

1-4-2019

Collaborative Optimal Scheduling Method for Production and Energy Consumption in Discrete Manufacturing Process

Wenjie Chen

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Wang Yan

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Collaborative Optimal Scheduling Method for Production and Energy Consumption in Discrete Manufacturing Process

Abstract

Abstract: The collaborative optimization of manufacturing process and energy consumption is one of the hot research issues in intelligent optimization manufacturing. To solve the scheduling optimization problem of discrete manufacturing process, a coordinated scheduling optimization of production and energy consumption with the shortest processing time and lowest energy consumption is established. *The model proposes an improved differential evolution algorithm based on adaptive mutation and crossover probability factor to solve the optimal scheduling problem.* By establishing the operation-based coding method, the application of continuous algorithm in discrete optimization scheduling problem is realized by using ascending ordering rules. The effectiveness of the proposed algorithm is verified by simulation, and the performances of particle swarm optimization algorithm, genetic algorithm and the improved algorithm are compared. The results show that the target solution obtained by this algorithm is significantly better than the other two algorithms, which verifies the superiority of the algorithm.

Keywords

collaborative optimization scheduling, differential evolution algorithm, self-adaptive operator, discrete manufacturing process

Recommended Citation

Chen Wenjie, Wang Yan. Collaborative Optimal Scheduling Method for Production and Energy Consumption in Discrete Manufacturing Process[J]. Journal of System Simulation, 2018, 30(11): 4367-4375.

离散制造过程生产协同能耗优化调度方法

陈文杰, 王艳

(江南大学物联网技术应用教育部工程研究中心, 江苏 无锡 214122)

摘要: 制造过程生产与能耗的协同优化是目前智能优化制造的热点研究问题之一, 为解决离散制造过程调度优化问题, 建立了以加工时间最短和能耗最低为目标的生产与能耗协同优化调度模型; 提出了一种基于自适应变异、交叉概率因子的改进差分进化算法, 求解优化调度问题; 通过建立基于操作的编码方式, 采用升序排序规则实现了连续算法在离散优化调度问题中的应用; 通过仿真验证了该算法的有效性, 并对比了粒子群算法、遗传算法与改进算法的性能。结果表明通过该算法得到的目标解明显优于另外两种算法, 验证了算法的优越性。

关键词: 协同优化调度; 差分进化算法; 自适应算子; 离散制造过程

中图分类号: TP278

文献标识码: A

文章编号: 1004-731X (2018) 11-4367-09

DOI: 10.16182/j.issn1004731x.joss.201811038

Collaborative Optimal Scheduling Method for Production and Energy Consumption in Discrete Manufacturing Process

Chen Wenjie, Wang Yan

(Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: The collaborative optimization of manufacturing process and energy consumption is one of the hot research issues in intelligent optimization manufacturing. To solve the scheduling optimization problem of discrete manufacturing process, a coordinated scheduling optimization of production and energy consumption with the shortest processing time and lowest energy consumption is established. The model proposes an improved differential evolution algorithm based on adaptive mutation and crossover probability factor to solve the optimal scheduling problem. By establishing the operation-based coding method, the application of continuous algorithm in discrete optimization scheduling problem is realized by using ascending ordering rules. The effectiveness of the proposed algorithm is verified by simulation, and the performances of particle swarm optimization algorithm, genetic algorithm and the improved algorithm are compared. The results show that the target solution obtained by this algorithm is significantly better than the other two algorithms, which verifies the superiority of the algorithm.

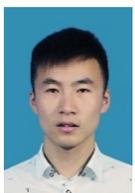
Keywords: collaborative optimization scheduling; differential evolution algorithm; self-adaptive operator; discrete manufacturing process

引言

随着我国制造业规模的不断扩大, 能源消耗量

也急剧增加。作为制造业主体的离散制造系统生产过程中耗能量十分巨大, 因此, 研究离散制造过程中生产与能耗优化控制, 对社会和经济的健康发展具有十分重要的意义。

离散制造系统生产过程复杂, 总体表现出目标多样性以及加工过程多约束性等特点^[1]。当前离散制造过程能耗优化控制主要是对生产过程的优化



收稿日期: 2018-05-21 修回日期: 2018-07-01;
基金项目: 国家自然科学基金(61572238), 江苏省杰出青年基金(BK20160001);
作者简介: 陈文杰(1992-), 男, 安徽阜阳, 硕士生, 研究方向为离散制造系统能耗优化; 王艳(1978-), 女, 江苏盐城, 教授, 博导, 研究方向为制造系统能效优化。

