示工序和设备的选择关系, $F_{ij}^k = 1$ 表示工序 O_{ij} 在设备 k 上进行; 整数变量 ORD_{ij} 表示工序 O_{ij} 的加工顺序, 数值小者顺序靠前。

多目标 FJSP 优化数学模型如下: 式(3) 是根据权重 W 不同, 分别取最长完工期、单台设备最大负载、全部设备总负载和总加工成本, 或它们的任意组合最小化为优化目标, 并计算工序的完成时刻; 式(4) 表示设备 k 的工作时间; 式(5) 表示全部设备的总工作时间; 式(6) 表示总加工成本; 式(7) 保证每工序只在台设备上; 式(8) 保证同一工件各工序的顺序; 式(9) 保证 ORD_{ij} 取得 $1 \sim pm$ 的组合; 式(10) 保证各工序的开始加工时刻或者为零, 或者要在同一设备前一工序完工之后。即

$$\begin{aligned} \min z &= W_1 \frac{z_1}{z_1^*} + W_2 \frac{z_2}{z_2^*} + W_3 \frac{z_3}{z_3^*} + W_4 \frac{z_4}{z_4^*} \\ z_1 &= \max(ET_{ij}); i \in N, j \in P \\ z_1^* &= \min z_1 \\ z_2 &= \max(MET^k); k \in M \\ z_2^* &= \min z_2 \\ z_3 &= MWT \\ z_3^* &= \min z_3 \\ z_4 &= COST \\ z_4^* &= \min z_4 \end{aligned} \quad (3)$$

$$\begin{aligned} \text{st. } ET_{ij} &= ST_{ij} + \sum_{k=1}^m F_{ij}^k T_{ij}^k; \\ i &\in N, j \in P, k \in M \\ MET^k &= \max(F_{ij}^k ET_{ij}); \\ i &\in N, j \in P, k \in M \end{aligned} \quad (4)$$

$$MWT = \sum_{k=1}^m MET^k \quad (5)$$

$$COST = \sum_{k=1}^m MET^k C^k \quad (6)$$

$$\sum_{k=1}^m F_{ij}^k = 1; i \in N, j \in P \quad (7)$$

$$ST_{ij+1} \geq ST_{ij}; i \in N, j \in \{1,2,\dots,p-1\} \quad (8)$$

$$ORD_{ij} \neq ORD_{i'j'};$$

$$i, i' \in N, j, j' \in P \text{ 且 } i \neq i', j \neq j' \quad (9)$$

$$\begin{aligned} ST_{ij} &\geq \text{sgn}(ORD_{ij} - ORD_{i'j'}) F_{ij}^k F_{i'j'}^{k'} ST_{i'j'}; \\ k &\in M, i, i' \in N, j, j' \in P \text{ 且 } i \neq i', j \neq j' \end{aligned} \quad (10)$$

式中: 中间变量 ST_{ij}, ET_{ij} 表示工序 O_{ij} 的加工开始与完成时刻; MET^k, MWT 表示设备 k 的工作时间和全部设备的总工作时间; $COST$ 为总加工成本; C^k 为设备 k 的工时成本。

1.2 基于 MAS 的车间调度模型

本调度系统主要由设备 Agent (device agent, DA)、任务 Agent (task agent, TA) 和策略 Agent (strategy agent, SA) 组成, 如图 1 所示。TA 与 DA、TA 与 SA、TA 与 TA 之间可以通信。TA 在确定好 DA 后, 会在 DA 对应的设备时间轴上选择合适的位置, 也即安排工序至此设备, 实现调度功能。

