

doi:10.11835/j.issn.1000-582X.2020.024

基于差分进化算法的 FMS 中机器与 AGV 同时调度方法

伍 乐, 宋豫川, 吕向飞, 雷 琦

(重庆大学 机械传动国家重点实验室, 重庆 400044)

摘要:针对柔性制造系统中机器与 AGV(automated guided vehicle)同时调度问题,提出一种混合变邻域搜索的改进离散差分进化算法。以最大完工时间最小为优化目标,考虑机器与 AGV 双资源约束,建立相应的数学模型。为了同时调度机器与 AGV,采用基于工序、机器、AGV 的 3 层编码结构。通过改进差分进化(differential evolution, DE)算法的变异、交叉算子产生新个体以提高算法的全局搜索能力,并引入模拟退火算法中解的接受准则选择下一代。同时,为了增强算法的局部搜索能力,对算法每次迭代的最优个体进行变邻域搜索。通过算例计算和对比,证明了提出的改进 DE 算法的有效性、稳定性和优越性。

关键词:柔性制造系统;调度;差分进化算法;变邻域搜索

中图分类号:TP273

文献标志码:A

文章编号:1000-582X(2021)12-116-14

An improved differential evolution algorithm for simultaneous scheduling of machines and AGVs in an FMS

WU Yue, SONG Yuchuan, LYU Xiangfei, LEI Qi

(State Key Laboratory of Mechanical Transmission, Chongqing University,
Chongqing 400044, P. R. China)

Abstract: An improved discrete differential evolution algorithm with variable neighborhood search was proposed for solving simultaneous scheduling of machines and AGVs in flexible manufacturing systems. With the optimization goal of making the maximum completion time minimum, considering the dual resource constraints of machines and AGVs, the corresponding mathematical model was established. The three-layer coding structure of operation, machine and AGV was employed to schedule machines and AGVs simultaneously. In order to improve the global search capability, the differential evolution algorithm generated new individuals by improved mutation and crossover operators, and introduced the acceptance criterion of solution in simulated annealing algorithm to select next

收稿日期:2020-07-17 **网络出版日期:**2020-09-08

基金项目:国家自然科学基金资助项目(51205429);工信部“船用柴油机关重件行业数字化车间集成标准研究与试验验证”项目(CSICXX002)。

Supported by National Natural Science Foundation of China (51205429) and Ministry of Industry and Information Technology “Research and Experimental Verification on Integration Standard of Digital Workshop in Key Parts Industry of Marine Diesel Engine” (CSICXX002).

作者简介:伍乐(1996—),女,重庆大学硕士研究生,主要从事生产运作管理、智能优化算法研究,(E-mail)20141999@cqu.edu.cn。

通讯作者:宋豫川,男,重庆大学教授,博士生导师,(E-mail)syc@cqu.edu.cn。

generation. Furthermore, a variable neighborhood search was performed on the optimal individual in each iteration of the algorithm in order to enhance the local search capability. Finally, the effectiveness, stability and superiority of the improved differential evolution algorithm were proved by calculation and comparison of examples.

Keywords: flexible manufacturing systems; scheduling; differential evolution algorithm; variable neighborhood search

在现代制造业生产中,多品种小批量、多样化个性化定制的生产模式已成为发展趋势,这必将会对柔性制造系统(flexible manufacturing system, FMS)提出更高的要求。生产调度是柔性制造系统的基础,制定科学合理的调度方案,对柔性制造系统的性能具有重要影响。其中,柔性作业车间调度问题(flexible job shop scheduling problem, FJSP)是非常复杂的 NP 难题。在以往的 FJSP 研究中,为了简化数学模型解决问题,大多忽略了工件在不同机器之间的运输时间,或将其包含在工件加工时间内,这都是不合理的,特别是当工件的移动完全依赖于自动导引小车(automated guided vehicle, AGV)并且运输时间与加工时间相当时^[1]。因此,有必要对 FMS 中的机器与 AGV 同时调度问题进行研究。

针对机器与 AGV 同时调度问题,国内外学者进行了很多研究。早期, Bilge 等^[2]将问题表述为一个非线性混合整数规划模型,采用带滑动时间窗的迭代启发式方法进行求解。这种精确求解算法能够在简单小规模问题上取得全局最优解,但随着问题复杂度的增加,无法在合理时间内搜索到满意解。近年来,智能优化算法如遗传算法^[3-8]、差分进化算法^[9]、粒子群算法^[10-11]、蚁群算法^[12]、花授粉算法^[13]等,在求解具有复杂约束的同时调度问题上,取得了较好的进步。Abdelmaguid 等^[3]提出一种新的启发式编码方案,采用混合遗传算法同时调度机器与 AGV。Kumar 等^[9]提出机器选择和车辆分配两种启发式算法,并与差分进化算法相结合,为工件分配合适的机器与 AGV,但没有对算法进行改进,仅对工序操作进行了基本差分变换。Zhang 等^[10]结合主粒子和嵌套粒子的特点,设计了一种改进粒子群优化算法来求解同时调度问题。

差分进化(differential evolution, DE)算法是一种基于种群进化的智能优化算法,具有全局搜索能力强,操作简单,鲁棒性好的优点。为了增强算法性能,通常需要对算法进行改进。Pan 等^[14]提出一种求解车间调度问题的离散 DE 算法,并结合局部搜索算法以改善算法性能。杨锋英等^[15]提出一种改进 DE 算法求解 AGV 调度问题,设计了基于个体生存时间的种群更新机制以增强种群多样性,提高算法的搜索能力。吴秀丽等^[16]针对分布式柔性作业车间调度问题,为离散 DE 算法设计了新的变异和交叉方式,并引入模拟退火方法以提高算法的局部搜索能力。

从现有文献来看,DE 算法在求解复杂组合优化问题上能有较好的表现,但在机器与 AGV 同时调度问题上应用还较少。并且,大多文献把集成调度问题分解为了机器调度和 AGV 调度两个子问题,忽视了机器调度与 AGV 调度的紧密联系,以及忽略了可变工艺路线对 AGV 搬运的影响。因此,针对 FMS 中机器与 AGV 同时调度问题,在完善集成调度数学模型和改进算法性能方面有进一步研究价值。另外,当 AGV 数量大于 1 时,可能会发生冲突,对机器与 AGV 调度造成影响。但文中主要围绕零件加工尺寸较小的车间,其运输的 AGV 载重和尺寸也较小,虽然可能发生冲突,但是在车间区域内能并行通过,冲突问题产生的概率较小。

笔者针对 FMS 中机器与 AGV 同时调度问题,建立了以最大完工时间最小为优化目标的调度数学模型,提出了基于工序、机器、AGV 的 3 层编码结构,以解决机器分配和 AGV 分配。针对 DE 算法易陷入局部最优的问题,设计了一种混合变邻域搜索的改进离散差分进化算法,以每次迭代的最优解作为局部搜索的初始解,设计了 3 种不同的邻域结构进行变邻域搜索,并采用种群更新策略以增强种群多样性,进而提升算法搜索到更好解的可能性。最后,通过对比实验,验证了所提算法求解 FMS 中机器与 AGV 同时调度问题的有效性和优越性。