<http://www.china-simulation.com>

• 4367 •

调度,对生产过程中设备和零件的加工顺序进行合理的安排,从而实现加工过程的节能降耗,进而缩短设备待机时间,有效地提高了设备的生产效率,该方法对于提高企业生产效率、降低生产成本具有很高的研究意义。

离散制造过程中的能耗优化调度问题一直是制造业研究的热点问题,近年来针对上述问题的研究,众多专家学者提出了许多求解算法。如张静等^[2]提出一种基于 Pareto 支配的混合粒子群算法求解多目标车间调度问题,结合模拟退火技术的多目标局部搜索策略,实现了算法的全局搜索和局部搜索的平衡。Zhen 等^[3]通过对离散制造车间生产过程、性能指标以及约束条件进行分析,建立了基于遗传算法的调度优化模型,并选择基于操作的编码方式应用于遗传算法求解过程,提高了求解效率和精度。以上方法虽然在求解离散制造车间调度优化问题取得了一定的成果,但仍存在求解过程复杂、求解精度低等一系列问题。

针对离散制造过程车间调度问题中存在的难题,提出了一种基于差分进化(Differential Evolution, DE)算法的能耗优化控制方法。在满足加工设备能耗和相关资源约束的前提下,根据现实生产环境,构建以完工时间和加工能耗为优化目标的调度模型;对 DE 算法进行控制参数改进,选择基于操作的编码方式将连续算法应用于离散制造过程车间调度问题的求解过程^[4]。最后将该算法应用于实际生产过程中的相关案例中,测试结果表明,该算法对于离散制造过程能耗优化调度问题的最优方案寻找具有速度快、效果好等优点。

1 问题描述

离散制造过程生产与能耗协同优化调度问题可以描述为: m 台设备用于加工 n 个工件,已知各工件在相应加工设备上的单位时间加工耗能量和加工完成所需的时间。加工过程中一台设备不能同时加工多个工件,同理,一个工件也不能同时在多台设备上加工;工件的加工是一个连续的过程;不

同的工件有不同的加工工艺路线。在离散制造过程中,运用相关优化算法对调度问题进行求解,确定不同加工设备上各工件的加工次序,使得所需性能指标达到最优,是离散制造过程优化调度问题的最终目标^[5]。以生产过程中全部工件加工完成后各加工设备所消耗的总能量最少、所有工件加工完成所需要的加工时间最短为期望指标,对离散制造过程调度问题作出如下假设:

1. 工件加工过程连续不间断;
2. 不同工件的加工工序不存在先后顺序约束;
3. 一台设备不能同时加工多道工序;
4. 一个零件不能被多台设备同时加工;
5. 设备加工完一道工序,当前状态为待机时,下一工序可立即开始加工;
6. 从第一个工件第一道工序加工开始,到所有工件完成结束,这一过程设备始终处于开机运行状态。

1.1 参数定义

为了方便建立数学模型,定义符号如下:

n : 工件数量;

m : 加工设备数量;

$J = \{J_i | 1 \leq i \leq n\}$: 表示工件集;

$M = \{M_k | 1 \leq k \leq m\}$: 表示设备集;

O_{ij} : 第 i 个工件的第 j 道工序;

t_{ijk} : 工序 O_{ij} 在设备 M_k 的加工时间;

st_{ijk} : 工序 O_{ij} 在设备 M_k 加工的开始时间;

ot_{ijk} : 工序 O_{ij} 在设备 M_k 的完工时间;

C_{ik} : 工件 J_i 在设备 M_k 的加工时间;

PE : 所有工件加工完成后设备的总能耗;

NE : 设备空转状态的总能耗。

1.2 模型建立

根据离散制造过程生产调度的实际情况,对生产过程进行分析,调度目标函数和约束条件描述如下:

目标函数:分别从完工时间和生产能耗总量两个方面建立目标函数,完工时间用最小化最大完工

时间 f_1 表示, 生产能耗总量用最小化生产耗能总量 f_2 表示, 公式表示为:

$$f_1 = \min \max (\max_{1 \leq i \leq n} \max_{1 \leq k \leq m} C_{ik}) \quad (1)$$