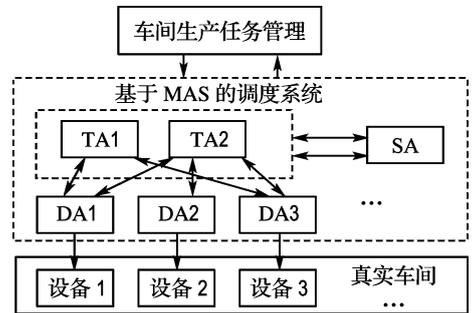


图 1 基于 MAS 的柔性作业车间调度模型

Fig. 1 FJSP model based on MAS

1.2.1 设备 Agent 设备 Agent (结构见图 2) 是车间中加工设备的代理, 每台加工设备对应一个设备 Agent, 设备 Agent 通过设备接口可获得加工设备的技术参数和设备状态等信息, 再把加工任务发送给加工设备执行。设备 Agent 可与任务 Agent 交互信息, 实现任务资源分配, 并向任务 Agent 反馈状态信息。

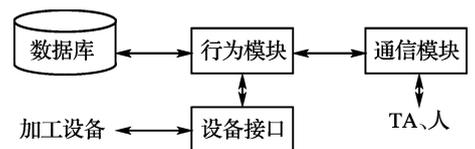


图 2 设备 Agent 结构

Fig. 2 Structure of device agent

Agent 可分为慎思型、反应型和混合型。设备 Agent 本身不需要具有判断和推理能力,它的主要作用是动态地标定自身状态,并激发其他 Agent 的进程,因此,设备 Agent 被设计成反应型 Agent。

1.2.2 任务 Agent 任务 Agent(结构见图 3)为慎思型。它根据到达的任务动态地生成。每道工序会对应一个任务 Agent,该工序的生产调度过程及相关数据都由其对应的任务 Agent 管理。任务 Agent 撤销即代表该工序已完成。日志数据库中保存生产调度过程中形成的数据。

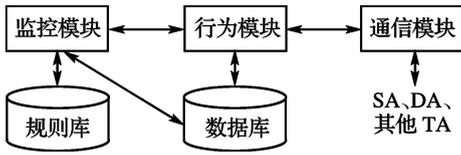


图 3 任务 Agent 结构

Fig. 3 Structure of task agent

任务 Agent 通过与策略 Agent 交互,将设备 Agent 投招标和优化目标等信息及时传递给策略 Agent,然后从策略 Agent 得到调度方案后调配,最终将生产任务下达到设备,确定工序的加工设备和开始加工时间。

任务 Agent 与其他任务 Agent 交互信息,传递完成任务所必须满足的前继和后继条件,在满足先后顺序约束的条件下使任务能实现实时推进;与设备 Agent 交互信息,为任务分配适当的资源,并监测和监控任务的执行情况及资源的负载情况;与策略 Agent 交互信息,实现实时动态调度。

1.2.3 策略 Agent 策略 Agent(结构见图 4)为慎思型。其与任务 Agent 交互可实现动态实时调度,且因其内部封装了算法和调度规则,其在接受任务 Agent 的信息后,能提供不同调度方案并返回任务 Agent;提供人机接口,可调整算法参数,制定新的调度方案。

策略 Agent 是基于 MAS 的车间调度核心部分,对车间的动态调度有效执行起关键作用。考虑到 job-shop 中存在的多目标优化、柔性作业、有 JIT 要求等调度问题,在策略 Agent 中特别封装了一种改进遗传算法,以实现车间调度所应达到

的高效、稳定。本文并没有把算法规则封装在任务 Agent 中,而是独立设计了一个策略 Agent,这既保证了系统的可重构性、柔性,也便于系统的维护和升级。策略 Agent 模块中设置了人机接口,保证当生产过程中有插单等紧急情况时,有可提供人工干预调度的能力。

