

# 混流柔性加工单元自动导引小车的调度优化

徐立云, 陈延豪, 高翔宇, 李爱平

(同济大学 机械与能源工程学院, 上海 201804)

**摘要:** 多品种混流柔性加工单元中的自动导引运输车 (AGV) 数量和运行路径直接影响单元的运行效率. 在考虑产品加工工时、批量需求、设备物理位置等约束下, 以最小化搬运任务时间为优化目标, 基于改进 Memetic 算法, 通过编码和搜索机制的调整, 对不同 AGV 数量以及不同设备加工任务分配方案条件下的调度策略进行协同优化求解, 有效避免了迭代过程中易出现非法解的状况, 从而获得了 AGV 最优调度路径. 最后通过实例验证了该方法的可行性和有效性.

**关键词:** 自动导引运输车 (AGV) 调度; 改进 Memetic 算法; 柔性加工单元; 混流生产

中图分类号: TH165

文献标志码: A

## AGV Scheduling Optimization in Mixed Flexible Manufacturing Cell

XU Liyun, CHEN Yanhao, GAO Xiangyu, LI Aiping

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

**Abstract:** In multi-product mixed flexible manufacturing cell, the number of automated guided vehicles (AGVs) and their routings will directly affect the cell's efficiency. Considering the constraints such as processing time, batch requirement and facility location, a mathematical model was built aimed at minimizing the handling time of AGVs. To co-optimize the number of AGVs and their schedulings under different distributions of manufacturing tasks, an improved Memetic algorithm was proposed with the adjustment of coding and search mechanism, which effectively avoids the problem that conventional crossovers easily result in illegal solutions and the optimized AGV routing is also obtained. At last, a case study is illustrated to prove the validity of the proposed method.

**Key words:** AGV scheduling; improved Memetic algorithm; flexible manufacturing cell; mixed production

随着电子计算机和自动化技术的革新, 制造业愈加偏向智能化发展. 自动导引运输车 (AGV) 作为柔性制造系统关键设备之一, 为系统的智能分配、灵活调度和准确控制提供了快捷、有效的物流输送.

目前, 国内外学者对 AGV 调度问题做了许多研究. Ho 等<sup>[1]</sup>提出了区域分区设计和动态区域控制的路径规划方法, 在规避碰撞的同时, 实现了 AGV 在不同区域间的负载平衡. Rezapour 等<sup>[2]</sup>提出了在 AGV 路径规划过程中, 对机床布局以循环的方式重新设计, 使得 AGV 调度路径最短. 侯玲娟等<sup>[3]</sup>针对不确定需求和时间下的车辆路径随机问题, 提出了一种带有自适应机制的规划模型. 胡彬等<sup>[4]</sup>针对柔性制造系统多 AGV 路径规划问题, 提出了一种基于时间窗的动态路径规划算法, 有效避免了 AGV 间的冲突问题. 贺丽娜等<sup>[5]</sup>结合预先规划算法和实时规划算法, 提出了基于先验决策的路径规划方法, 有效避免了锁死及碰撞问题. Guan 等<sup>[6]</sup>提出了一种改进的类电磁机制, 对可重构系统下的单向 AGV 系统调度路径进行了优化设计. Caridá 等<sup>[7]</sup>将分层模糊规则模型和自适应遗传模糊控制 2 种方法应用于 AGV 调度过程中, 有效提升了调度系统的敏感度. Fauadi 等<sup>[8]</sup>关注于多 AGV 运输任务的动态分配, 提出了一种先进的组合拍卖方法以调解 AGV 任务分配过程. Zheng 等<sup>[9]</sup>提出了一种基于 tabu 方法的启发式算法, 以完工时间最短为目标, 优化柔性制造系统中机床操作的加工顺序与 AGV 路径. Digani 等<sup>[10]</sup>以双层控制构型和路线定义自动算法为依据, 提出了一种整体协调方法, 使系统的柔性和全局效率得到了提升. Umar 等<sup>[11]</sup>提出了一种混合遗传算法, 采用自适应权重方法有效优化了完工时间、调度时间与物流成本.

上述文献的研究重点多针对固定加工方案下的

收稿日期: 2016-09-21

基金项目: 国家科技重大专项 (2011ZX04015-022); 上海市科委“创新行动计划”项目 (16111106502); 上海市经信委产学研项目 (沪 CXY-2013-31)

第一作者: 徐立云 (1973—), 男, 教授, 博士生导师, 工学博士, 主要研究方向为智能制造、系统建模与优化、产品数字化设计与管理.  
E-mail: Lyxu@tongji.edu.cn

AGV 调度问题,而对于由具备完整加工能力的柔性加工设备组成的加工单元而言,可实现同族不同产品的混流加工.在这种情况下,由于不同产品的加工工时、批量需求的差异以及上下料点与不同柔性加工设备间物理位置的差异,使得各设备间加工任务的合理配置存在较大意义.本文针对多品种混流柔性加工单元中柔性设备的加工任务分配方式和 AGV 数量对调度的影响,对多 AGV 调度优化问题进行了研究,建立了多 AGV 调度的数学模型,基于改进的 Memetic 算法对模型有效求解,得到不同 AGV 数量下机床加工任务的最优分配方案和 AGV 最优调度路径.