生产能耗总量主要考虑能量消耗量比重较大的加工能耗以及空载能耗。

① 加工能耗: 设备为了完成工序的加工操作所消耗的能量, 加工能耗的计算如公式(2)所示:

$$PE = \sum_{k=1}^m \sum_{i=1}^n (pe_{ik} \times pt_{ik}) \quad (2)$$

式中: pe_{ik} 为设备 M_k 加工工件 J_i 时单位时间所消耗的能量; pt_{ik} 为设备 M_k 加工工件 J_i 所需要的总时间。

② 空转能耗: 设备在开机状态, 但是未进行加工操作时所消耗的能量^[6]。由于设备在加工过程中, 一直保持开机状态, 直到所有工件加工完成才停止运行, 因此, 空转能耗在生产过程中也占很大一部分比例。空转能耗的计算如公式(3)所示:

$$NE = \sum_{k=1}^m (ne_k \times nt_k) \quad (3)$$

式中: ne_k 为设备 M_k 在空转状态下的单位待机耗能量; nt_k 为设备 M_k 的总的空转时间。

由此可得, 最小化生产耗能总量 f_2 如公式(4)所示:

$$f_2 = \min(NE + PE) = \min(\sum_{k=1}^m ne_k \times nt_k + \sum_{k=1}^m \sum_{i=1}^n (pe_{ik} \times pt_{ik})) \quad (4)$$

通过合理的加工顺序安排使得所有零件加工完成后各机器所消耗的能量最低, 以及待机时间最短, 实现总能耗最低与总加工时间最短的目标。

约束条件:

① 设备约束

$$ot_{ijk} - st_{egk} \geq t_{egk} \quad (5)$$

式(5)表示一台设备同一时刻只能加工一个工件;

② 加工过程约束

$$ot_{ijk} - st_{ijk} = t_{ijk} \quad (6)$$

式(6)表示某道工序开始加工, 不可中断;

③ 工艺路线约束

$$st_{ijk} - ot_{i(j-1)h} \geq 0 \quad (7)$$

公式(7)表示同一工件的不同工序之间存在顺序约束。

2 改进差分进化算法

2.1 差分进化算法的基本原理

差分进化算法(Differential Evolution, DE)采用实数编码, 经过变异、交叉和选择操作产生下一代种群, 其基本过程为: 对于一个随机产生的初始种群, 任意选取其中两个个体向量作差, 并与第三个个体向量进行求和, 得到的新个体。然后通过变异、交叉和一对一选择操作, 在两个个体之间选择适应度较高的个体进入下一代种群^[7]。从而通过不断进化, 保留优良个体, 淘汰劣质个体, 使搜索结果不断向最优化方向发展。标准差分进化算法的具体求解过程如下:

(1) 初始化: 由式(8)随机产生 NP 个初始个体 x_i^0 , 构成一个初始种群 $X^0 = \{x_1^0, x_2^0, \dots, x_i^0, \dots, x_{NP}^0\}$ 。

$$x_{i,j}^0 = \text{rand}() \times (x_{i,j}^{\max} - x_{i,j}^{\min}) + x_{i,j}^{\min} \quad (8)$$

式中: $x_{i,j}^0$ 表示个体 i 的第 j 维; $\text{rand}()$ 是在 $[0, 1]$ 之间产生的均匀分布随机数; $x_{i,j}^{\max}$ 和 $x_{i,j}^{\min}$ 分别表示个体 i 第 j 维的上限和下限。

(2) 变异: 对于每一代, 种群中每个个体 x_i^t 执行变异操作产生一个变异个体 v_i^{t+1} 。变异操作是产生新个体的主要操作, 根据个体间的差分向量得以实现, 如公式(9)所示:

$$v_i^{t+1} = x_{r1}^t + F(x_{r2}^t - x_{r3}^t) \quad (9)$$

式中: x_{r1}^t 、 x_{r2}^t 、 x_{r3}^t 为第 t 代种群中 3 个差分向量; $r1$ 、 $r2$ 、 $r3$ 为 $[1, NP]$ 内的随机整数, 变异因子 $F \in (0, 1)$, 用来控制解产生的幅度。

(3) 交叉: 将当前个体 x_i^t 的部分分量用变异个体 v_i^{t+1} 的对应分量替换, 产生实验个体 u_i^{t+1} 。操作由交叉概率因子 CR 来确定, 对变异后的个体 v_i^{t+1} 与当前个体 x_i^t 进行混合。随机生成一个数 $\text{rand}()$, 如果 $\text{rand}()$ 的值大于 CR , 那么变异后个体 v_i^{t+1} 进入新一代种群; 否则, 当前个体 x_i^t 进入新一代种群, 如公式(10)所示:

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{rand()} \leq CR \\ x_{i,j}^t, & \text{rand()} > CR \end{cases} \quad (10)$$

(4) 选择: 分别计算实验个体 u_i^{t+1} 、 x_i^t 的适应值, 采用“贪婪”的搜索策略, 对 u_i^{t+1} 和 x_i^t 执行一对一选择操作, 若 u_i^{t+1} 的适应度值优于 x_i^t , 那么 u_i^{t+1} 就作为子代; 否则, x_i^t 就作为子代, 选择操作如公式(11)所示:

$$x_i^{t+1} = \begin{cases} u_{i,j}^{t+1}, & \text{if}(f(u_i^{t+1})) < f(x_i^t) \\ x_i^t, & \text{otherwise} \end{cases} \quad (11)$$