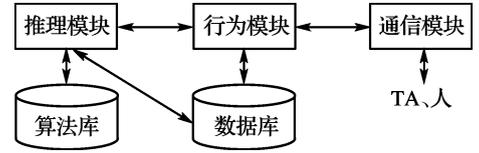


图 4 策略 Agent 结构

Fig. 4 Structure of strategy agent

1.2.4 Agent 间的协作模型 多 Agent 系统运作的基础是彼此间的通信与协作。本文采用合同网方式实现多 Agent 的控制,通过任务招标、投标和订立合同进行合作。

(1)Agent 间的协作

利用招标-投标-中标机制,任务 Agent 确定完成某一任务的设备 Agent,设备 Agent 与此任务 Agent 签订合同后执行相应生产任务。在整个生产过程中,任务 Agent 将遇到的紧急突发状况(如人员和设备的选择与冲突等)反馈至策略 Agent,由策略 Agent 提供相应解决方案,根据情况或者由系统自带算法调整重排生产任务,或者通过人机交互,手动重排生产任务。

(2)合同网机制

任务 Agent: $TA = \{TA_1, TA_2, \dots, TA_{np}\}$, np 为分解后的工序数。传递如下消息内容: $\{Q_{ij}, D_{ij}, L_{ij}, C_{ij}, DS_{ij}\}$; Q_{ij} 为工序 O_{ij} 的作业量, D_{ij} 为该任务所需设备表, L_{ij} 为完工期限, C_{ij} 为其他约束, DS_{ij} 为任务描述。然后由任务 Agent 并发地向可加工该工序的所有的设备 Agent 广播,消息格式为 $\{Q_{ij}, O_{ij}, L_{ij}, C_{ij}, DS_{ij}\}$; Q_{ij} 为需执行的作业量, O_{ij} 为作业名称, L_{ij} 为完工期限, C_{ij} 为其他约束, DS_{ij} 为任务描述。设备 Agent 收到招标书后,根据标书内容(任务时间、优先级等)和自己的能力、状态等决定是否投标。按如下格式返回任务 Agent: $\{TB_k, Q_k, T_k\}$; TB_k 为该设备可执行的时间段表, Q_k 为保证期限的最大作业量, T_k 为期限截止前该设备可利用时间。任务 Agent 根据收到

的标书和自身的策略进行评价,选出中标者,并将任务分配给中标的设备 Agent,确认消息,格式为 $\{O_{ij}, Q_{ij}, L_{ij}, C_{ij}, DS_{ij}\}$; O_{ij} 为作业名称, Q_{ij} 为需执行的作业量, L_{ij} 为完工期限, C_{ij} 为任务约束, DS_{ij} 为任务描述^[10]. 中标者更新自身知识库,执行子任务并返回结果,如果最大作业量小于任务要求作业量,任务 Agent 会考虑由数个设备承担;如仍无法办到,则通知策略 Agent 无法承担所分配任务;策略 Agent 若根据一定的规则调整未果,则发出警报,操作人员可通过人机接口,手动强制执行或调整生产计划,必要时或寻求外协.

最后,任务 Agent 提交给车间生产任务管理系统任务的完成情况,同时注销相应的任务 Agent. 在建立合同关系之后,任务 Agent 还需监测任务的执行情况,只要发现设备 Agent 处于过载、故障等状态,就必须进行资源的转移分配. 如任务 Agent 自身不能调停处理,需反馈至策略 Agent,策略 Agent 会根据需要修改计划. 所采用的多 Agent 的控制过程(基于合同网机制)如图 5 所示.