1 问题描述及建模

1.1 问题描述

假设柔性作业车间有 n 个待加工工件在 m 台机器上加工,若干辆相同的 AGV 负责工件在机器之间运送。每个工件有 n_i 道不同的工序,每道工序可选择的加工机器不唯一,每台 AGV 可以在任意两台机器之间运送工件。已知每个工件加工所需的工序,每道工序在对应加工机器上的加工时间,以及小车在不同机器间的运输时间,要求在满足一定约束条件下,为每道工序选择最合适的加工机器和搬运小车,确定每个工件的每道工序在各机器上的最佳加工顺序和开工、完工时间,以及在各 AGV 上的最佳搬运顺序和开始搬运、完成搬运时间,最终满足优化目标。

为了简化 FMS 中机器和 AGV 同时调度问题,假设条件如下:

- 1) 零时刻,所有机器与 AGV 都可用,机器与 AGV 连续工作。
- 2) 任意时刻,每台机器一次只能加工一个工件,每辆小车一次只能搬运一个工件。
- 3) 同工件的工序有先后约束,不同工件之间无约束。
- 4) AGV 数量提前确定。
- 5) AGV 行驶速度恒定,其负重不影响小车行驶速度。
- 6) AGV 在任意两台机器之间的运输路线是固定的,运输时间仅与机器之间的距离有关。
- 7) 每台机器前设置有缓冲区,缓冲区无限大,以避免死锁。
- 8) 所有待加工工件在开始加工前都放在装载台 M_0 ,完成所有工序加工的工件放入卸载台 M_{m+1} 。
- 9) 小车每次执行完任务后停靠在加工机器旁,不返回起点。
- 10) 小车没有装载工件的行驶过程称为空载,装载工件的运送过程称为负载。
- 11) 不考虑 AGV 可能发生冲突、故障、电量不足等问题。

1.2 数学模型

为了建立问题的模型,引入如下符号和变量:

J_i 表示工件 $i, i \in \{1, 2, \dots, n\}$; M_k 表示机器 $k, k \in \{1, 2, \dots, m\}$; R_v 表示小车 $v, v \in \{1, 2, \dots, w\}$; O_{ij} 表示工件 J_i 的第 j 道工序, $j \in \{1, 2, \dots, n_i\}$; C_i 表示工件 J_i 的完工时间; p_{ijk}^s 表示工序 O_{ij} 在机器 k 上开始加工的时间; p_{ijk} 表示工序 O_{ij} 在机器 k 上的加工时间; p_{ijk}^c 表示工序 O_{ij} 在机器 k 上完成加工的时间; T_{ij} 表示小车运送工序 O_{ij} 的搬运任务; t_{ijv}^s 表示小车 R_v 搬运任务 T_{ij} 的空载开始时间; t_{ijv}^c 表示小车 R_v 搬运任务 T_{ij} 的空载完成时间; t_{ijv}^s 表示小车 R_v 搬运任务 T_{ij} 的负载开始时间; t_{ijv}^c 表示小车 R_v 搬运任务 T_{ij} 的负载完成时间; t_{hk} 表示小车从机器 M_h 到 M_k 的运输时间; t' 表示小车的空载运输时间; t 表示小车的负载运输时间; M 表示一个无穷大的正数。

$$x_{ijk} = \begin{cases} 1, & \text{工序 } O_{ij} \text{ 在机器 } M_k \text{ 上加工。} \\ 0, & \text{其余。} \end{cases}$$

$$z_{ijv} = \begin{cases} 1, & \text{搬运任务 } T_{ij} \text{ 由 } R_v \text{ 运送。} \\ 0, & \text{其余。} \end{cases}$$

$$\alpha_{i'j'ijk} = \begin{cases} 1, & \text{工序 } O_{i'j'} \text{ 先于 } O_{ij} \text{ 在机器 } M_k \text{ 上加工。} \\ 0, & \text{其余。} \end{cases}$$

$$\beta_{T_{pq}ijv} = \begin{cases} 1, & \text{搬运任务 } T_{pq} \text{ 先于 } T_{ij} \text{ 由 } R_v \text{ 搬运。} \\ 0, & \text{其余。} \end{cases}$$

为了便于理解,变量的表示含义如图 1 所示。假设工序 O_{ij} 在机器 M_k 上加工,工件 i 的前一道工序 $O_{i(j-1)}$ 在机器 $M_{k'}$ 上加工,工序 O_{pq} 在机器 M_h 上加工。若小车 R_v 有一搬运任务 T_{ij} ,需从当前位置 M_h 空载行驶至 $M_{k'}$ 领取工件 i ,再从 $M_{k'}$ 搬运工件 i 至 M_k 进行加工。

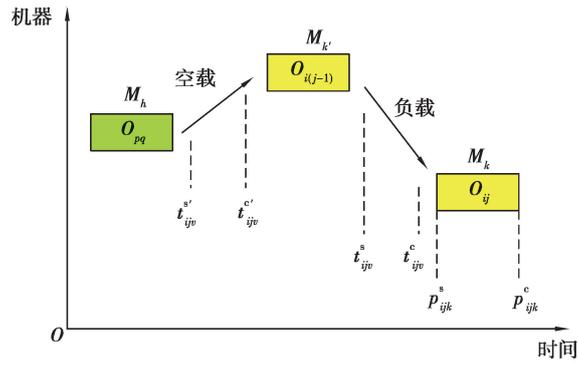


图 1 变量含义示意图

Fig. 1 Schematic diagram of variable meaning

以最大完工时间最小为优化目标,针对 FMS 中机器与 AGV 同时调度问题,建立数学模型如下。

目标函数:

$$f = \min(\max_{1 \leq i \leq n} C_i) \tag{1}$$

约束条件:

1) 每道工序一次只能分配给一台机器加工。

$$\sum_{k=1}^m x_{ijk} = 1 \tag{2}$$

2) 每道搬运任务一次只能分配给一辆小车运送。

$$\sum_{v=1}^w z_{ijv} = 1 \tag{3}$$

3) 同一辆小车,只有前一道任务 T_{pq} 搬运完后,才能开始下一道任务 T_{ij} 的搬运。

$$t_{pqv}^c \leq t_{ijv}^s + M(1 - \beta_{T_{pqijv}}) \tag{4}$$

式中, t_{pqv}^c 表示小车 R_v 搬运任务 T_{pq} 的负载完成时间, $\forall p \in \{1, 2, \dots, n\}, q \in \{1, 2, \dots, n_i\}$ 。

4) 小车从当前的机器位置 M_h 行驶至该工件前一道工序的加工机器 $M_{k'}$ 处领取工件。

$$t_{ijv}^s + \sum_{h=1}^m \sum_{k'=1}^m x_{pqh} \cdot x_{i(j-1)k'} \cdot t_{hk'} = t_{ijv}^s \tag{5}$$