### 1 问题描述

在一个多品种混流柔性加工单元(FMC)中有  $d$  种工件共  $i$  个需在  $m$  台柔性机床上进行加工.每种工件所需加工工时不同,加工需求数量亦不同.机床按照一定的布局方式分布在线上,每台机床具备完整加工能力,任务开始前各台机床的加工任务未定,均可选择一种工件进行加工. AGV 作为物料输送设备,但数量未定. AGV 搬运任务分为上料和下料.上料搬运任务指 AGV 从当前位置移至上料区取料(如当前位置为上料区,直接取料),按工件类型运至目标机床上料;下料搬运任务指 AGV 从当前位置移至工件所在机床取料,运至下料区下料,如图 1 所示.上、下料点在不同机床间物理位置上存在差异,AGV 上下料时脱离主运行轨道停靠,以避免对后续 AGV 的影响.当 AGV 和机床处于空载状态时,将上料区的所有搬运任务作为一个任务集进行调度,针对各机床加工任务不同的分配方案和 AGV 数量,将任务集内所有搬运任务按一定次序分配给各台 AGV,AGV 按分配结果执行任务.

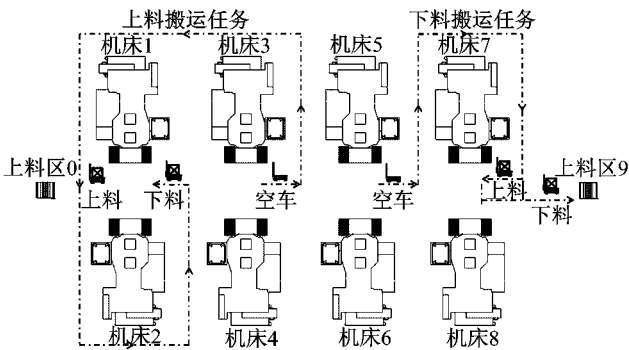


图 1 FMC 中 AGV 搬运示意图  
Fig.1 AGV handling diagram in FMC

基于实际问题 and 研究必要性,作出如下假设:

- (1)上、下料区和机床之间的节点位置关系以及 AGV 节点间行走所需时间已经确定.
- (2)各类型工件所需加工时间已经确定.
- (3)每台机床只负责一种类型工件的加工,每次只能加工一个工件.
- (4)机床的缓冲区容量不影响 AGV 调度.
- (5)所有工件初始位置在上料区.
- (6)对机床输入/输出缓存区采用先进先出(FIFO)管理规则.
- (7)每辆 AGV 初始位置在上料区,AGV 完成搬运任务后,直接前往下一任务所在目标机床.
- (8)AGV 可执行任一搬运任务.
- (9)AGV 在搬运过程中不会发生干涉和碰撞.

该柔性加工单元的优化目标为任务集内 AGV 完成搬运任务的时间最短.

### 2 数学模型

为便于问题描述,针对图 1 对相关符号进行说明,如表 1 所示.

表 1 参数定义  
Tab.1 Description of parameters

符号	符号所表示意义
$k$	AGV 编号, $k = \{1, 2, \dots, P_k\}$ , $P_k$ 表示 AGV 数量
$d$	工件类型, $d = \{1, 2, \dots, P_d\}$ , $P_d$ 表示工件类型数量
$m$	机床编号, $m = \{1, 2, \dots, P_m\}$ , $P_m$ 表示机床数量
$m_0$	上料区
$m_1$	下料区
$i$	工件编号, $i = \{1, 2, \dots, P_i\}$ , $P_i$ 表示工件数量
$s$	任务编号, $s = \{1, 2, \dots, P_s\}$ , $P_s$ 表示任务数量
$T_{im}$	工件 $i$ 在机床 $m$ 上加工所需时间
$T_{bm}$	机床 $m$ 开始空闲时间
$N_{im}$	工件 $i$ 分配的目标机床 $m$
$T_{bim}$	工件 $i$ 在机床 $m$ 上加工开始的时间节点
$T_{cim}$	工件 $i$ 在机床 $m$ 上加工完成的时间节点
$T_{bks}$	小车 $k$ 开始执行工件 $i$ 搬运任务 $s$ 的时间节点
$T_{eks}$	小车 $k$ 完成工件 $i$ 搬运任务 $s$ 的时间节点
$I_{kis}$	小车 $k$ 开始执行搬运任务 $s$ 时工件 $i$ 所在的机床
$K_{ks}$	小车 $k$ 开始执行搬运任务 $s$ 时小车所在的机床
$T_{(m_1, m_2)}$	小车从机床 $m_1$ 移动到 $m_2$ 所用时间
$T_{ks}$	小车 $k$ 执行任务 $s$ 所用时间
$T_{kis}$	执行下料搬运任务 $s$ 时,装载工件 $i$ 前小车 $k$ 所用时间

约束模型构建如下:

$$\sum_{i=1}^{P_i} \delta_{kis} \leq 1, \quad \forall k, s \tag{1}$$

$$\sum_{d=1}^{P_d} \beta_{md} = 1, \quad \forall m \tag{2}$$

$$T_{bks} + T_{kis} \geq T_{cim}, \quad \forall m, i, k, s \tag{3}$$

$$K_{k0} = m_0, \quad \forall k \quad (4)$$

$$T_{bim} \geq T_{bkis} + T_{(K_{ks}, m_0)} + T_{(m_0, N_{im})}, \quad \forall m, i, k, s \quad (5)$$

$$T_{ekis} - T_{bkis} = \delta_{kis} T_{ks}, \quad \forall i, k, s \quad (6)$$

$$T_{eim} = T_{bim} + T_{im}, \quad \forall i, m \quad (7)$$

其中,

$$\delta_{kis} = \begin{cases} 1, & \text{工件 } i \text{ 是第 } k \text{ 台 AGV} \\ & \text{的第 } s \text{ 个任务} \\ 0, & \text{其他} \end{cases}$$

$$\beta_{md} = \begin{cases} 1, & \text{工件种类 } d \text{ 在机床 } m \text{ 上加工} \\ 0, & \text{其他} \end{cases}$$