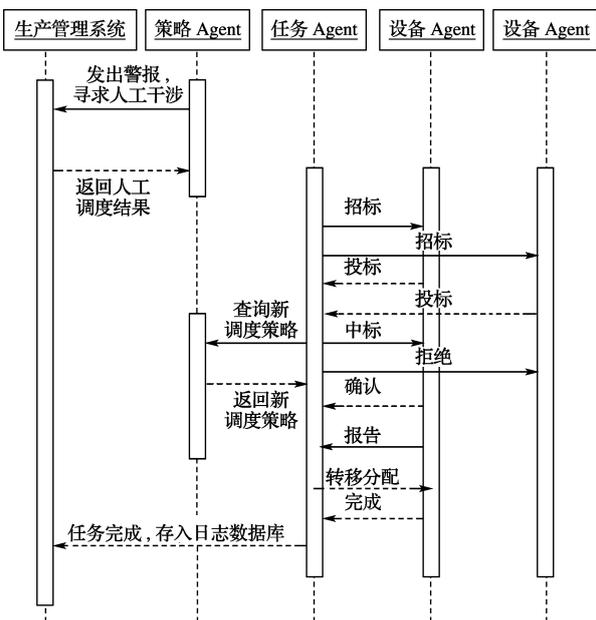


图 5 合同网中投招标机制图

Fig. 5 Tendering-bidding mechanism graph of contract net

2 基于遗传优化的 MAS 调度算法设计

如前文所述,策略 Agent 为本 MAS 调度系

统的核心一环,其中封装了改进的遗传算法以解决实际车间多目标优化的柔性作业调度问题,常见柔性作业车间调度加工时间如表 1 所示.

本文对于非完全柔性工序(即存在某设备 k 不能用来完成工序 O_{ij}),人为地将加工时间 T_{ij}^k 改为很大值,如取为 1 000. 这样在优化过程中,算法会自动规避这样的工序设备选择. 同时引入对所有设备加工时间都是 0 的虚工序,可以保证各工件的工序数一致. 于是原问题工时表变为表 2.

表 1 一般的柔性作业车间调度问题加工时间
Tab. 1 Processing time of general FJSP

工件	工序	加工时间				
		M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	O ₁₁	25	—	15	18	—
	O ₁₂	—	20	—	14	10
	O ₁₃	—	—	15	13	11
J ₂	O ₂₁	—	4	—	4	3
	O ₂₂	9	—	—	6	8
	O ₂₃	—	—	—	3	2
J ₃	O ₃₁	6	—	—	4	3
	O ₃₂	—	—	7	5	5

表 2 规划为标准 FJSP 后的加工时间表

Tab. 2 Processing time after transforming into standard FJSP

工件	工序	加工时间				
		M ₁	M ₂	M ₃	M ₄	M ₅
J ₁	O ₁₁	25	1 000	15	18	1 000
	O ₁₂	1 000	20	1 000	14	10
	O ₁₃	1 000	1 000	15	13	11
J ₂	O ₂₁	1 000	4	1 000	4	3
	O ₂₂	9	1 000	1 000	6	8
	O ₂₃	1 000	1 000	1 000	3	2
J ₃	O ₃₁	0	0	0	0	0
	O ₃₂	6	1 000	1 000	4	3
	O ₃₃	1 000	1 000	7	5	5

2.1 多目标优化

实际车间通常会有不同的目标要求,生产中常见的优化目标为最长完工期、单台设备最大负载、全部设备总负载和总加工成本最小化. 本文分两步实现多目标规划:第一步分别取各单目标为遗传算法的适应度函数,得到各自的较优值 z_1^* 、 z_2^* 、 z_3^* 和 z_4^* ;第二步取各目标的归一化加权组合 $z = W_1 \frac{z_1}{z_1^*} + W_2 \frac{z_2}{z_2^*} + W_3 \frac{z_3}{z_3^*} + W_4 \frac{z_4}{z_4^*}$ 作为适应

度函数进行遗传优化。权重可根据实际情况进行设置调整,或者自动调整(如侧重总加工成本最小化,可设置其相应权重系数大于其他目标函数权重系数)。实验证明,通过合理调整权重,该方法能较好地协调各目标,快速获得具有实际应用价值的解。

2.2 遗传算法设计

(1) 编码方式

本文采用直接编码,由两部分组成:第一部分为基于工序排列的编码,用来确定工序的加工顺序;第二部分为基于机器分配的编码,用来选择每道工序的加工机器。编码

$O_{11}, O_{12}, O_{13}, O_{21}, O_{22}, O_{23}, O_{31}, O_{32}, O_{33} | O_{11}, O_{12}, O_{13}, O_{21}, O_{22}, O_{23}, O_{31}, O_{32}, O_{33}$

