

多目标柔性作业车间稳健性动态调度研究

朱传军¹ 邱文¹ 张超勇² 金亮亮²

1.湖北工业大学机械工程学院,武汉,430068

2.华中科技大学数字制造装备与技术国家重点实验室,武汉,430074

摘要:研究了以调度稳定性和鲁棒性为优化目标的柔性作业车间动态调度问题。建立了多目标柔性作业车间动态调度问题模型;针对动态事件特点,采用工件到达发生器和机器故障预维护发生器模拟动态事件,提出了基于周期与事件驱动的混合重调度策略,该策略在每个重调度点对工件进行完全重调度或右移重调度;为克服传统差分算法早熟的缺点,设计了一种改进的多目标差分进化算法求解该调度问题;实例验证了提出策略和算法的有效性。

关键词:柔性作业车间动态调度问题;多目标;动态调度;稳健性;差分进化

中图分类号:TP18

DOI:10.3969/j.issn.1004-132X.2017.02.009

Multi-objective Flexible Job Shop Dynamic Scheduling Strategy Aiming at Scheduling Stability and Robustness

ZHU Chuanjun¹ QIU Wen¹ ZHANG Chaoyong² JIN Liangliang²

1.School of Mechanical Engineering,Hubei University of Technology,Wuhan,430068

2.State Key Laboratory of Digital Manufacturing Equipment and Technology,
Huazhong University of Science and Technology,Wuhan,430074

Abstract: A dynamic FJSP aiming at scheduling stability and robustness was studied. Firstly, an optimization mathematical model was proposed based on scheduling stability and robustness. Secondly, by analyzing characteristics of dynamic FJSP, a job arrival generator and a predictive maintenance generator of machine breakdown were presented. In each rescheduling points, periodic and event driven rescheduling strategies were employed, and a complete rescheduling strategy combined with right-shift rescheduling strategy was proposed. In order to preserve the good characteristics of paternal generations, an improved differential evolution algorithm was designed to solve the dynamic FJSP. The validity of the proposed strategies and algorithm was verified by instances.

Key words: flexible job-shop scheduling problem(FJSP); multi-objective; dynamic schedule; robustness; differential evolution algorithm

0 引言

针对动态作业车间调度问题,潘全科等^[1]研究了作业车间动态调度问题中待加工工件的选取原则及再调度方法,但没有考虑机器柔性;ADIBI等^[2]针对工件随机到达和机器故障的动态调度问题,提出了基于周期和事件驱动的混合重调度策略,但是只考虑了完工时间和拖期这两个时间指标,没考虑调度方案本身的稳健性;HE等^[3]针对机器故障这一类动态调度问题,提出插入空闲时间、工件右移调度和变路径调度等方法来提高调度方案的稳定性和鲁棒性,但忽略了其他动态因素的影响;FANG等^[4]研究了具有工件随机到

达、机器故障、机器维修和变交货期等多变量的动态环境,采用一种滚动窗口机制求解该动态问题,但没考虑多目标的情况;LI等^[5]研究了机器预维护情况下的柔性作业车间调度问题(flexible job-shop scheduling problem, FJSP),在加工过程中动态加入机器维护,但只考虑了时间性能;FATTAHI等^[6]针对工件随机到达情况下的柔性作业车间动态调度问题,提出完整的动态调度问题模型,并采用改进遗传算法求解,但没有考虑机器故障的情况;XIONG等^[7]研究了随机机器故障情况下的FJSP,采用机器故障水平和机器平均维修时间描述机器故障,但没有考虑其他动态因素;AL-HINAI等^[8]采用两阶段混合遗传算法求解机器随机故障情况下的FJSP,没有考虑工件随机到达的情况。实际生产中,既要考虑工件随机到达,又要考虑机器故障等动态因素,目前国内对同时考虑两种动态因素的柔性作业车间调度的研究

收稿日期:2016-03-18

基金项目:国家自然科学基金国际(地区)合作与交流项目(51561125002);国家自然科学基金资助项目(51275190, 51575211);中央高校基本科研业务费专项资金资助项目(HUST:2014TS038)