5) 小车空载到达机器 $M_{k'}$ 并且工序 $O_{i(j-1)}$ 加工完成,才能开始搬运。

$$\max(t_{ijv}^s, p_{i(j-1)k'}^c) \leq t_{ijv}^s \tag{6}$$

6) 小车从机器 M_k 搬运工件到工序 O_{ij} 的加工机器 M_k 处进行加工。

$$t_{ijv}^s + \sum_{k'=1}^m \sum_{k=1}^m x_{i(j-1)k'} \cdot x_{ijk} \cdot t_{k'k} = t_{ijv}^c \tag{7}$$

7) 小车负载到达机器 M_k 并且机器空闲,才能开始加工。

$$\max(t_{ijv}^c, p_{i'j'k}^c) \leq p_{ijk}^s + M(1 - \alpha_{i'j'ijk}) \tag{8}$$

式中, $p_{i'j'k}^c$ 表示机器 M_k 前一道工序 $O_{i'j'}$ 的加工完成时间, $\forall i' \in \{1, 2, \dots, n\}, j' \in \{1, 2, \dots, n_i\}$ 。

8) 工序 O_{ij} 在机器 M_k 上加工。

$$p_{ijk}^s + \sum_{k=1}^m x_{ijk} \cdot p_{ijk} = p_{ijk}^c \tag{9}$$

2 混合变邻域搜索的改进离散差分进化算法

基本 DE 算法是基于实数编码的,主要作用于连续域的优化问题。而柔性作业车间的调度问题是离散的组合优化问题,所以需要改进 DE 算法的变异、交叉操作,以适应算法的离散搜索过程。同时,变邻域搜索(variable neighborhood search, VNS)算法能够有效避免算法陷入局部最优,提高算法的局部搜索能力。因此,提出混合变邻域搜索的改进离散差分进化(improved discrete differential evolution VNS, IDDE-VNS)算

法求解 FMS 中机器与 AGV 同时调度问题。

2.1 总体流程设计

IDDE-VNS 算法的总体流程设计如图 2 所示。

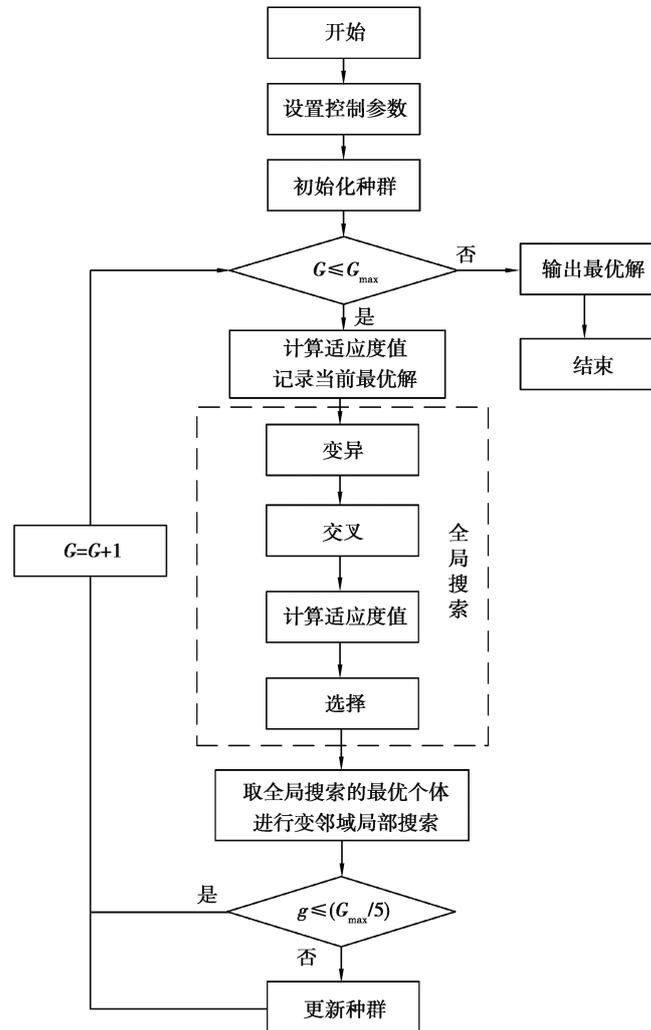


图 2 IDDE-VNS 算法流程图

Fig. 2 Flow chart of IDDE-VNS algorithm

具体步骤如下。

步骤 1: 输入调度信息, 工件加工信息和小车运输时间。设置 DE 算法控制参数, 初始种群规模 P , 最大迭代次数 G_{\max} , 变异概率 F , 交叉概率 C_r , 小车数量 N_{agv} , 初始温度 T_0 。

步骤 2: 根据编码方式和初始种群规模, 随机生成初始种群。

步骤 3: 判断终止迭代条件, 即当前迭代次数 G 是否达到最大迭代次数 G_{\max} , 若是, 输出最优解, 结束迭代; 若否, 计算种群适应度值, 记录当前最优解 x_{best} 。

步骤 4: 全局搜索。

1) 根据变异概率 F , 选择初始目标种群 X 执行变异操作得到变异种群 V 。

2) 根据交叉概率 C_r , 对目标种群 X 与变异种群 V 进行交叉操作得到试验种群 U 。

3) 重新计算试验种群 U 的适应度值。

4) 引入模拟退火思想, 比较目标种群 X 与试验种群 U , 按退火概率选择相应的个体进入下一代。

步骤 5: 局部搜索。对全局搜索得到的最优个体进行变邻域搜索, 获得新的最优个体 x'_{best} 。

步骤 6: 更新种群。判断当前最优解 x'_{best} 与前一代最优解 x_{best} 是否相等, 若相等, $g = g + 1$; 若不相等,

$g=0$ 。当 $g=G_{\max}/5$ 时,即最优解连续代 $G_{\max}/5$ 不变,则新产生一部分个体替换原种群中的较差个体^[17]。否则,跳转步骤 3,经过步骤 4、5、6 操作后的种群作为下一代目标种群 X ,继续迭代。

2.2 详细设计

2.2.1 编码和解码方法

为了表示问题的解,首先需要设计编码。算法的性能和效率很大程度上取决于染色体的编码,需要考虑染色体的可行性、完备性、健全性和非冗余性。对于 FMS 中机器与 AGV 同时调度问题,不光要安排工序的加工顺序,还要确定每道工序的加工机器和每次负责搬运的小车。因此,编码分为 3 个部分。第 1 行表示基于工序的编码,第 2 行表示基于机器的编码,第 3 行表示基于 AGV 的编码。为了便于解释编码过程,采用一个 3×3 FJSP 算例来描述,其工件加工信息如表 1 所示。2 台 AGV 在任意机器之间的运输时间如表 2 所示。