式(1)表示每辆 AGV 每次只完成一个工件的搬运任务;式(2)表示每台机床只能加工一种工件;式(3)表示下料时,AGV 必须在工件加工完成后装载工件;式(4)表示 AGV 的初始位置是上料区;式(5)表示工件在 AGV 完成上料搬运任务后开始加工;式(6)表示 AGV 搬运任务结束时间是开始时间加上搬运时间;式(7)表示工件完工时间等于开始时间加上加工时间。

采用机床约束矩阵  $R_{M \times N} = (r_{uv})$  来为每台机床分配加工任务. 其中,  $M = A_{P_m}^m$ , 表示方案数量;  $N = P_m$ , 表示机床数量;  $v = 1, 2, \dots, M$ , 表示方案编号;  $u = 1, 2, \dots, N$ , 表示机床编号. 矩阵每 1 行表示一种分配方案, 如  $P_m = 4, v = 3$ , 矩阵  $R$  第 3 行为  $[r_{31} \ r_{32} \ r_{33} \ r_{34}] = [3 \ 1 \ 4 \ 2]$ , 表示第 3 种分配方案中机床 1 加工工件类型 3, 机床 2 加工工件类型 1, 机床 3 加工工件类型 4, 机床 4 加工工件类型 2. 如此, 可得到机床约束矩阵

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1N} \\ \vdots & & \vdots \\ r_{M1} & \cdots & r_{MN} \end{pmatrix}$$

需要注意的是, 当  $P_d = P_m$  时, 即每种工件由且仅由一台机床加工, 矩阵  $R$  中不会出现相同的行矩阵; 当  $P_d < P_m$  时, 即部分类型工件会由多台机床加工, 矩阵  $R$  会出现重复. 因此, 实际生产中, 矩阵  $R$  应根据工厂实际要求对约束进行调整.

根据以上约束模型, 目标函数构建如下:

$$\min \left\{ \max_{\substack{1 \leq k \leq P_k \\ 1 \leq s \leq P_s}} \left\{ \sum_{i=1}^{P_i} \delta_{kis} T_{ks} \right\} \right\}$$

最小化 AGV 完成各自搬运任务时间最大值, 即完成任务集内所有搬运任务的总时间最短. 根据 AGV 搬运任务的不同,  $T_{ks}$  的计算分为上料与下料 2 种工况. 当 AGV 的搬运任务为上料时, 搬运时间是 AGV 从当前位置移至上料区取料的时间加上从上

料区移至目标机床的时间, 如下所示.

$$\delta_{kis} T_{ks} = T_{(K_{ks}, m_0)} + T_{(m_0, N_{im})}, \quad \forall i, k, s \text{ 且 } \eta_{ks} = 1$$

当 AGV 的搬运任务是下料时, 如果工件已完工, 搬运时间是 AGV 从当前位置移至目标机床的时间加上从目标机床移至下料区的时间. 如果工件未完工, 搬运时间是工件的完工时间加上从目标机床运至下料区的时间, 如下所示:

$$\delta_{kis} T_{ks} = T_{(K_{ks}, I_{kis})} + T_{(I_{kis}, m_1)}, \quad \forall i, k, s \text{ 且 } \eta_{ks} = 0, C_{mi} = 1$$

$$\delta_{kis} T_{ks} = T_{eim} + T_{(I_{kis}, m_1)}, \quad \forall i, k, s \text{ 且 } \eta_{ks} = 0, C_{mi} = 0$$

$$\eta_{ks} = \begin{cases} 1, & \text{第 } k \text{ 台 AGV 第 } s \text{ 个任务为上料任务} \\ 0, & \text{第 } k \text{ 台 AGV 第 } s \text{ 个任务为下料任务} \end{cases}$$

$$C_{mi} = \begin{cases} 1, & \text{工件 } i \text{ 已在加工单元 } m \text{ 上加工完成} \\ 0, & \text{工件 } i \text{ 未在加工单元 } m \text{ 上加工完成} \end{cases}$$

### 3 基于改进的 Memetic 算法的优化方法

Memetic 算法<sup>[12-14]</sup>是一种将遗传算法和局部搜索相结合的新型智能算法, 基于遗传算法的全局搜索过程中对单个个体进行局部搜索, 保证迭代过程中所有个体均可到达局部最优, 提高了算法的求解效率.

#### 3.1 构造个体

考虑到机床加工任务分配和 AGV 数量对调度的影响, 个体编码采用多参数编码方式. 个体由多个基因片段组成, 每个基因片段由 3 位数组成, 第 1 位代表 AGV 编号, 第 2 位代表机床编号, 第 3 位代表工件编号, 表示第  $k$  台 AGV 本次的搬运任务是将第  $i$  个工件送至第  $m$  台机床. 这种编码方式可清楚表达机床加工任务分配信息和 AGV 搬运任务信息, 便于计算机解码. 需要注意的是, 因第 3 位为工件编号, 每个工件包含上料任务和下料任务各 1 次, 所以含有相同工件编号的基因片段需出现 2 次, 个体所含基因片段数为  $2P_i$  个. 个体编码为“123246375”时, 含义如图 2 所示.

按照上述编码方式以基因片段为单位对个体进行解码, 解读出 AGV 任务分配方案和 AGV 搬运路径. 通过数学模型求解 AGV 调度过程, 记录 AGV 每个搬运任务的完成时间、每个工件的开工时间节点和完工时间节点, 并获得工件集内所有搬运任务的完成时间为个体的目标函数值, 即评估值 value, 代表每个个体的优劣程度.