较少。

因此,笔者结合实际生产环境,深入研究工件随机到达和机器故障预维护两类动态事件触发的柔性作业车间动态调度。以调度方案稳定性和鲁棒性为优化目标,建立动态 FJSP 数学模型,采用工件随机到达发生器和机器故障预维护发生器来模拟车间生产过程中发生的动态事件,并提出基于周期与事件驱动的混合重调度策略,将各工件划分到对应的调度区间。在各调度区间,运用完全重调度和右移重调度策略,并采用改进多目标差分进化算法对其进行求解。最后验证了提出方法的有效性。

1 动态 FJSP

1.1 问题描述

本文主要考虑工件随机到达和机器故障预维护两类动态因素对生产车间的影响。

(1)随着时间的推移,不断有一定数量的新工件动态到达调度系统,针对工件的随机到达,大多数研究均认为工件的动态到达近似服从泊松分布,且任意两个相邻到达工件的到达时间间隔近似服从指数分布^[9-10]。

(2)机器故障预维护。假设每台机器故障预维护发生的时间间隔近似服从指数分布^[9]。故障维护间隔时间均值(mean time between failure, MTBF)和故障维护时间均值(mean time to repair, MTTR)是与机器故障预维护相关的两个参数。

在周期与事件驱动的混合重调度中,一旦机器的累计加工时间超过一定阈值或到达固定重调度周期就会触发一次重调度。

1.2 问题模型

为较好地适应车间动态环境,文中将完工时间 T 定义为调度区间内所有工件的最大完工时间:

$$T = \max(\max_{1 \leq i \leq n} c_{inik}, \max_{1 \leq i' \leq n'} c_{i'n'ik'}) \quad (1)$$

式中, c_{inik} 、 $c_{i'n'ik'}$ 分别为旧工件与新工件每道工序的完工时间。

调度方案的鲁棒性指标 R 表示调度方案对动态环境的免疫程度,用重调度产生的新调度方案与初始调度方案中参与调度但还未执行的所有工件的完工时间差值的绝对值之和表示:

$$R = \sum_{i=1}^{n'} |c_{pi} - c_{ri}| \quad (2)$$

式中, c_{pi} 、 c_{ri} 分别为工件 i 在预调度方案和重调度方案中的完工时间; n' 为新调度方案与初始调度方案中参与调度但还未被加工的工件总数。

由于工件预调度在某一机器上加工时,动态

事件的发生将导致该工件的开始加工时间发生变化,进而引起生产资源重新分配。初始调度中各机器上分配的工序数在重调度时也可能存在差异,因此本文的调度稳定性主要由工序偏离度 d_o 和机器偏离度 d_m 两个基本稳定性指标表示^[11]。工序偏离度定义为重调度方案与初始调度方案中在重调度时刻所有未加工完成的工序的开始时间总偏差:

$$d_o = \sum_{i=1}^n \sum_{j=1}^{n_i} p_j |s_{ijk} - s_{ijk'}| \quad (3)$$

式中, s_{ijk} 、 $s_{ijk'}$ 分别为预调度和重调度中工序 $O_{i,j}$ 在机器 k 和 k' 上的开始加工时间; p_j 为重调度导致工序开始加工时间变化而引起的惩罚系数。

文献[8]在求解多目标动态 FJSP 时,为简化计算复杂度,取 $p_j = 1$,本文同样取 $p_j = 1$ 。柔性作业车间动态调度中,每道工序可在多台机器加工,发生重调度时,分到每台机器上的工序可能会增加或减少,而工序的增加或减少将导致机器夹具或道具等发生变化,因此,在考虑稳定性时同样要考虑到机器的偏离度 d_m ^[11]。机器偏离度表示重调度方案与初始调度方案中所有机器上加工工序数的总偏差:

$$d_m = \sum_{k=1}^m p_k y_k \quad (4)$$

其中, p_k 为初始调度中工序在重调度时刻发生变化引起的机器惩罚系数,为简化计算复杂度,同样可参考文献[11]取 $p_k = 1$; y_k 为机器 k 上从初始调度增加和减少的总工序数。