通过“一对一”贪婪竞争机制得到优于当前个体的新种群中的目标, 从而有利于得种群平均性能的提升。

2.2 自适应算子参数改进策略

DE 算法具有原理简单易于实现、参数较少、易于与其他算法混合等优点, 在诸多领域内具有广泛的应用。由于差分进化算法关键步骤之一的变异操作是利用群体中两不同向量差值来修正个体的信息值^[8], 随着迭代次数的增加, 个体间的差异逐渐缩小, 导致进化后期收敛速度慢, 算法容易陷入局部最优。

为了弥补差分进化算法的不足之处, 提出一种改进的差分进化算法(IDE)用于解决离散制造过程中的车间调度问题。在差分进化算法中, 变异因子 F 和交叉概率因子 CR 是对算法的搜索能力影响较大的两个因子: 当 F 值较大时, 算法拥有较好的全局搜索效果, 但同时降低了算法的收敛速度; 当 CR 值较大时, 算法拥有较快的收敛速度, 但容易跳入局部极优^[9]。对 DE 算法控制参数的改进, 主要针对变异因子 F 和交叉概率因子 CR 进行动态控制, 通过制定合理的参数控制策略可以提高算法的搜索能力和收敛速度。标准差分进化算法中, 变异因子 F 和交叉概率因子 CR 均为固定值, 使得算法在具有良好的全局搜索能力的同时无法兼顾较快的收敛速度。为解决上述难题, 提出一种随着进化代数的变化而改变的自适应变异和交叉算子, 具体如公式(12)和公式(13)所示。

$$F = (F_{\max} - F_{\min}) * \frac{\sqrt{T^2 - t^2}}{T} + F_{\min} \quad (12)$$

$$CR = CR_{\min} + \frac{(CR_{\max} - CR_{\min})t^2}{T^2} \quad (13)$$

式中: t 和 T 分别为当前和最大进化代数; F_{\max} 和 F_{\min} 分别为变异因子 F 的最大值和最小值, 根据式(12)可知, 当 $t=0$ 时, $F=F_{\max}$ 。在算法搜索初期, F 取值较大, 有利于扩大搜索空间, 保持种群的多样性; 随着 t 的逐渐增加, F 取值逐渐减小到 F_{\min} , 即在进化后期随着迭代次数的增加搜索范围逐渐缩小, 有利于在最优区域的周围进行搜索, 从而提高了收敛速度和搜索精度。 CR_{\max} 和 CR_{\min} 分别为交叉概率因子 CR 的最大值和最小值, 由公式(13)可得, 交叉概率因子 CR 的值随着进化代数 t 的增加而增加, 从而使算法表现出前期全局搜索效果好, 后期快速收敛的优点。

3 差分进化算法优化调度方法

3.1 编码调度机制

制造过程生产与能耗优化调度问题是一类离散组合优化问题, 求解过程复杂, 且调度解的数量随着工件和加工设备数量的变化而变化^[10]。由于差分进化算法具有连续性特点, 而生产调度过程是离散空间问题, 为了将 DE 应用于离散制造过程生产与能耗优化调度问题的求解上, 首先要选择合适的编码机制, 将浮点向量映射到排列向量。采用基于操作的编码方式, 利用所有工件号产生的排列来对调度过程进行编码, 不仅可以提高求解效率, 同时也是 DE 应用于调度问题的关键。对于一个在 m 台设备上进行 n 个工件加工的生产调度问题, 编码后的序列包含 $n \times m$ 个正整数, 其中不同正数代表不同的工件号, 正数的数目等于工件数 n , 每个工件号在编码序列中出现 m 次。通过从左向右对编码序列进行扫描, 工件 J_i 第 j 次出现表示该工件的第 j 道工序。对于一个编码序列 [2,1,1,3,2,2,3,3], 从左向右每个位置上的数字具有不同的含义: 首先第一个位置上的数字为 2, 代表首先加工的工件 J_2

的第一道工序; 第二位为数字 1, 代表其次加工工件 J_1 的第一道工序, 第三位的数字 1 代表工件 J_1 的第二道工序, ..., 以此类推, 该编码序列可以表示为序列 $(O_{21}, O_{11}, O_{12}, O_{22}, O_{23}, O_{31}, O_{32}, O_{33}, O_{13})$ 。

由于制造过程优化调度问题是整数领域的组合优化问题^[11], 而差分进化算法采用连续实数编码表示个体, 因此需要将连续域的变量映转换到离散空间中。根据 Smallest-order-value (SOV) 规则对离散制造过程中的加工工件进行排序, 具体操作步骤如下:

Step1. 将个体 $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n \times m}\}$ 中各基因从小到大进行排列得到中间序列 $\phi_i = \{\phi_{i,1}, \phi_{i,2}, \dots, \phi_{i,n \times m}\}$, 其中 $i \in [1, NP]$, NP 为种群的大小。

Step2. 由式(14)计算得到工件的排序:

$$\pi_{i,k} = \lfloor (\phi_{i,k} - 1) / m \rfloor + 1 \quad (14)$$

其中, 维数 k 的变化范围为 1 到 $n \times m$, $\lfloor a \rfloor$ 表示对实数 a 进行取整。

以 3 个工件在 3 台设备上加工的调度问题为例, 随机产生一个如图 1 所示的个体 $x_i = \{0.36, 0.18, 0.25, 0.96, 0.68, 0.14, 0.42, 0.73, 0.51\}$ 。然后对个体中的每一维按升序进行标记, 由于 $x_{i,6}$ 是最小的分量, 则 $x_{i,6}$ 首先被赋予序号 1, 接着 $x_{i,2}$ 被第二个选中赋予序号 2, 然后给 $x_{i,3}$ 赋予序号 3, ..., 以此类推, 得到序列 $\phi_i = \{4, 2, 3, 9, 7, 1, 5, 8, 6\}$; 由公式 (14) 计算可得: 当 $k=1$ 时, 有 $\pi_{i,k} = \lfloor (4-1)/3 \rfloor + 1 = 2$; 当 $k=2$ 时, 有 $\pi_{i,k} = \lfloor (2-1)/3 \rfloor + 1 = 1$, ..., 以此类推, 得到上述个体对应的加工序列 $\pi_i = \{2, 1, 1, 3, 3, 1, 2, 3, 2\}$ 。而此序列即为 DE 在求解调度问题时的编码序列, 代

表调度问题的候选解, O_{ij} 是序列 $\pi_{i,k}$ 对应的操作序列, 其编码过程如图 1 所示。

3.2 解码调度机制

对上述编码序列从左向右进行解码操作, 即相应的加工设备按顺序对上述编码后的个体进行加工。解码过程中, 在满足相应的加工约束条件下, 工件的工序应尽可能的向左移动以减少设备的空转时间。采用全活动调度的解码方式^[12], 根据工序间存在的空闲时间, 尽可能地将新工序插入到这些空闲时间中, 以待插入工序的最早允许加工时间开始加工。该解码方式可以确保在不影响其他工序的加工且满足相关约束的条件下, 所有工序的加工时间均不能再提前, 并且能够缩小搜索区域, 提高算法的搜索能力。

建立设备矩阵 M , 储存工件各个工序对应的加工设备; 建立时间矩阵 T , 存储每台设备加工对应零件时所需的时间。对于一个 n 工件在 m 台设备上加工的调度问题, m_{11} 表示工件 J_1 的第一道工序的加工设备号; t_{11} 表示工件 J_1 的第一道工序的加工时间。如公式(15)所示:

$$M = \begin{bmatrix} m_{11} & \dots & m_{1m} \\ \vdots & \ddots & \vdots \\ m_{n1} & \dots & m_{nm} \end{bmatrix}, T = \begin{bmatrix} t_{11} & \dots & t_{1m} \\ \vdots & \ddots & \vdots \\ t_{n1} & \dots & t_{nm} \end{bmatrix} \quad (15)$$

以 3.1 中 3 个工件在 3 台设备进行加工的编码序列 $(O_{21}, O_{11}, O_{12}, O_{31}, O_{32}, O_{13}, O_{22}, O_{33}, O_{23})$ 为例, 对上述解码过程进行说明, 假设设备顺序矩阵 M 和加工时间矩阵 T 分别为:

$$M = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{bmatrix}, T = \begin{bmatrix} 1 & 3 & 3 \\ 2 & 3 & 1 \\ 2 & 4 & 2 \end{bmatrix} \quad (16)$$