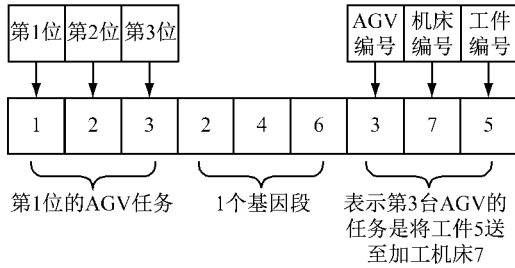


图 2 个体编码方式

Fig.2 Individual coding mode

3.2 遗传算子

(1) 选择算子

考虑到本算法在初代种群生成过程中存在一定随机性,如采用传统的轮盘赌方法进行算子选择,会存在 2 个问题:一是在进化前期, value 值高的个体被选中概率过高,导致子代继承过多,子代个体单一,从而使搜索陷入局部搜索;二是在进化后期,个体 value 值差距较小,无法做到有效选择.因此,在进行轮盘赌之前,将个体的 value 值由小到大排序,获得个体排序序号  $l$ ,按式(8)计算个体的适应度值  $fitness(j)$ .通过式(9)计算出个体的  $cFitness(j)$ ,作为其进行轮盘赌时被选中的概率上限,如下所示:

$$fitness(j) = a(1 - a)^{l-1}, \quad a \in (0, 1) \quad (8)$$

$$cFitness(j) = \frac{\sum_{j=1}^l fitness(j)}{C} \quad (9)$$

式中:  $C = \sum_{j=1}^{popSize} fitness(j)$ ,其中  $popSize$  为种群大小.

采取此方法,个体被选中的概率只取决于它在种群中的排位  $l$  与参数值  $a$  的大小,个体与其他个体 value 值的绝对差值大小并不影响到其遗传到子代的数量,从而在一定程度上避免了上述 2 种情况.其中,  $a$  的取值影响个体被选取的概率,  $a$  越趋向于 0,排位相近的 2 个个体被选中的概率越接近, value 值低的个体越不易被淘汰,算法越不易陷入局部搜索.反之,  $a$  越趋向于 1,不同排位的个体被选中的概率差异性越大, value 值高的个体易被选中,算法收敛速度加快.在本文案例中取  $a=0.6$ .

(2) 交叉算子

考虑到编码方式的特殊性(第 3 位上工件编号具有唯一性),个体如按传统遗传算法(GA)的交叉方式会生成非法解,需对个体交叉重组后对其进行修正,方法如下:提取个体  $g_1, g_2$  对应位置上的基因片段  $b_1, b_2$ ,根据  $b_1, b_2$  的第 3 位(工件编号)  $i_1, i_2$  选择修正方法.如  $i_1 = i_2$ ,将  $g_1$  中工件编号为  $i_1$  的基因片段的第 2 位改为  $b_2$  的第 2 位,  $g_2$  的修正方法与

$g_1$  相同;如  $i_1 \neq i_2$ ,将  $g_1$  中工件编号为  $i_1$  的基因片段的第 2 位改为  $b_2$  的第 2 位,将  $g_1$  中 2 个工件编号为  $i_2$  的基因片段的后 2 位改为  $b_1$  的后 2 位,  $g_2$  的修正方法与  $g_1$  相同.修正方法如图 3、4 所示.经过改进的交叉方式不会产生非法解,有利于提升可行解的搜索效率,增加 AGV 任务分配方案的多样性.

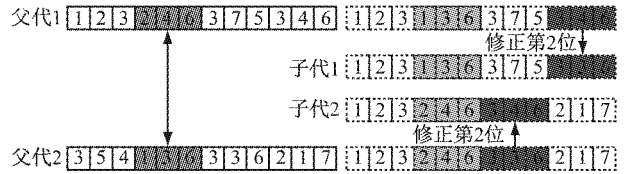


图 3  $i_1 = i_2$  时修正方法

Fig.3 Updating method when  $i_1 = i_2$

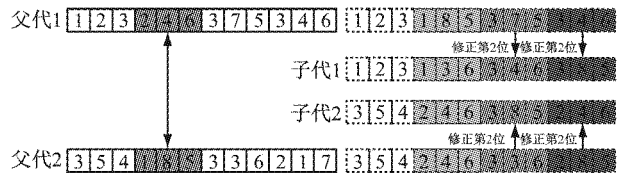


图 4  $i_1 \neq i_2$  时修正方法

Fig.4 Updating method when  $i_1 \neq i_2$

采用爬山算法实现局部搜索,搜索机制如下:

在算法中对于每一个个体,随机生成 2 个  $[1, 2P_i]$  之间不等的整数,取这 2 个数位上的基因片段相互交换,产生邻域范围解,如图 5 所示.通过多次搜索,计算每次交换前后个体的评估值 value,选出值最小的个体作为局部搜索后的最优个体.

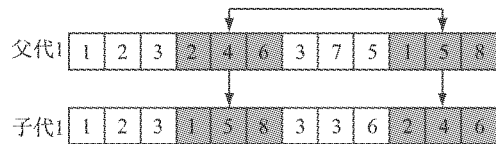


图 5 局部搜索机制

Fig.5 Local search mechanism

3.3 算法流程

**步骤 1** 设置相关参数:种群大小  $popSize$ ,各类型工件数表  $patternNum$ ,执行局部搜索的次数  $searchNum$ ,最大迭代次数  $MaxIter$ ,交叉概率  $P_c$ ,最大 AGV 数量  $AGV_{max}$ ,各类型工件生产节拍表  $MA_{proctime}$ ,AGV 各节点间行驶时间表  $AGV_{timetable}$ ,机床约束矩阵  $R$ .