表 1 FJSP 算例的工件加工信息

Table 1 Processing information of FJSP examples

工件	工序	可选择的机器和加工时间/min		
		M_1	M_2	M_3
J_1	Q_{11}	6	—	7
	Q_{12}	5	7	5
	Q_{21}	—	7	4
J_2	Q_{22}	8	5	6
	Q_{23}	3	—	5
	Q_{24}	—	—	4
J_3	Q_{31}	8	6	—
	Q_{32}	10	—	9

表 2 各机器之间 AGV 的运输时间

Table 2 Transportation time of AGV between machines min

机器	M_0	M_1	M_2	M_3	M_4
M_0	0	2	1	1	2
M_1	1	0	2	1	2
M_2	1	2	0	2	2
M_3	1	1	2	0	1
M_4	2	2	1	2	0

根据工件加工信息,给出一组染色体编码见图 3,表示一个可行调度解。算例中有 3 个工件加工,依次对应的工序数为 2、4、2,染色体长度为 $L = \sum_{i=1}^n n_i = 8$ 。第 1 行基于工序编码的基因串中工件 1 出现 2 次,工件 2 出现 4 次,工件 3 出现 2 次,工件 i 出现的第 j 次表示工件 i 的第 j 道工序;第 2 行基于机器编码的基因串表示工件 1 的第 1 道工序由机器 1 加工,第 2 道工序由机器 3 加工,同理工件 2、工件 3 以此类推;第 3 行基于 AGV 编码的基因串表示搬运的小车号,与第 1 行工序加工顺序一一对应。因此,工序基因的第 1 个数字 2 可解读为工件 2 由小车 R_1 搬运至机器 M_2 进行工序 O_{21} 的加工操作,以此类推。

考虑 AGV 约束的柔性作业车间调度问题解码时,比传统的车间调度问题更加复杂。除了要确定每个工件的每道工序的开工时间和完工时间,还要确定 AGV 运送工件的空载、负载开始和完成时间。针对 1 个工

件的某道工序 O_{ij} 加工过程,如图 1 所示,设计解码方法如下。

1) 确定 AGV 运送工序 O_{ij} 空载、负载开始和完成时间,具体步骤如下。

步骤 1: 读取由上述编码方式生成的染色体信息,工序顺序 P 、对应加工机器 M_k 和运送小车编号 R_v 。

步骤 2: 记录 AGV 当前位置及其走过的路径。

步骤 3: 判断本道工序 O_{ij} 是否为工件 i 的第 1 道工序。若是,跳转步骤 4; 若否,跳转步骤 6。

步骤 4: 判断 AGV 的当前位置是否在装载台 M_0 。

① 若是,AGV 空载运输时间 $t' = 0$ 。

② 若不是,AGV 需要从当前位置 M_h 行驶到装载台 M_0 领取工件,AGV 空载运输时间 $t' = M_h \rightarrow M_0$ 。

步骤 5: AGV 从装载台 M_0 运送工件行驶到本道工序的加工机器 M_k 处,AGV 负载运输时间 $t = M_0 \rightarrow M_k$ 。

步骤 6: 判断 AGV 的当前位置是否在紧前工序加工机器 $M_{k'}$ 处。

① 若是,AGV 空载运输时间 $t' = 0$ 。

② 若不是,AGV 需要从当前位置 M_h 行驶至该工件的前一道工序的加工机器 $M_{k'}$ 领取工件。若该工件正在机器 $M_{k'}$ 上加工,需等待加工完成后再搬运。AGV 空载运输时间 $t' = M_h \rightarrow M_{k'}$ 。

步骤 7: AGV 从紧前工序加工机器 $M_{k'}$ 运送工件行驶到本道工序的加工机器 M_k 处,AGV 负载运输时间 $t = M_{k'} \rightarrow M_k$ 。

步骤 8: 输出 AGV 空载开始和完成时间,负载开始和完成时间。

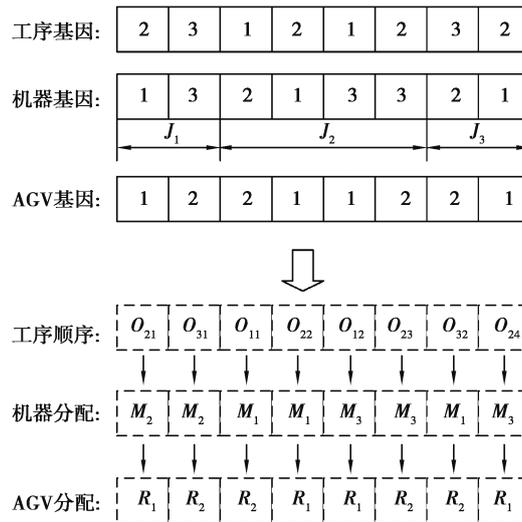


图 3 基于工序、机器、AGV 的 3 层编码结构

Fig. 3 The three-layer coding structure of operation, machine and AGV

2) 确定工序 O_{ij} 的开工时间和完工时间。

$$p_{ijk}^s = \max\{p_{ij'k}^c, (p_{i(j-1)k'}^c + t), (t_{ijv}^s + t' + t)\}, \tag{10}$$

$$p_{ijk}^c = p_{ijk}^s + p_{ijk}. \tag{11}$$

根据设计的编码和解码方法,结合工件的加工信息和 2 台 AGV 的运输信息,随机产生一个可行调度方案,其甘特图如图 4 所示。分析甘特图可知,工件加工过程和小车的运输过程具体如下,其中“* →”表示 AGV 空载运行。

R_1 的运输过程: $M_0 * \rightarrow M_2$ (取工件 2) $\rightarrow M_3$ (加工 O_{23} 并取得工件 1) $\rightarrow M_1$ (等待加工 O_{12}) $* \rightarrow M_3$ (取工件 2) $\rightarrow M_4$ (工件 2 完成加工后运到卸载台)。

R_2 的运输过程: M_0 (取工件 2) $\rightarrow M_3$ (加工 O_{21}) $* \rightarrow M_0$ (取工件 3) $\rightarrow M_2$ (加工 O_{31}) $* \rightarrow M_0$ (取工件 1) $\rightarrow M_3$ (加工 O_{11} 并取得工件 2) $\rightarrow M_2$ (等待加工 O_{22} 并取得工件 3) $\rightarrow M_1$ (加工 O_{32}) $\rightarrow M_4$ (工件 3 完成加工后运到卸载台) $* \rightarrow M_3$ (加工 O_{24}) $* \rightarrow M_1$ (取工件 1) $\rightarrow M_4$ (工件 1 完成加工后运到卸载台)。