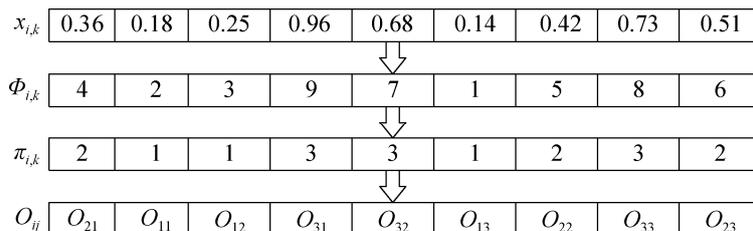


图 1 基于 SOV 规则的编码转换
Fig. 1 Code conversion based on SOV rules

由上述编码序列和矩阵信息可知, 首先, J_2 的第1道工序在 M_3 加工, 用时2个时间单位, 从零时刻开始 M_3 有空闲时间, 直接将 J_2 放入 M_3 上进行加工; 其次, J_1 的第1道工序在 M_1 上加工, 用时1个时间单位, 直接将 J_1 放入 M_1 。接下来是 J_1 的第2道工序在 M_3 上进行加工, 此时 M_3 处于加工状态, 根据设备约束条件, 当 J_1 第1道工序加工完成后, J_1 的第2道工序从第二个时间单位开始加工; 然后, J_3 的第1道工序在 M_2 加工, 从零时刻开始有空闲时间, 直接将 J_3 放入 M_2 上加工; 接着, J_3 的第2道工序在 M_1 加工, 根据加工工艺约束条件, J_3 的第1道工序完成之后, 即在第2个时间单位后, 第2道工序开始进行加工; 同理, 当 J_1 的第2道工序完成后, 即在第五个时间单位后, J_1 第3道工序才允许在 M_2 加工。以此类推将剩下的工件放入对应设备上加工。解码过程如图2所示, 其中横轴表示加工时间, 纵轴表示设备编号。由此可得, 按上述规则进行解码操作, 完成3个工件在3台设备的加工需要8个单位时间。

4 实验结果及分析

为了分析改进差分进化算法对于生产调度问题求解的性能, 根据实际生产过程所得的真实数据, 选取遗传算法(GA)和粒子群算法(PSO)两个典

型群智能算法作为对比, 在 intel CPU, 32 GB 内存的 PC 机上进行了仿真实验, 采用 Matlab 2011b 编程实现生产调度优化算法。对改进差分进化算法的参数设计如下: 种群规模 $NP=500$, 最大进化代数 $T=1000$, 变异因子最值 $F_{\max}=0.9$, $F_{\min}=0.4$, 交叉概率因子最值 $CR_{\max}=0.8$, $CR_{\min}=0.3$ 。

4.1 实验数据设置

针对某工厂实际生产过程中, 6个工件在8台设备上加工, 总共需要26道工序的生产情况, 通过上述三种算法对这一过程进行仿真, 得出对应的加工时间和加工能耗信息。表1为各设备单位时间待机耗能情况, 表2给出了工件加工期间的每台设备所需加工时间和消耗能量。

考虑到算法运行的时间问题, 将迭代次数设为1000次, 循环迭代数为10。为了便于更直观的比较三种算法的求解结果, 同时避免算法最终得到的结果超出实际需求, 将设备的最大完工时间设为75。对于 PSO、GA 算法, 将迭代次数和循环迭代次数与 IDE 算法保持一致, 此外, 需要对这两种算法的其他参数进行设置, 其中, 将 PSO 算法参数设置为: $C_1 = C_2 = 1.5$, $w_{\max} = 0.9$, $w_{\min} = 0.4$; GA 算法设置为: $p_a = 0.8$, $p_b = 0.1$, $L_c = 1$ 。