**步骤 2** 取 AGV 数量  $AGV_{Num}=1$ ,机床加工任务分配方案编号  $S=1$ ,分配方案数  $S_T$  为矩阵  $R$  行数,迭代次数  $y_1=1$ .

**步骤 3** 令  $Relation=R(S, :)$ ,根据  $Relation$  中约束关系对各机床加工任务进行分配.按本文个体

编码方式,将 patternNum 中各类型工件按 Relation 中约束关系选择对应目标机床,分配给 AGVNum 台 AGV,以此生成初始种群.

**步骤 4** 计算个体的评估值  $value(j)$ .

**步骤 5** 计算个体的适应度值  $cFitness(j)$ ,采用轮赌法进行算子选择.

**步骤 6** 交叉算子:以概率  $P_c$  对随机选取的 2 个个体进行交叉,按本文修正方法对交叉后的个体进行修正.

**步骤 7** 局部搜索:采用爬山算法对每个个体进行 searchNum 次局部搜索,选出个体较优解,将适应度最高的个体作为当前个体最优解.

**步骤 8** 选出父代和子代中适应度值最高的个体作为最优个体 Best.

**步骤 9** 若  $y_1 < MaxIter$ ,则令  $y_1 = y_1 + 1$ ,转步骤 4,否则转步骤 10.

**步骤 10** 若  $y_1 = MaxIter$  且  $S < S_T$ ,则令  $y_1 = 1, S = S + 1$ ,转步骤 3,否则令  $Best_{AGVNum} = Best$ ,放入最优解集,转步骤 11.

**步骤 11** 若  $AGVNum < AGVmax$ ,则令  $y_1 = 1, S = 1, AGVNum = AGVNum + 1$ ,转步骤 3,否则终止循环,转步骤 12.

**步骤 12** 将最优解集  $\{Best_1, Best_2, \dots, Best_{AGVmax}\}$  进行解码.求得各 AGV 数量下调度时间  $T_{max}$ 、各台 AGV 调度行走路径及工件搬运次序.

## 4 实例分析

### 4.1 实例描述

本文以某柴油发动机企业混流柔性精加工岛为例,需对其进行机床加工任务分配、AGV 任务分配和路径优化.如图 6 所示,该精加工岛有一个上料区、一个下料区和一个工件加工区.工件加工区共有 8 台加工中心,每台加工中心都可以加工任一类型产品.区域被分为 4 个子区域,每个子区域有 2 台机床和 1 台机械手,机床编号分别为 1,2, ..., 8,上料区编号为 0,下料区编号为 9.工件从上料区 0 上料,运至子区域由机械手送入机床加工,加工完成后运至下料区 9 下料.4 个子区域分别负责对 4 种不同型号的工件进行加工,工件类型编号分别为 A、B、C、D.纵向相邻的 2 台机床加工同一型号工件,由一名工人负责这 2 台机床的操作.每种工件的加工数量已定,所需加工时间各不相同,每台机床的加工任务和

AGV 数量未定.各型号工件所需加工时间和加工数量如表 2 所示,机床和上、下料区任意 2 个节点之间的 AGV 行驶时间如表 3 所示,且为最小路径.

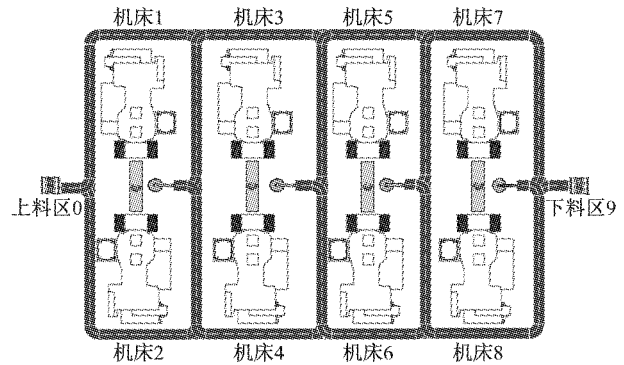


图 6 柔性精加工单元示意图

Fig.6 Diagram of flexible finishing cell

表 2 工件加工计划表

Tab.2 Machining schedule

工件类型	加工时间/s	数量/个
A	600	16
B	650	6
C	700	10
D	750	4

### 4.2 案例求解

为验证本文算法的可行性,以及与标准 Memetic 算法和遗传算法对比,在控制参数的选择上,参考文献[13]中的取值,交叉概率  $P_c = 0.6$ .由于搜索空间较大,在局部搜索参数 searchNum 的取值上,取高于文献[13]中的取值,以保证解空间的多样性,避免陷入未成熟收敛,取  $searchNum = 100$ .种群规模  $popSize = 20$ ,最大迭代次数  $MaxIter = 400$ .应企业成本控制要求,精加工岛 AGV 数量控制在 4 台之内,  $AGVmax = 4$ .取工件数表  $patternNum = 36$ ,依照企业实际生产计划,将各类型工件数表  $patternNum$ 、各类型工件生产节拍表  $MAproctime$ 、AGV 各节点间行驶时间表  $AGVtimetable$  作为约束条件按表 2 和表 3 输入参数.

考虑到该精加工岛布局和生产要求,纵向相邻 2 台机床加工同一型号工件,故机床加工任务分配方案数  $M = A_4^4 = 24$  种,机床数量  $N = 8$ ,矩阵  $R$  按式(9)约束关系输入,求解过程中各台机床加工任务按矩阵  $R$  进行分配,见式(16).如取矩阵  $R$  第 2 行  $[r_{21} \ r_{22} \ r_{23} \ r_{24} \ r_{25} \ r_{26} \ r_{27} \ r_{28}] = [A \ A \ B \ B \ D \ D \ C \ C]$ ,表示第 2 种分配方案中机床 1、2 加工工件类型 A,机床 3、4 加工工件类型 B,机床 5、6 加工工件类型 D,机床 7、8 加工工件类型 C.