对于 n 个工件的 $\sum_{i=1}^k n_i$ 道工序在 m 台机器上加工的动态 FJSP,初始调度的偏离度 d_i 定义为工序偏离度 d_o 和机器偏离度 d_m 的总和:

$$d_i = \sum_{i=1}^n \sum_{j=1}^{n_i} p_j |s_{ijk} - s_{ijk'}| + \sum_{k=1}^m p_k y_k \quad (5)$$

实际生产中,工序间存在一定的加工约束条件,下面分析具体的约束条件。

机器分配约束条件,即每道工序需且仅需分配给一台机器:

$$\sum_{k=1}^m X_{i,j,k} \leq 1 \quad \forall i, j \quad (6)$$

$$\sum_{k=1}^m X'_{i',j',k} \leq 1 \quad \forall i', j' \quad (7)$$

式中, $X_{i,j,k}$ 表示初始调度中工件 i 的第 j 道工序是否分配到机器 k 上加工; $X'_{i',j',k}$ 表示重调度中工件 i' 的第 j' 道工序是否分配到机器 k 上加工。

每道工序一旦开始,加工就不能中断,除非机器故障,可用下式表示:

$$C_{i,j} = S_{i,j} + \sum_{k=1}^{M_{i,j}} P_{i,j,k} X_{i,j,k} \quad \forall i,j \quad (8)$$

$$C'_{i',j'} = S'_{i',j'} + \sum_{k'=1}^{M'_{i',j'}} P'_{i',j',k'} X'_{i',j',k'} \quad \forall i',j' \quad (9)$$

工序加工顺序约束条件,即各工件的前任工序完工后才能安排其后续工序加工:

$$C_{i,j+1} \geq S_{i,j} + \sum_{k=1}^{M_{i,j}} P_{i,j,k} X_{i,j,k} \quad \forall i,j \quad (10)$$

$$C'_{i',j'+1} \geq S'_{i',j'} + \sum_{k'=1}^{M'_{i',j'}} P'_{i',j',k'} X'_{i',j',k'} \quad \forall i',j' \quad (11)$$

各工序在机器上的先后约束关系如下:

$$C_{h,l,k} - p_{h,l,k} + M(1 - Y_{ijhk}) \geq C_{i,j,k} \quad (12)$$

$$C'_{h',l',k'} - p_{h',l',k'} + M(1 - Y_{i'j'h'l'}) \geq C'_{i',j',k'} \quad (13)$$

其中, M 为足够大的正实数; Y_{ijhk} 为 0-1 变量, $Y_{ijhk} = 1$ 表示工序 $O_{i,j}$ 先于工序 $O_{h,l}$ 在机器 k 上加工,反之亦然。

2 动态重调度方法

2.1 动态重调度流程

文中所提动态重调度流程包括预调度方案、发生器、问题条件更新和重调度算法^[9],具体流程如图 1 所示。

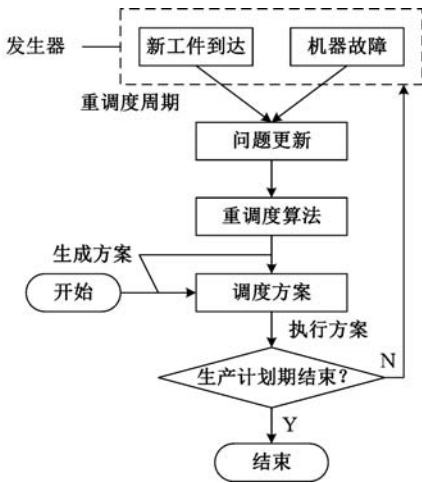


图 1 柔性作业车间重调度方法流程图

Fig.1 Flow chart of flexible job shop scheduling methods

2.2 发生器

在调度的初始时刻,所有工件和机器的相关参数已预先确定。在每个重调度时刻点,调度系统将重新产生新的预调度方案并执行该方案直至下一个重调度时刻点,保证车间生产的连续性。发生器的主要作用是在生产过程中不断产生动态扰动直到调度结束。与工件相关的参数和与机器相关的参数是发生器的两个主要参数,其参数的产生过程及有关符号说明如下: t_{JAEL} 为工件到达时间间隔; t_{MTBF} 为每台机器的预维护发生时间均