(2, 4, 6, 3, 5, 8, 1, 7, 9 || 1, 2, 2, 3, 3, 3, 4, 2, 5)

表示工序 O_{31} 第 1 个安排,在 M_4 上加工;工序 O_{11} 第 2 个安排,在 M_1 上加工;其余依次类推。

(2) 群体初始化

随机生成 population size 个染色体个体。

(3) 适应度函数

直接取目标函数为适应度函数。

(4) 解码方式

解码分为 3 个步骤:首先按基因中的顺序与设备选择,依次将各工序投射对应到设备;然后考虑各工序加工时间进行整理,获得各工序的开始和结束时刻、各设备工作时间和结束时刻、车间工作终止时刻 $ENDTIME$;最后按照单目标或多目标要求生成适应度函数。解码过程的流程图如图 6 所示。 $PUTIN(x, y)$ 为插入矩阵, x 代表插入顺序, $PUTIN(x, 1)$ 、 $PUTIN(x, 2)$ 、 $PUTIN(x, 3)$ 分别代表工件号、工序号和设备号; $ST(i, j)$ 、 $ET(i, j)$ 、 $MET(k)$ 、 $MWT(k)$ 分别代表工序开始时刻、结束时刻、设备空闲时刻和设备工作时间; $F(i, j)$ 是工序插入完成标志; pn 为工序总数。

(5) 遗传算子的设计

①选择算子:选择操作采用精英机制(elite),即直接将适应度最好的 elite count 个染色体个体直接遗传到下一代。

②交叉与变异算子:交叉操作和变异操作的父代在精英机制后剩下的个体中产生,采用锦标赛 tournament 即从随机选取的 tournament size

个染色体个体中选择适应度最好的个体作为交叉与变异的父代;在生成的父代中按 crossover fraction 进行交叉,余下部分进行变异操作。

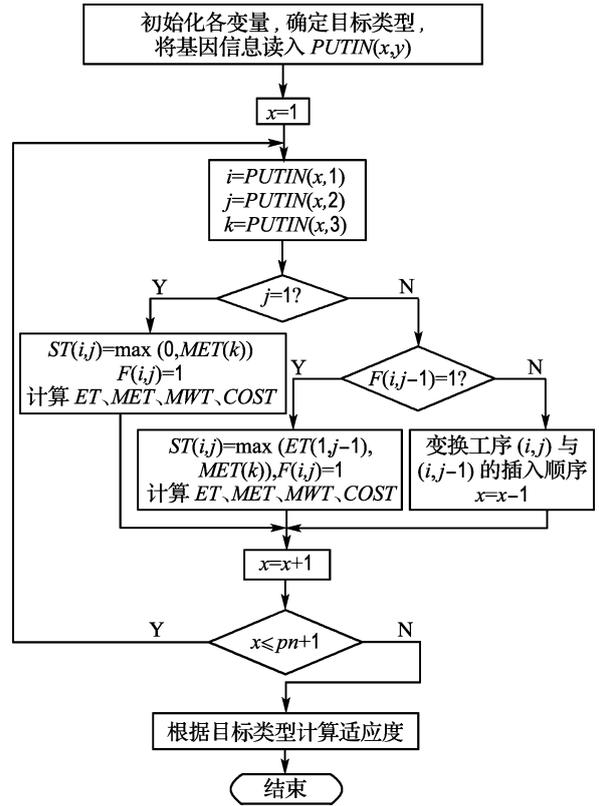


图 6 解码流程图

Fig. 6 Flow chart of decoding

a. 交叉算子

选择两个父代, 等比随机进行交叉。有两种方式: 总体取父代 1, 某插入顺序取父代 2, 这会出现不合理排列, 在解码过程会解决; 总体取父代 1, 某设备选择改为父代 2。

b. 变异算子

按比例随机选择进行 3 种交叉: 调换插入顺序; 调换设备; 随机更改设备。其中第 3 条能保证排除弱智方案。

可见, 以上的算子可以保证子代的基因值总是有效的, 这样就保证了算法运行时不会生成无效基因, 从而有利于提高算法的可靠性和效率。

(6) 遗传算法的操作步骤

①初始化, 随机生成 population size 个染色体个体;