表 3 各节点间 AGV 行驶时间  
Tab.3 AGV transfer time between the machines

节点	各节点间 AGV 行驶时间/s									
	机床 1	机床 2	机床 3	机床 4	机床 5	机床 6	机床 7	机床 8	上料区 0	下料区 9
机床 1	0	60	110	160	200	230	250	270	30	240
机床 2	60	0	160	110	230	200	270	250	30	240
机床 3	110	160	0	180	130	270	170	290	120	210
机床 4	160	110	180	0	270	130	290	170	120	210
机床 5	200	230	130	270	0	300	80	290	240	200
机床 6	230	200	270	130	300	0	290	80	240	200
机床 7	250	270	170	290	80	290	0	270	260	150
机床 8	270	250	290	170	290	80	270	0	260	150
上料区 0	30	30	120	120	240	240	260	260	0	280
下料区 9	240	240	210	210	200	200	150	150	280	0

$$R = \begin{pmatrix} A & A & B & B & C & C & D & D \\ A & A & B & B & D & D & C & C \\ A & A & C & C & B & B & D & D \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D & D & B & B & C & C & A & A \\ D & D & C & C & A & A & B & B \\ D & D & C & C & B & B & A & A \end{pmatrix} \quad (16)$$

根据此案例中 AGV 调度机制,针对每种机床加工任务分配方案,将 AGV 完成任务集内所有搬运任务的调度时间  $T_{max}$  作为目标函数值,即算法中个体评估值 value 越小,说明这种 AGV 调度方式越优越,该个体被选择到下一代的概率也就越大。

本案例中,AGV 的调度算法中个体评估值 value 的求解步骤如下:

**步骤 1** 判断搬运任务为上料还是下料,如为上料任务,执行步骤 2,如为下料任务,执行步骤 6。

**步骤 2** 如为上料任务,上料任务所需时间的计算公式为  $T_{ks} = T_{(K_{ks}, m_0)} + T_{(m_0, N_{im})}$ , AGV 本次搬运任务完成时间,即下次搬运任务的开始时间  $T_{ekis} = T_{ks} + T_{bkis}$ 。

**步骤 3** 判断目标机床是否空闲。如空闲,执行步骤 4;如不空闲,执行步骤 5。

**步骤 4** 如空闲,目标工件的开始加工时间为 AGV 搬运该目标工件任务的结束时间,将其加上该目标工件在机床上的加工时间即为目标工件的完工时间,  $T_{bin} = T_{ekis} + T_{im}$ 。

**步骤 5** 如不空闲,目标工件的开始加工时间为目标机床的开始空闲时间,将其加上该目标工件在机床上的加工时间即为目标工件的完工时间,  $T_{bin} = T_{bin} + T_{im}$ 。

**步骤 6** 如为下料任务,AGV 移至目标机床的时间  $T_{ks} = T_{(K_{ks}, I_{kis})}$ 。

**步骤 7** 比较  $T_{ks}$  与  $T_{ein}$  大小,判断目标工件是否加工完成。如  $T_{ks} \geq T_{ein}$ ,则完工,执行步骤 8;如  $T_{ks} < T_{ein}$ ,则未完工,执行步骤 9。

**步骤 8** 下料任务所需时间的计算公式为  $T_{ks} = T_{ks} + T_{(I_{kis}, m_1)}$ , AGV 本次搬运任务完成时间,即下次搬运任务的开始时间  $T_{ekis} = T_{ks} + T_{bkis}$ 。

**步骤 9** 下料任务所需时间的计算公式为  $T_{ks} = T_{ein} + T_{(I_{kis}, m_1)}$ , AGV 本次搬运任务完成时间,即下次搬运任务的开始时间  $T_{ekis} = T_{ks} + T_{bkis}$ 。

**步骤 10** 通过累积,可得出每台 AGV 最后一次搬运任务(下料任务)的完成时间  $T_{ks}$ 。此时,  $T_{ks}$  即为每台 AGV 完成各自所有搬运任务需要的时间,取其中的时间最大值作为该个体的 AGV 调度时间  $T_{max}$ ,即 value 值。

### 4.3 结果分析

本文选用计算机的硬件参数为 Intel i7 处理器、4GB 内存,通过 Matlab R2010b 软件进行算法运算,对 AGV 调度机制进行优化,运算过程耗时近 12 h,得到不同 AGV 数量及不同机床加工任务分配方案下目标函数值及其收敛情况。不同 AGV 数量下各机床加工任务分配方案的目标函数值波动曲线如图 8 所示。因各产品加工时间不同、数量不同,不同分配方案下 AGV 路径有较大的浮动。

图 7a 为在由 1 台 AGV 完成任务集内所有搬运任务的情况下,24 种不同机床加工任务分配方案对最终所需调度时间  $T_{max}$  的影响。图 7b~d 分别为由 2 台、3 台、4 台 AGV 完成搬运任务时,分配方案对  $T_{max}$  的影响。由图可知,当同一条波动曲线中目标函数值(即调度时间  $T_{max}$ )越小时,代表该机床加工任务分配方案越合理,调度效率越高。反之,目标函数值越大时,代表该分配方案下,完成 AGVnum 所有搬运任务所需花费的时间越长,调度效率越低。