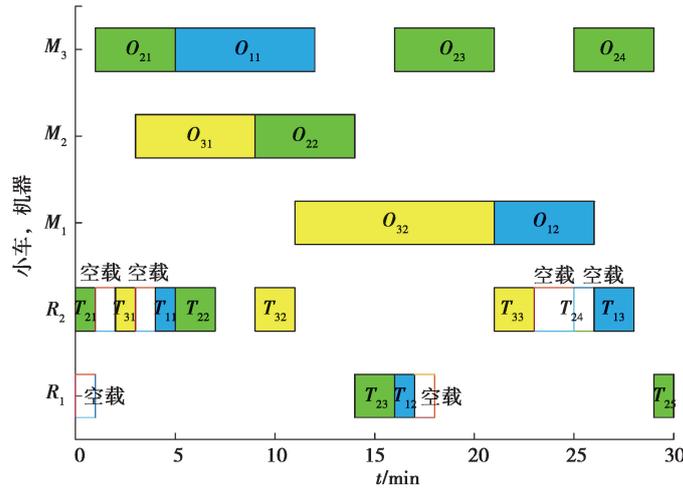


图 4 调度方案的甘特图

Fig. 4 Gantt chart of scheduling scheme

2.2.2 全局搜索

根据上述编码方式,为了使个体直接在离散域搜索,以提高算法搜索速度,对 DE 算法的操作算子进行了改进。

1) 变异操作。从目标种群 X 中随机选择 3 个不同的个体 $X_{r_1}^t, X_{r_2}^t$ 和 $X_{r_3}^t$ 进行变异操作生成变异个体 V_i^t 。

$$V_i^t = F \otimes G(F \otimes G(X_{r_2}^t, X_{r_3}^t), X_{r_1}^t), \tag{12}$$

式中:目标种群 $X^t = \{X_1^t, X_2^t, \dots, X_P^t\}$, 变异种群 $V^t = \{V_1^t, V_2^t, \dots, V_P^t\}$, $r_1, r_2, r_3, i \in [1, 2, \dots, P], t$ 为当前运行代数, F 为变异概率。

式(12)由 2 个部分构成。首先,由目标个体 $X_{r_2}^t$ 和 $X_{r_3}^t$ 采用如下操作生成差分矢量 Δ_i^t 。生成一个随机数 $r \in [0, 1]$,若 r 小于变异概率,差分矢量 $\Delta_i^t = G(X_{r_2}^t, X_{r_3}^t)$;否则, $\Delta_i^t = X_{r_2}^t$ 。

$$\Delta_i^t = F \otimes G(X_{r_2}^t, X_{r_3}^t) = \begin{cases} G(X_{r_2}^t, X_{r_3}^t), & r < F. \\ X_{r_2}^t, & \text{其他。} \end{cases} \tag{13}$$

然后,第二部分由差分矢量 Δ_i^t 和目标个体 $X_{r_1}^t$ 采用如下操作生成变异个体 V_i^t ,操作同上。

$$V_i^t = F \otimes G(\Delta_i^t, X_{r_1}^t) = \begin{cases} G(\Delta_i^t, X_{r_1}^t), & r < F. \\ \Delta_i^t, & \text{其他。} \end{cases} \tag{14}$$

其中, $G(X_{r_2}^t, X_{r_3}^t)$ 和 $G(\Delta_i^t, X_{r_1}^t)$ 表示对染色体进行变换操作,生成可能更优的个体。由于染色体编码分为 3 个部分,需要对 3 行基因分别进行操作。

对基于工序编码的基因串,采用 IPOX 交叉^[18]。其具体交叉过程为:将工件集 $\{1, 2, \dots, n\}$ 随机分为非空互补的两个子集 S_1 和 S_2 ,复制父代染色体 P_1 中包含 S_1 的工件号到子代染色体 C_1 ,复制 P_1 中包含 S_2 的工件号到 C_2 ,保留它们的位置;同样复制 P_2 中包含 S_1 的工件号到 C_1 ,复制 P_1 中包含 S_1 的工件号到 C_2 ,也保留它们的位置,如图 5 所示。

对基于机器编码的基因串,采用 MPX 交叉^[18]。其具体交叉过程为:首先随机产生一行长度与父代染色体相等,由 0 和 1 组成的基因串。将父代染色体 P_1 中 1 对应位置的机器号复制到子代染色体 C_2 ,父代染色体 P_2 中 1 对应位置的机器号复制到 C_1 , P_1 中剩余的机器号保留到子代 C_1 , P_2 中剩余的机器号保留到子代 C_2 ,如图 6 所示。

对基于 AGV 编码的基因串,采用简单的两点交叉。

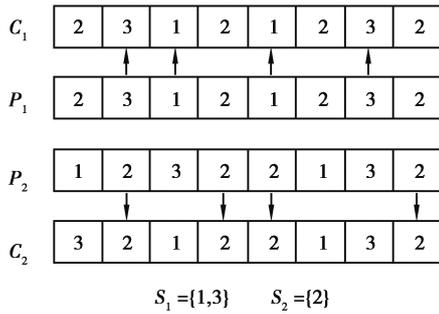


图 5 IPOX 交叉过程图

Fig. 5 Crossover process of IPOX

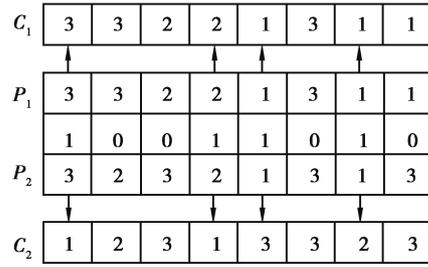


图 6 MPX 交叉过程

Fig. 6 Crossover process of MPX

2) 交叉操作。选择变异个体 V_i^t 和目标个体 X_i^t 进行交叉操作生成试验个体 U_i^t , 操作过程同上。

$$U_i^t = C_r \otimes G(V_i^t, X_i^t) = \begin{cases} G(V_i^t, X_i^t), & r < C_r, \\ V_i^t, & \text{其他。} \end{cases} \quad (15)$$

式中: 试验种群 $U^t = \{U_1^t, U_2^t, \dots, U_P^t\}; i \in [1, 2, \dots, P]; C_r$ 为交叉概率。

3) 选择操作。基本 DE 算法的选择操作采用“一对一”贪婪思想, 比较试验个体 U_i^t 和目标个体 X_i^t 的适应度值, 选择适应度值更高的个体进入下一代。但这种选择方式可能会淘汰与最优解距离很近的较差个体, 导致算法的收敛速度和搜索质量下降。研究中引入模拟退火算法中解的接受准则, 即按一定概率接受较差个体, 且概率受温度变化控制, 随着温度的降低, 接受较差个体的概率变小。

选择操作如下: 当 $f(U_i^t) < f(X_i^t)$ 时, 选择目标函数值更小的个体 U_i^t 进入下一代; 否则按概率 $r < \exp\left(-\frac{f(U_i^t) - f(X_i^t)}{T}\right)$ 接受较差的个体。

$$X_i^{t+1} = \begin{cases} U_i^t, & f(U_i^t) < f(X_i^t)。 \\ U_i^t, & f(U_i^t) \geq f(X_i^t) \text{ 且 } r < \exp\left(-\frac{f(U_i^t) - f(X_i^t)}{T}\right)。 \\ X_i^t, & \text{其他。} \end{cases} \quad (16)$$