②求解各单目标最优值;

③构造等权多目标函数;

④利用 GA 搜索;

⑤判断优化结果能否被接受,是则终止,否则继续;

⑥人工:根据方案指标与实际需求,决定调整方向;自动:根据方案指标与范围约束,决定调整方向,或根据模糊规则决定调整方向;

⑦根据⑥的决策调整权重;

⑧转④.

(7)运行参数

遗传算法的运行参数为 $population\ size = 200$, $elite\ count = 7$, $tournament\ size = 3$, $crossover\ fraction = 0.2$, $generations = 200$.

3 实例验证

山西太原某公司机械车间的调度属于典型的多目标柔性 job-shop 调度,该制造单元有 9 台进口设备,包括并行机和多功能加工中心,各加工设备代号、每单位工时的加工成本,及对不同工件各工序的加工时间如表 3、4 和 5 所示.

应用本文提出的基于多 Agent 的车间调度模型及优化调度算法开发了生产管理软件.图 7 为多目标调度参数界面,可选择调度任务,并可选择预调度先查看调度结果.如有不合适之处,可人工调整;如果符合,则可选择正式调度并执行“开始”.在优化参数一栏,可人工修改遗传算法参数,否则就是系统默认的参数值.优化目标部分,在各优化目标后面方框中的数字代表不同权重,数值大小可调.优化参数和优化目标权重值的调整会影响调度时间和结果,可根据工厂车间实际要求做出调整.多目标优化的选择,只要在优化目标处勾选相应的目标即可.如图 8 显示的是以成本和工期最小为多目标优化的甘特图,在优化目标处选中“最小成本”和“最短工期”,“0%”表明该道工序尚未进行,完成进度为 0.图 9 显示的是最小成本、最短工期和最小全部设备负载为多目标的优化调度甘特图.调度结果显示为不同设备加工不同工序的甘特图.

表 3 设备代号对应表

Tab. 3 Corresponding device code

设备名称	设备代号	设备名称	设备代号
车床	M ₁	中心	M ₆
车床	M ₂	热处理炉	M ₇
钻床	M ₃	磨床	M ₈
钻床	M ₄	磨床	M ₉
中心	M ₅		

表 4 设备加工成本

Tab. 4 Processing cost of device

设备	工时成本/(元·min ⁻¹)	设备	工时成本/(元·min ⁻¹)
M ₁	50	M ₆	130
M ₂	65	M ₇	20
M ₃	30	M ₈	75
M ₄	50	M ₉	80
M ₅	120		

表 5 设备加工时间表

Tab. 5 Processing time of device

工件	工序	加工时间/min								
		M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇	M ₈	M ₉
J ₁ 上盖	1 车	25				18	15			
	2 车		20			14	10			
	3 钻				15	13	11			
J ₂ 调整 螺母	1 钻			4		4	3			
	2 车		9			6	8			
	3 钻				3	3	2			
J ₃ 侧盖	1 车	6				4	3			
	2 钻				7	5	5			
J ₄ 六角 螺母	1 热								6	
	2 车	2				1	1			
	3 钻				2	2	1			
	4 车		4			2	3			
	5 车	3				2	2			
	6 调								6	
J ₅ 扭杆	1 车	15				12	8			
	2 热								5	
	3 磨									5
	4 磨									11
J ₆ 隔套	1 车	12				8	6			
	2 钻			7		4	5			
	3 热								6	
	4 磨									17

注:“热”为“热处理”简称;“调”为“调质”简称

综上所述,该车间优化调度系统可动态及时调整调度结果,并可人工调整相应参数以满足不同的调度要求,最终实现车间实时动态多目标调度.