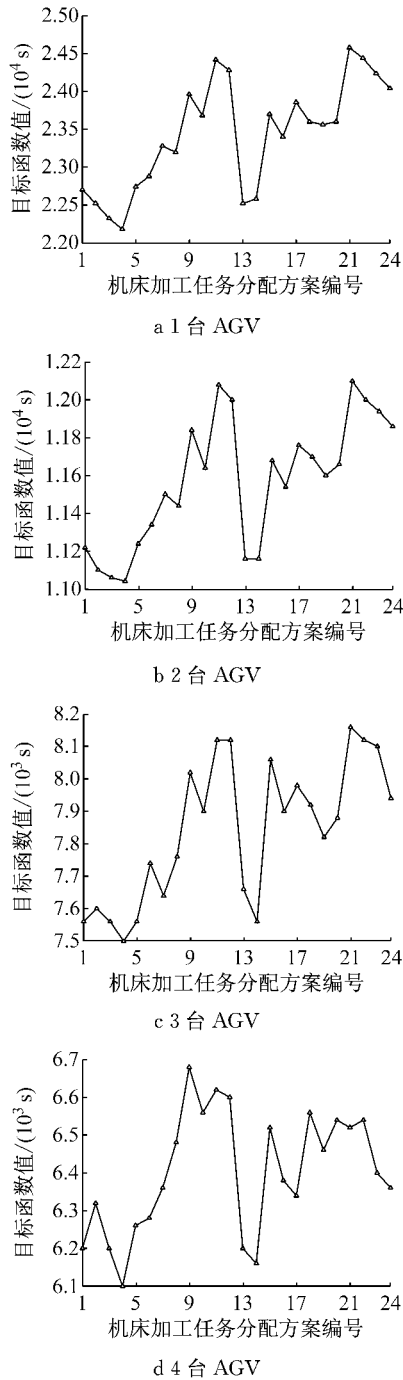


图 7 各 AGV 数量下机床分配方案波动  
Fig.7 Fluctuation of machine distribution with different schemes

取图 7 中 4 条曲线的峰值和谷值所对应的机床加工任务分配方案编号作为各 AGV 数量下机床加工任务最优和最劣方案. 由图 7a 中可以看出, 采用 1 台 AGV 进行调度时, 当采取第 4 种机床加工任务分配方案时, 即取矩阵  $R$  第 4 行  $[r_{41} \ r_{42} \ r_{43} \ r_{44} \ r_{45} \ r_{46} \ r_{47} \ r_{48}] = [A \ A \ C \ C \ D \ D \ B \ B]$  时, 目标函数值最小, AGV 调度效率最高; 当采取第 21 种分配方案时, 即取矩阵  $R$  第 21 行  $[r_{213} \ r_{214} \ r_{215} \ r_{216} \ r_{217} \ r_{218}] = [D \ D \ B \ B \ A \ A \ C \ C]$  时, 目标函数值最大, AGV 调度效率最低. 采用 2、3、4 台 AGV 进行调度时的最优、最劣方案一样可从其他 3 条曲线得出.

从图 7 中亦可发现, 为满足不同任务分配下的调度, AGV 并不能每次达到最优完工时间和路径最短的要求. 在混流生产中企业需要根据计算在选取一定数量 AGV 的机床上采用一定的分配任务.

单纯考虑 AGV 数量时, 不同 AGV 数量下机床最优、最劣分配方案的目标函数值如图 8 所示, 目标函数值随 AGV 数量变化的趋势如图 9 所示.

单纯考虑 AGV 数量时, 不同 AGV 数量下机床最优、最劣分配方案的目标函数值如图 8 所示, 目标函数值随 AGV 数量变化的趋势如图 9 所示.

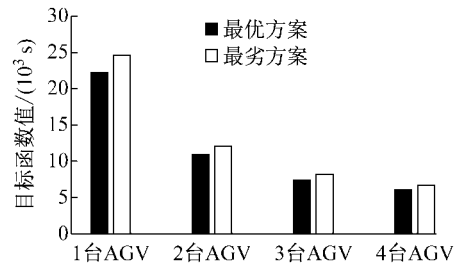


图 8 不同 AGV 数量下机床分配方案对目标函数值的影响  
Fig.8 Effect of different machine distribution schemes on objective function value

由图 8 可知, AGV 数量相等的情况下, 不同的机床加工任务分配方案会对目标函数值造成较大的影响, 最优方案相较于最劣方案可提升 AGV 调度效率近 10%. 可见, 在混流柔性加工单元 AGV 调度过程中, 根据不同类型工件的来料数量和加工时间分配各台机床的加工任务对提升 AGV 调度效率有着重要意义.

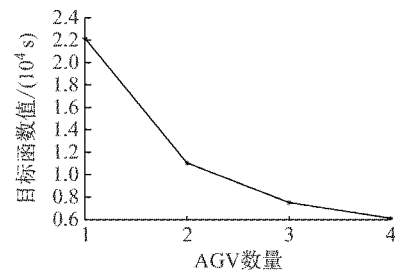


图 9 AGV 数量对目标函数值的影响  
Fig.9 Effect of AGV number on objective function value

从图 9 中可以看出, 随着 AGV 数量的增加, 目标函数值的变化幅度越来越小, 即 AGV 数量对同一任务集调度效率的增幅随 AGV 数量的增加而减小. 因此, 考虑到 AGV 的高成本, 在满足生产节拍要求的情况下可选择较少 AGV 数量, 最终根据工厂实际情况选择 3 台 AGV 进行物料运输. 由图 7 可知,