式中, T 为退火温度, 为了提高算法搜索效率, 采用自适应变温方法, 不进行循环。 T 的变化见式(17)。

$$T = T \times \left(1 - \frac{G}{G_{\max} + 1}\right) \quad (17)$$

2.2.3 局部搜索

变邻域搜索算法比单一的邻域搜索算法性能更强, 它通过改变算法的邻域结构来扩大其搜索范围, 从而找到更优解。文中对当前种群中的最优个体执行变邻域搜索, 以提高 DE 算法的局部搜索能力。

设计 VNS 算法首先需要构造邻域结构, 算法采用了以下 3 种邻域结构来产生局部最优解:

1) 邻域结构 N_1 。在基于工序编码的染色体上, 随机选择 2 个基因位置, 所选的 2 个位置对应 2 道不同工件的工序, 互换这 2 个位置。

2) 邻域结构 N_2 。在基于机器编码的染色体上, 随机选择 1 个基因位置, 所选位置对应工序的可加工机器不唯一, 重新从该工序的可选机器集合中选择不同的机器替换该位置的机器。

3) 邻域结构 N_3 。在基于 AGV 编码的染色体上, 随机选择 1 个基因位置, 从可选 AGV 集合中选择不同的 AGV 编号替换该位置的 AGV 编号。

基于上述 3 种邻域结构, 变邻域搜索算法的操作步骤如下。

步骤 1: 算法初始化, 将全局搜索得到的最优解设为初始解 X , 设置 $q \leftarrow 1, q_{\max} = 3, c_{\max} = 40$ 。

步骤 2: 判断循环终止条件 $q > q_{\max}$ 是否满足, 若满足, 输出最优个体; 否则, 转到步骤 3。

步骤 3: 扰动环节。初始解 X 通过邻域结构 N_q 搜索得到 1 个扰动解。

步骤 4: 局部搜索。

①以扰动后的新解作为局部搜索的初始解 X' , 设置 $c \leftarrow 1$ 。

②判断循环终止条件 $c > c_{\max}$ 是否满足,若满足,输出局部最优解 X' ;否则转到步骤③。

③使当前解 X' 在 N_q 中搜索得到一个新解 X'' ,若满足 $f(X'') < f(X')$,令 $X' \leftarrow X''$ 。

④设置 $c \leftarrow c + 1$,返回步骤②循环。

步骤 5:判断邻域移动条件 $f(X') < f(X)$ 是否满足,若满足,令 $X \leftarrow X'$,并继续在邻域 N_q 内搜索;否则,设置 $q \leftarrow q + 1$,返回步骤 2 循环。

3 实验分析

实验测试以 MATLAB2016b 为开发环境,运行环境为 Intel Core i5-6200 CPU,2.40 GHz 主频,4 GB 内存。

为了验证 IDDE-VNS 算法的性能,展开以下测试实验:

1)算法对比试验。针对遗传算法、差分进化算法和混合变邻域搜索的改进差分进化算法,对比这 3 种算法产生的解的质量,为客观评价提出的改进方法提供依据。

2)非柔性算例实验。选取 Bilge 等^[2]提出的 82 个算例中的一部分进行实验,与其他文献使用各种优化算法得到的计算结果进行对比,为客观评价 IDDE-VNS 算法的性能提供依据。

3)柔性算例实验。选取 Zhang 等^[10]提出的柔性算例进行计算,并与 Zhang 等^[10]得到的优化结果对比,为客观评价 IDDE-VNS 算法作用在柔性作业车间时的性能提供依据。

3.1 算法对比实验

为了验证提出的改进方法的有效性和可行性,采用 Bilge 等^[2]提出的测试算例中的 EX11 算例进行实验。将遗传算法、差分进化算法和混合变邻域的改进差分进化算法分别命名为 GA、DE、IDDE-VNS,重复运行 20 次,对比 3 种算法的迭代曲线和优化结果。

3 种算法的种群规模、迭代次数和小车数量都设为: $P = 80, G_{\max} = 200, N_{\text{agv}} = 2$;GA 的交叉概率和变异概率设为: $P_c = 0.8, P_m = 0.4$;DE 的变异概率和交叉概率设为: $F = 0.7, C_r = 0.5$;IDDE-VNS 的变异概率、交叉概率和初始退火温度设为: $F = 0.7, C_r = 0.5, T_0 = 20$ 。

3 种算法的搜索过程曲线如图 7 所示,红色实线为 GA,蓝色虚线为 DE,黄色点划线为 IDDE-VNS。由收敛曲线图可知,GA 能快速收敛到一个较优解,但易陷入局部最优;DE 在进化前期较 GA 收敛得慢,但在搜索后期发现更优解的可能性更大,使搜索最终得到更好的解;改进后的 IDDE-VNS 的收敛速度在 GA 和 DE 之间,在进化后期,由于引入了变邻域搜索算法增强了 DE 算法的局部搜索能力,使其最终搜索到的解的质量在 3 种算法中最好。

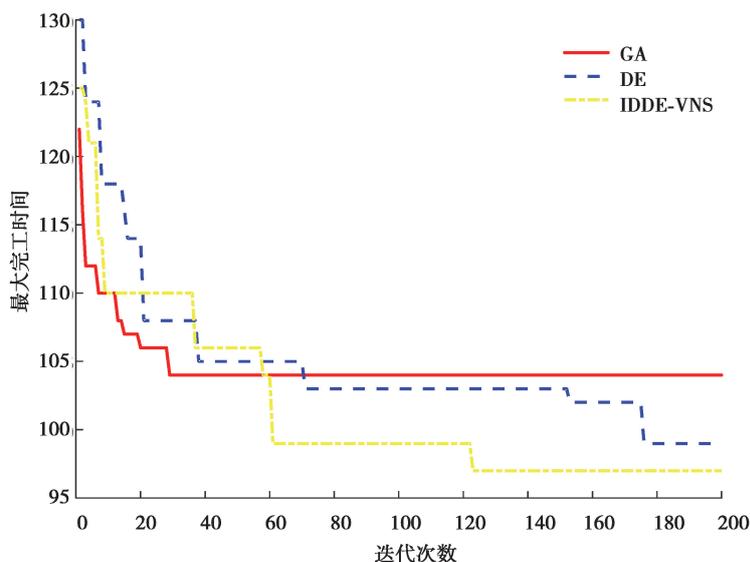


图 7 3 种算法收敛曲线对比

Fig. 7 Comparison of convergence curves of three algorithms

3 种算法计算 EX11 算例的优化结果如表 3 所示,采用 GA 进行实验 20 次,得到的最优解的平均值为 100.5,搜索到全局最优解 96 的概率为 5%;采用 DE 进行实验 20 次,得到的最优解的平均值为 98.5,搜索到全局最优解 96 的概率为 10%;采用 IDDE-VNS 进行实验 20 次,得到的最优解的平均值为 97.1,搜索到全局最优解 96 的概率为 35%。由此可得,3 种算法搜索到的解的质量和稳定性的关系为 IDDE-VNS > DE > GA。