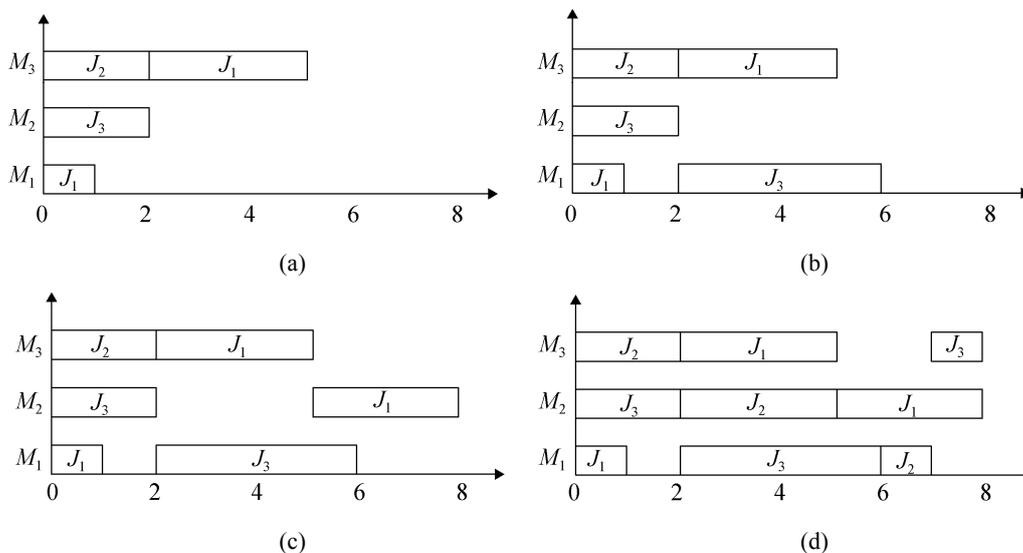


图2 解码调度过程示意图
Fig. 2 Schematic diagram of decoding scheduling process

表 1 设备单位时间待机能耗/kW·h

Tab. 1 Machine standby energy consumption per unit time/kW·h

机器号	待机能耗	机器号	待机能耗	机器号	待机能耗	机器号	待机能耗
M_1	0.30	M_5	0.21	M_3	0.24	M_7	0.33
M_2	0.42	M_6	0.24	M_4	0.21	M_8	0.42

表 2 工件加工信息表

Tab. 2 Workpiece processing information table

工件	工序	加工能耗(kW/h)/加工时间(h)							
		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
J_1	1	-	4.60/12	-	-	4.89/14	-	6.15/20	-
	2	5.83/18	-	-	6.31/19	-	-	3.95/11	-
	3	-	-	5.51/14	-	-	5.76/17	-	-
	4	-	4.13/11	-	-	3.42/9	-	-	4.47/12
	5	4.81/15	-	-	3.42/8	-	-	-	6.29/18
J_2	1	-	-	4.43/12	5.92/19	4.83/14	-	-	-
	2	3.06/8	-	-	3.09/9	-	-	4.76/15	-
	3	4.95/16	3.19/7	-	-	-	3.69/9	-	-
J_3	1	-	-	-	4.24/11	3.80/10	-	-	4.50/13
	2	-	4.54/12	5.83/18	-	-	-	4.50/14	-
	3	3.08/9	-	-	4.50/15	2.93/7	-	-	-
	4	-	4.53/12	-	-	-	2.67/5	-	3.09/9
	5	1.79/3	-	-	2.59/4	-	3.10/8	-	-
J_4	1	-	6.05/19	-	-	3.01/7	-	4.44/13	-
	2	-	-	3.25/8	-	3.96/11	-	-	4.73/16
	3	-	4.27/11	-	3.74/8	-	-	-	-
	4	2.26/6	-	-	-	-	-	4.44/14	3.58/9
J_5	1	-	-	-	7.43/22	-	4.39/12	4.89/17	-
	2	-	4.11/18	-	-	3.46/11	-	-	3.60/9
	3	3.37/9	-	4.55/12	-	-	-	-	2.72/7
J_6	1	-	4.19/11	-	-	-	3.48/9	-	-
	2	3.11/8	-	-	-	3.06/6	-	-	3.79/9
	3	-	-	4.07/11	-	5.95/17	-	-	5.98/18
	4	1.92/5	-	-	5.54/15	-	-	2.79/7	-
	5	-	4.47/11	-	-	-	2.96/8	-	2.98/7
	6	6.07/19	-	-	4.08/7	-	4.71/15	-	-

4.2 实验结果分析

将 GA, PSO 和 IDE 算法分别应用于离散制造过程生产与能耗协同优化, 表 3 列出了上述三种算法在不考虑最大完工时间时机器总能耗的最优值和平均值, 从最优值和平均值两个评价指标可以看出 IDE 算法对于求解能耗最低方面明显优于 PSO 和 GA。图 3~5 是三种算法生成的最优调度方案的甘特图, 通过对比可以发现 IDE 算法得到的调度方案中完工时间和机器空闲时间明显短于 PSO、