当前位置：计划编制 > 调度计划 > 调度参数

调度名称: GS70207068预调度

调度任务: 调入任务1 | 调度类型: 预调度 正式调度

优化参数:

种群数量	200	精英数	7
锦标赛数	3	交叉概率	0.2
进化代数	200		

优化目标:

最小成本 1 最短工期 2 最小单台设备总负载 3 最小全部设备负载 4

开始 停止 调度结果

图 7 多目标调度参数界面

Fig. 7 Parameters interface of multi-objective scheduling

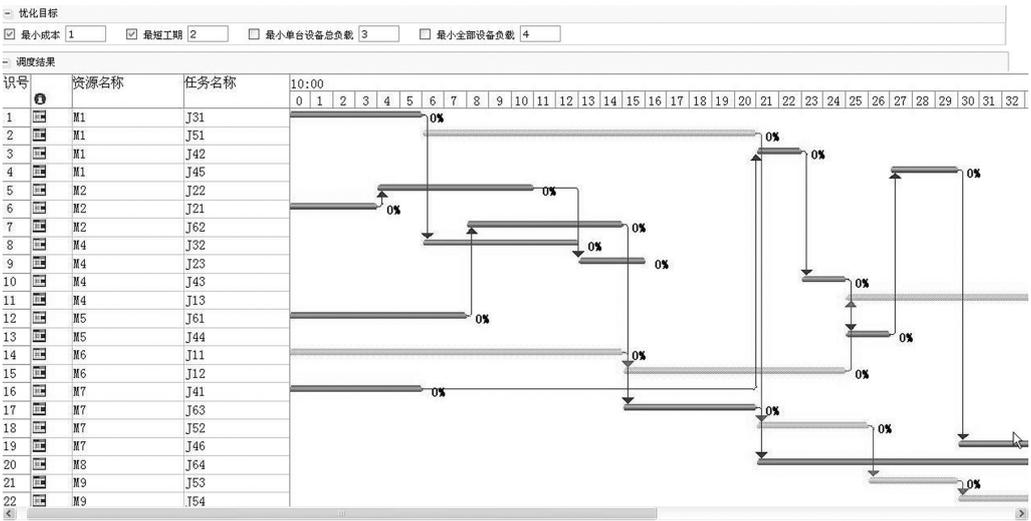


图 8 多目标优化(成本+工期最小)调度结果

Fig. 8 Scheduling result of multi-objective optimization (minimum cost and makespan)

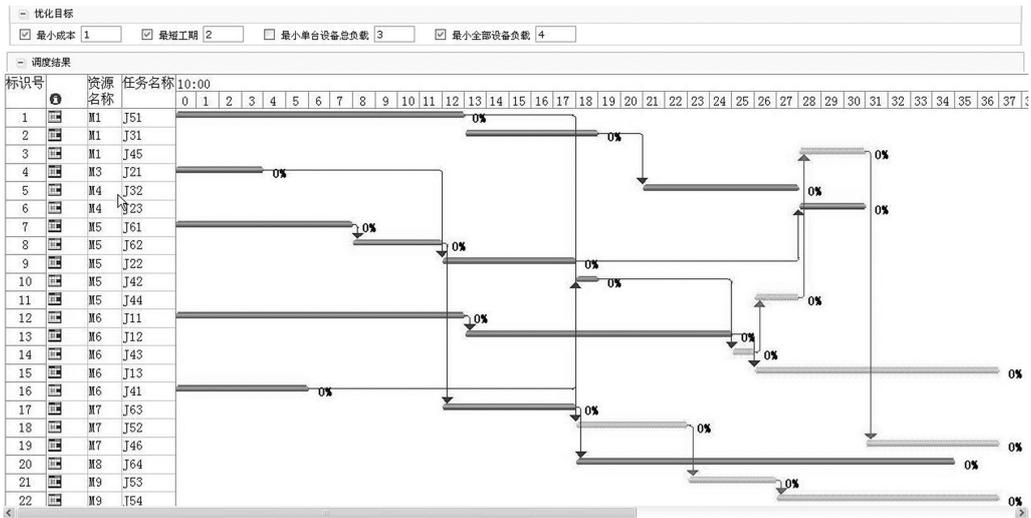


图 9 多目标优化(成本+总负载+工期最小)调度结果

Fig. 9 Scheduling result of multi-objective optimization (minimum cost, total load and makespan)