表 3 3 种算法求解 EX11 算例实验结果

Table 3 Experimental results of three algorithms for EX11 example

次数	最优解		
	GA	DE	IDDE-VNS
1	99	98	98
2	99	98	97
3	104	100	96
4	101	100	97
5	97	96	96
6	99	97	97
7	100	97	97
8	99	97	97
9	100	97	96
10	98	101	97
11	104	101	96
12	99	97	96
13	104	97	97
14	101	100	96
15	101	99	96
16	101	101	97
17	104	100	99
18	102	100	99
19	96	96	98
20	102	97	100

说明:加粗的 96 为截至目前研究文献中 EX11 算例的最优结果。

综上所述, IDDE-VNS 比 GA、DE 的稳定性和寻优能力更好,证明了文中提出的改进方法是有效可行的。

3.2 非柔性算例实验

Bilge 等^[2]提出了 82 个算例用来测试算法的有效性,由装载台、卸载台、4 台加工机器和 2 台 AGV 构成的 4 种不同布局和 10 个不同的工件集组成。EX 后面的数字中第一个表示工件集编号,第二个表示布局编号,如“EX32”表示工件集 3-布局 2。为了验证 IDDE-VNS 相比其他算法的优越性,选取其中一部分算例进行了测试。IDDE-VNS 算法的控制参数设置为:种群规模 $P = 80$,最大迭代次数 $G_{\max} = 500$,变异概率 $F = 0.7$,交叉概率 $C_r = 0.5$,初始退火温度 $T_0 = 20$,小车数量 $N_{\text{agv}} = 2$ 。

每个算例独立计算 20 次取最优值,与其他文献得到的最优解进行对比,如表 4 所示。其中 STW 表示文献[2]采用滑动时间窗求得的最优解,AGA 表示文献[3]采用遗传算法混合启发式方法求得的最优解,FDE 表示文献[9]采用差分进化算法求得的最优解,FMAS 表示文献[19]采用基于 MAS 方法求得的最优解。

由表 4 可知, IDDE-VNS 算法计算测试算例时, 其中 100% 的解优于或等于 STW 和 AGA 的实验结果; 85% 的解优于或等于 FDE 的实验结果; 90% 的解优于或等于 FMAS 的实验结果。特别地, IDDE-VNS 算法计算 EX71 和 EX92 的解在对比实验中结果最好。由此可看出, IDDE-VNS 在计算机器与 AGV 同时调度问题时表现优秀, 与其他优秀算法相比有较高的搜索质量, 证明了 IDDE-VNS 算法的优越性。

表 4 非柔性算例实验结果

Table 4 Experimental results of non-flexible examples

算例	最优解				
	STW	AGA	FDE	FMAS	IDDE-VNS
EX11	96	96	96	111	96
EX21	105	102	100	128	100
EX31	105	99	99	114	99
EX41	118	112	112	163	112
EX51	89	87	87	97	87
EX61	120	118	115	138	118
EX71	119	115	112	124	111
EX81	161	161	161	191	161
EX91	120	118	114	138	116
EX101	153	147	147	170	147
EX12	82	82	82	87	82
EX22	80	76	76	88	76
EX32	88	85	85	99	85
EX42	93	88	85	99	87
EX52	69	69	69	78	69
EX62	100	98	98	94	98
EX72	90	79	79	78	79
EX82	151	151	153	149	151
EX92	104	104	104	107	102
EX102	144	136	135	154	135

3.3 柔性算例实验

上述算例中工件的每道工序可选择的加工机器是唯一的, 没有体现生产系统的柔性。在 FMS 中, 不同加工机器的选择会影响 AGV 路线的选择, 从而影响 AGV 的空载时间和负载时间以及工序的开工时间和完工时间。为了更好体现 IDDE-VNS 算法在 FMS 中的作用, 借鉴 Zhang 等^[10]给出的柔性作业车间集成调度算例, 由 6 个工件、6 台机器和 4 台 AGV 构成。

IDDE-VNS 算法的控制参数设置与非柔性算例实验一致, 其中最大迭代次数与文献^[10]一致, 小车数量改为 $N_{agv} = 4$, 运行 10 次取最优值, 最优实验结果的调度甘特图见图 8, 最大完工时间为 75.6 min。对比 Zhang 等^[10]采用改进粒子群优化算法得到的最大完工时间 85 min, IDDE-VNS 算法搜索到的解的质量更好, 证明了 IDDE-VNS 算法求解 FMS 中机器与 AGV 同时调度问题的优越性。