GA 算法求得的调度方案, 证明该算法在求解离散制造过程加工能耗和加工时间优化问题具有一定的优越性。

表 3 三种智能算法设备总能耗值/kW·h

Tab. 3 Total energy consumption of three intelligent algorithms devices /kW·h

算法	最优值	平均值
PSO	121.35	128.79
GA	112.42	116.73
IDE	102.23	107.15

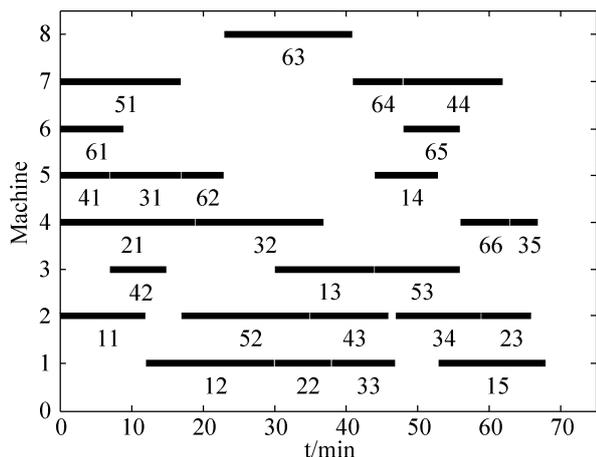


图3 IDE 算法最优调度甘特图
Fig. 3 IDE algorithm for optimal scheduling Gantt chart

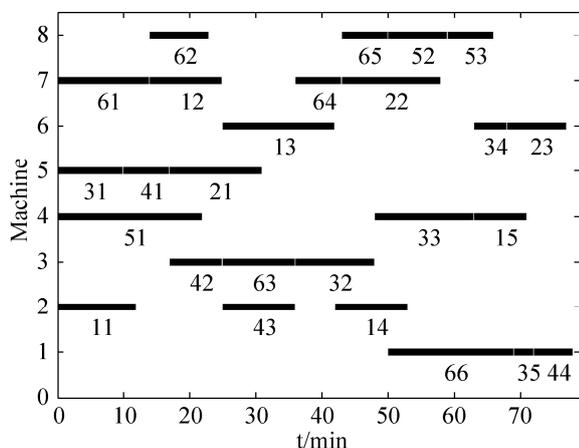


图4 GA 算法最优调度甘特图
Fig. 4 GA algorithm for optimal scheduling Gantt chart

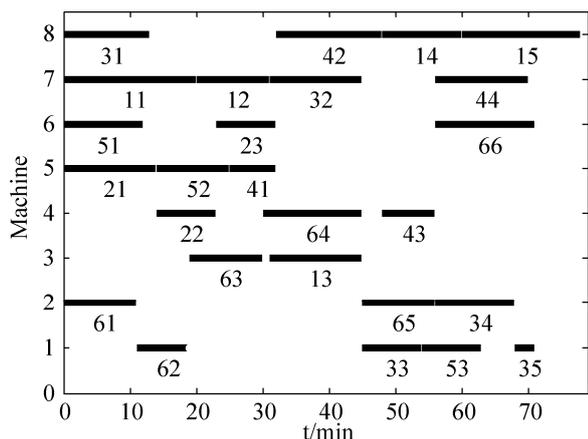


图5 PSO 算法最优调度甘特图
Fig. 5 PSO algorithm for optimal scheduling Gantt chart

5 结论

离散制造过程生产与能耗协同优化调度问题

的研究, 对提高企业竞争力有着极其重要的意义。本文面向离散制造过程, 建立以加工过程中最大完工时间最短和最小化能量消耗作为优化目标的调度模型, 根据实际生产过程, 对算法控制参数进行改进, 采用一种基于自适应变异、交叉概率因子的改进差分进化算法, 求解优化调度问题, 加快了算法寻优过程中的全局搜索速度以及增强跳出局部最优解的能力。最后通过具体应用实例对改进后的算法性能进行了分析, 结果表明, 相对于对比算法, 改进差分进化算法对车间作业优化调度问题的求解速度快, 可以得到更满意的优化调度方案, 具有一定的实际应用价值。

参考文献:

- [1] Steuben J C, Iliopoulos A P, Michopoulos J G. Discrete element modeling of particle-based additive manufacturing processes[J]. *Computer Methods in Applied Mechanics & Engineering* (S0045-7825), 2016, 305(15): 537-561.
- [2] 张静, 王万良, 徐新黎, 等. 混合粒子群算法求解多目标柔性作业车间调度问题 [J]. *控制理论与应用*, 2012, 29(6): 715-722.
Zhang Jing, Wang Wanliang, Xu Xinli, et al. Hybrid particle swarm optimization algorithm for multi-objective flexible job shop scheduling problem[J]. *Control Theory and Application* (S1000-8152), 2012, 29(6): 715-722.
- [3] Zhen Y J, Xiao H L, Jiang Y Y, et al. Research on job-shop scheduling problem based on genetic algorithm [J]. *International Journal of Production Research* (S0020-7543), 2011, 49(12): 3585-3604.
- [4] Shasavari-Pour N, Ghasemishabankareh B. A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling[J]. *Journal of Manufacturing Systems* (S0278-6125), 2013, 32(4): 771-780.
- [5] Bruzzone A A G, Anghinolfi D, Paolucci M, et al. Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops [J]. *CIRP Annals - Manufacturing Technology* (S0007-8506), 2012, 61(1): 459-462.
- [6] 王伟. 面向能耗优化的车间调度方法研究及其应用环境开发 [D]. 哈尔滨: 哈尔滨工业大学, 2011.
Wang Wei. Research on workshop scheduling method for energy consumption optimization and its application environment development[D]. Harbin: Harbin Institute of Technology, 2011.

(下转第 4386 页)