AGV 数量为 3 台时,当采取机床约束矩阵  $R$  中第 4 种分配方案 $[r_{41} \ r_{42} \ r_{43} \ r_{44} \ r_{45} \ r_{46} \ r_{47} \ r_{48}] = [A \ A \ C \ C \ D \ D \ B \ B]$ ,即机床 1、2 加工工件类型 A,机床 3、4 加工工件类型 C,机床 5、6 加工工件类型 D,机床 7、8 加工工件类型 B 时,AGV 调度时间最短,其最优解由表 4 给出。

表 4 3 台 AGV 最优调度路径  
Tab.4 Optimal routing with 3 AGVs

运输车编号	AGV 调度路径
AGV1	0→4→0→1→9→0→5→9→0→4→9→0→3→9→0→2→9→0→7→9→0→1→9→0→3→9→0→3→9→1→9→0→2→0→1→3→9
AGV2	0→7→0→1→0→4→9→2→9→0→2→9→0→8→0→4→9→3→9→0→2→9→0→2→9→0→1→9→7→9→8→9→5→9→6→9→1→9→8→9
AGV3	0→1→0→2→0→3→0→5→0→6→0→1→9→7→9→0→1→9→0→7→0→6→9→0→3→0→4→9→0→8→9→2→9→0→2→0→8→4→9→1→9→8→9

注:1~8 表示机床 1~8;0 和 9 分别表示上、下料区。

图 10 给出了 3 台 AGV 数量下,改进 Memetic 算法与标准遗传算法、标准 Memetic 算法迭代收敛情况的对比,结果表明改进算法收敛速度更快,所得最优解也相对更好。

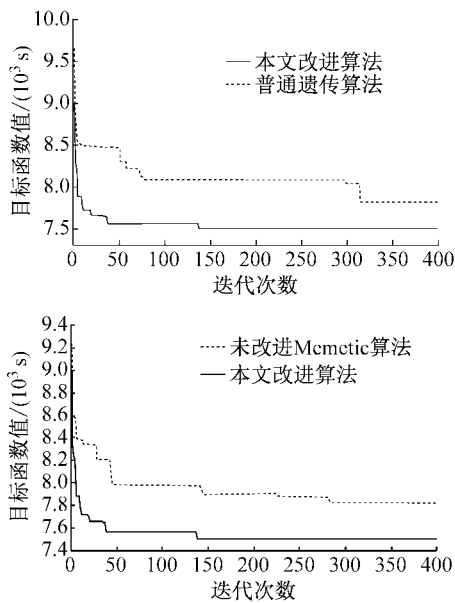


图 10 不同算法目标函数值收敛情况对比

Fig.10 Convergence contrast of objective function value among different algorithms

### 5 结语

本文研究了多品种混流生产柔性加工单元多 AGV 任务分配和调度问题. 基于柔性加工单元机床

加工任务可分配的特点,提出了在 AGV 调度优化过程中考虑机床加工任务的不同分配方案和 AGV 数量的影响,设计了改进 Memetic 算法,并以 8 台机床的柔性精加工单元为例进行详细分析和求解,验证了该方法的有效性,这对该类加工单元的调度具有较好的借鉴意义。

### 参考文献:

[1] HO Y C, LIAO T W. Zone design and control for vehicle collision prevention and load balancing in a zone control AGV system[J]. Computer and Industrial Engineering, 2009, 56 (1):417.

[2] REZAPOUR S, FARAHANI R Z, MIANDOABCHI E. A machine-to-loop assignment and layout design methodology for tandem AGV systems with single-load vehicles [J]. International Journal of Production Research, 2011, 49 (12):3605.

[3] 侯玲娟,周泓,梁春华. 不确定需求和旅行时间下的车辆路径问题[J]. 计算机集成制造系统, 2011,17(1):101. HOU Lingjuan, ZHOU Hong, LIANG Chunhua. Vehicle routing problem with uncertain demand and travel time[J]. Computer Integrated Manufacturing Systems, 2011,17(1):101.

[4] 胡彬,王冰,王春香,等. 一种基于时间窗的自动导引车动态路径规划方法[J]. 上海交通大学学报,2012,46(6):967. HU Bin, WANG Bing, WANG Chunxiang, et al. Dynamic routing of automated guided vehicles based on time window[J]. Journal of Shanghai Jiaotong University, 2012,46(6):967.

[5] 贺丽娜,楼佩煌,钱晓明,等. 基于时间窗的自动导引车无碰撞路径规划[J]. 计算机集成制造系统,2010, 16(12):2630. HE Lina, LOU Peihuang, QIAN Xiaoming, et al. Conflict-free automated guided vehicles routing based on time window[J]. Computer Integrated Manufacturing Systems, 2010, 16 (12):2630.

[6] GUAN Xianping, DAI Xianzhong, LI Jun. Revised electromagnetism-like mechanism for flow path design of unidirectional AGV systems [J]. International Journal of Production Research, 2011, 49(2):401.

[7] CARIDÁ V F, MORANDIN O, Jr., TUMA C C M. Approaches of fuzzy systems applied to an AGV dispatching system in a FMS[J]. International Journal of Advanced Manufacturing Technology, 2015, 79(1):615.

[8] FAUADI M H F, YAHAYA S H, MURATA T, et al. Intelligent combinatorial auctions of decentralized task assignment for AGV with multiple loading capacity[J]. Ieee Transactions on Electrical & Electronic Engineering, 2013, 8(4):371.

[9] ZHENG Y, XIAO Y, SEO Y. A tabu search algorithm for simultaneous machine/AGV scheduling problem [J]. International Journal of Production Research, 2014, 52 (19):5748.

(下转第 1858 页)