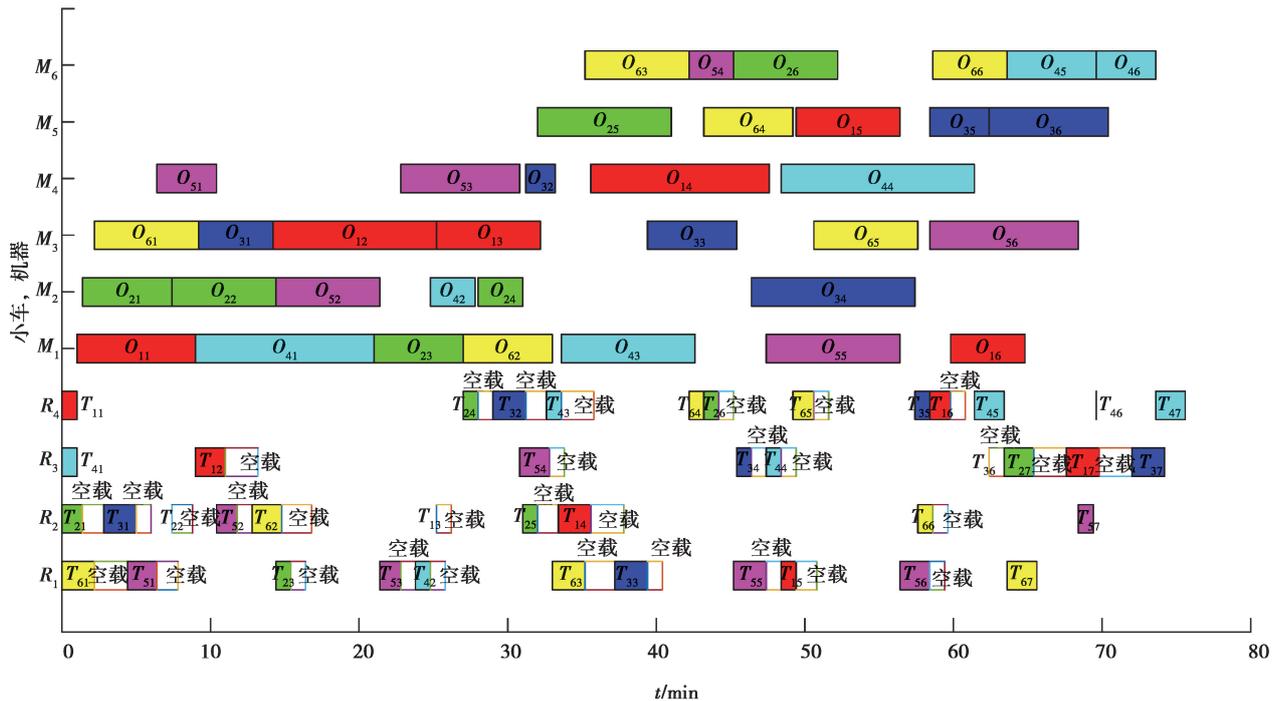


图 8 柔性算例调度方案的甘特图

Fig. 8 Gantt chart of flexible example scheduling scheme

4 结 论

针对 FMS 中机器与 AGV 同时调度问题,提出了 IDDE-VNS 算法。

1) 基于工序、机器、AGV 的 3 层编码方案,提出了改进离散差分进化算法,使个体直接在离散域搜索,提高了算法的全局搜索能力和搜索速度。

2) 将变邻域搜索算法嵌入改进差分进化算法中,并按策略对种群进行扰动操作,增加了种群多样性,提高了算法跳出局部最优的能力,从而获得质量更好的解。

3) 实验结果和比较证明了 IDDE-VNS 算法求解 FMS 中机器与 AGV 同时调度问题的有效性,稳定性和优越性。

4) 在文中研究问题基础上,结合实际生产情况,下一步将考虑更多影响因素和约束条件,如小车冲突问题,或多目标问题。

参考文献:

- [1] 付建林, 张恒志, 张剑, 等. 自动导引车调度优化研究综述[J]. 系统仿真学报, 2020, 32(9): 1664-1675.
Fu J L, Zhang H Z, Zhang J, et al. Review on AGV scheduling optimization[J]. Journal of System Simulation, 2020, 32(9): 1664-1675. (in Chinese)
- [2] Bilge Ü, Ulusoy G. A time window approach to simultaneous scheduling of machines and material handling system in an FMS[J]. Operations Research, 1995, 43(6): 1058-1070.
- [3] Abdelmaguid T F, Nassef A O, Kamal B A, et al. A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles[J]. International Journal of Production Research, 2004, 42(2): 267-281.
- [4] 龙传泽, 杨煜俊. 基于遗传算法的柔性机器人制造单元调度问题研究[J]. 组合机床与自动化加工技术, 2015(11): 141-144, 148.
Long C Z, Yang Y J. Research of flexible robotic manufacturing cell scheduling problem based on hybrid genetic algorithm[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2015(11): 141-144, 148. (in Chinese)

- [5] Umar U A, Ariffin M K A, Ismail N, et al. Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment[J]. The International Journal of Advanced Manufacturing Technology, 2015, 81(9/10/11/12): 2123-2141.
- [6] Nouri H E, BelkahlDriss O, Ghédira K. Simultaneous scheduling of machines and transport robots in flexible job shop environment using hybrid metaheuristics based on clustered holonicmultiagent model[J]. Computers & Industrial Engineering, 2016, 102: 488-501.
- [7] Lin J T, Chiu C C, Chang Y H. Simulation-based optimization approach for simultaneous scheduling of vehicles and machines with processing time uncertainty in FMS[J]. Flexible Services and Manufacturing Journal, 2019, 31(1): 104-141.
- [8] Lyu X F, Song Y C, He C Z, et al. Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems[J]. IEEE Access, 2019, 7: 74909-74924.
- [9] Kumar M V S, Janardhana R, Rao C S P. Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing[J]. The International Journal of Advanced Manufacturing Technology, 2011, 53(1/2/3/4): 339-351.
- [10] Zhang F Q, Li J J. An improved particle swarm optimization algorithm for integrated scheduling model in AGV-served manufacturing systems[J]. Journal of Advanced Manufacturing Systems, 2018, 17(3): 375-390.
- [11] 徐云琴, 叶春明, 曹磊. 含有 AGV 的柔性车间调度优化研究[J]. 计算机应用研究, 2018, 35(11): 3271-3275.
Xu Y Q, Ye C M, Cao L. Research on flexible Job-Shop scheduling problem with AGV constraints[J]. Application Research of Computers, 2018, 35(11): 3271-3275.(in Chinese)
- [12] Saidi-Mehrabad M, Dehnavi-Arani S, Evazabadian F, et al. An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs[J]. Computers & Industrial Engineering, 2015, 86: 2-13.
- [13] 刘二辉, 姚锡凡, 陶韬, 等. 基于改进花授粉算法的共融 AGV 作业车间调度[J]. 计算机集成制造系统, 2019, 25(9): 2219-2236.
Liu E H, Yao X F, Tao T, et al. Improved flower pollination algorithm for job shop scheduling problems integrated with AGVs[J]. Computer Integrated Manufacturing Systems, 2019, 25(9): 2219-2236.(in Chinese)
- [14] Pan Q K, Wang L, Qian B. A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems[J]. Computers & Operations Research, 2009, 36(8): 2498-2511.
- [15] 杨锋英, 刘会超. AGV 作业调度模型及改进的 DE 算法研究[J]. 计算机工程与应用, 2014, 50(9): 225-230.
Yang F Y, Liu H C. Research on AGV job scheduling model and improved differential evolution algorithm[J]. Computer Engineering and Applications, 2014, 50(9): 225-230.(in Chinese)
- [16] 吴秀丽, 刘夏晶. 差分进化算法求解分布式柔性作业车间调度问题[J]. 计算机集成制造系统, 2019, 25(10): 2539-2558.
Wu X L, Liu X J. Differential evolution algorithm for solving distributed flexible job shop scheduling problem[J]. Computer Integrated Manufacturing Systems, 2019, 25(10): 2539-2558.(in Chinese)
- [17] 崔琪, 吴秀丽, 余建军. 变邻域改进遗传算法求解混合流水车间调度问题[J]. 计算机集成制造系统, 2017, 23(9): 1917-1927.
Cui Q, Wu X L, Yu J J. Improved genetic algorithm variable neighborhood search for solving hybrid flow shop scheduling problem[J]. Computer Integrated Manufacturing Systems, 2017, 23(9): 1917-1927.(in Chinese)
- [18] 张超勇, 董星, 王晓娟, 等. 基于改进非支配排序遗传算法的多目标柔性作业车间调度[J]. 机械工程学报, 2010, 46(11): 156-164.
Zhang C Y, Dong X, Wang X J, et al. Improved NSGA-II for the multi-objective flexible job-shop scheduling problem[J]. Journal of Mechanical Engineering, 2010, 46(11): 156-164.(in Chinese)
- [19] Sahin C, Demirtas M, Erol R, et al. A multi-agent based approach to dynamic scheduling with flexible processing capabilities[J]. Journal of Intelligent Manufacturing, 2017, 28(8): 1827-1845.