4 结 论

本文针对多目标柔性作业车间调度问题提出的基于多 Agent 的车间调度系统结构简单,便于系统维护和升级,并保证了系统的可重构性和柔性,能满足车间柔性作业、多目标优化等实际问题的要求.实践证明,该调度系统适用于复杂的柔性车间作业调度环境,适应性和自治性较高,能保证车间生产持续优化地进行,应用前景广阔.

参考文献:

- [1] GAO Jie, GEN Mitsuo, SUN Lin-yan, *et al.* A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems [J]. **Computers & Industrial Engineering**, 2007, **53**(3):149-162
- [2] BLAZEWICZ J, FINKE G, HAOPT G. New trends in machine scheduling [J]. **European Journal of Operational Research**, 1988, **37**(2):303-317
- [3] CHEN H, IHLOW J, LEHMANN C. A genetic algorithm for flexible job-shop scheduling [C] // **IEEE International Conference on Robotics and Automation**. Detroit:IEEE, 1999:1120-1135
- [4] JIA H Z, NEE A Y C, FUH JY H, *et al.* A modified genetic algorithm for distributed scheduling

- problems [J]. **International Journal of Intelligent Manufacturing**, 2003, **14**(3):351-362
- [5] HO N B, TAY J C. GENACE: An efficient cultural algorithm for solving the flexible job-shop problem [C] // **IEEE International Conference on Robotics and Automation**. Detroit:IEEE, 2004:1759-1766
- [6] KACEM I, HAMMADI S, BORNE P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems [J]. **IEEE Transactions on Systems, Man, and Cybernetics, Part C**, 2002, **32**(1):1-13
- [7] KACEM I, HAMMADI S, BORNE P. Pareto — optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic [J]. **Mathematics and Computers in Simulation**, 2002, **60**(2):245-276
- [8] PEZZELLA F, MORGANTI G, CIASCETTI G. A genetic algorithm for the flexible job-shop scheduling problem [J]. **Computers & Operations Research**, 2008, **35**(5):3202-3212
- [9] 饶运清,谢 畅,李淑霞. 基于多 Agent 的 Job Shop 调度方法研究[J]. **中国机械工程**, 2004, **15**(5):873-877
- [10] 潘 颖,张文孝. 基于多 agent 的离散制造业制造执行系统框架研究[J]. **计算机应用研究**, 2009, **26**(1):244-249

Research on multi-agent-based flexible job-shop scheduling

PAN Ying^{*1,2}, SUN Wei¹, MA Yue¹, MA Qin-yi¹

(1. School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China;

2. Mechanical Engineering Institute, Dalian Ocean University, Dalian 116023, China)

Abstract: Aiming at multi-objective flexible job-shop scheduling problem, job-shop scheduling model based on multi-agent is constructed. Exchange and coordination mechanism between agents is also researched. What's more, an improved genetic algorithm is presented and encapsulated in strategy agent. Through the improvement of coding rules, decoding algorithm, crossover and mutation operators, the practicality and optimization effect of scheduling optimization algorithm are enhanced, such methods realize multi-objective dynamic scheduling and improve system's adaptability and robustness. The example in certain enterprise workshop testifies that the scheduling model can satisfy high efficient and stable demands of job-shop scheduling.

Key words: agent; multi-agent system; flexible job-shop scheduling; multi-objective; genetic algorithm