值; t_{MTTR} 为每台机器预维护的时间均值; t_{FEL} 为每台机器预维护发生的时间点,当该机器进行预防性维护时, t_{FEL} 归零; t_{Life} 为每台机器上的累积负荷,当机器进行预防性维护时, t_{Life} 归零; t_{repair} 为每台机器的预维护时间。

很多研究显示,工件的随机到达近似服从指数分布,本文工件到达发生器包括以下三个步骤:初始化工件到达时间间隔;计算新工件的到达时间;判断是否触发重调度,具体参见文献[12]。

机器预维护是根据机器故障的统计规律制定出长远的维修计划,相比于机器故障后维护,从经济成本和实用性来看,预防性维护更加经济、有效,且可在一定程度上提高系统对动态扰动的免疫程度。机器预维护发生器的实施具体参见文献[12]。

2.3 问题条件更新

动态调度中,在进行重调度前需要对重调度问题的条件进行更新。问题条件更新主要包括机器可获得时间 A_k (k 为机器编号)、参与重调度的所有工件以及各工件的工序数。当重调度被触发时,首先依据各机器的加工状态,将调度系统中的机器划分为三类:故障预维护机器、正在加工机器(该机器正在加工工件)和空闲机器。其中,故障机器的可用时间为该机器的修复时刻 R'_k ,记 $A_k = R'_k$;正在加工机器的可获得时间定义为机器上正在加工工序的完工时间,记 $A_k = C_k$,其中, C_k 为重调度时刻点机器 k 上正在加工工序的完工时间;空闲机器的可用时间为机器故障时刻点 B_k ,记 $A_k = B_k$ 。重调度工件集更新是指将当前重调度时刻已经加工完成的工件和正在加工的工件从工件集中剔除,对未完工工件及该工件的剩余工序和在上个重调度点与当前重调度时刻到达的新工件及新工序数进行统计更新。

3 重调度算法

差分进化(differential evolution, DE)算法是一种基于实数编码和差分思想的随机并行优化算法。本文设计的带局部搜索的多目标差分算法(multi-objective differential evolution algorithm with local search, MODE_L)流程如图 2 所示。MODE_L 算法与基本多目标差分(multi-objective differential evolution algorithm with no local search, MODE_NL)算法的关键区别是: ①MODE_L 算法提出了新的种群初始化策略; ②在交叉和变异之后增加变邻域局部搜索操作,针对机器部分和工序部分分别设计了邻域结构; ③在算法的最后加入了 Pareto 决策。

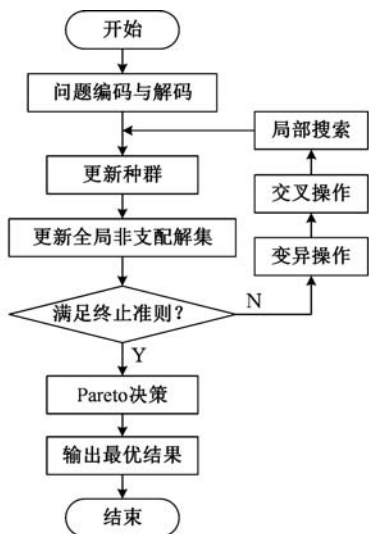


图 2 MODE_L 算法流程

Fig.2 Flow chart of MODE_L algorithm

3.1 种群初始化

本文采用文献[13]提出的混合初始化策略,在每个重调度时刻,为保证初始种群的多样性和解的有效性,针对机器部分和工序部分分别设计初始化方法。

3.2 编码与解码

3.2.1 染色体编码

由于工件随机到达、机器故障等动态事件的出现,本文将每个重调度时刻点的所有未加工工件和新到达工件归入重调度工件集中。例如,当一个重调度触发时,未加工工件集为 $\{O_{12}, O_{13}, O_{23}\}$,新工件集为 $\{O_{31}, O_{32}, O_{33}\}$,则重调度工件集为 $\{O_{12}, O_{13}, O_{23}, O_{31}, O_{32}, O_{33}\}$ 。以重调度工件集 $\{O_{12}, O_{13}, O_{23}, O_{31}, O_{32}, O_{33}\}$ 为例,对每道工序进行编号,得到工序染色体向量 S ,如表 1 所示。在进化过程中,当向量 $S = (3, 4, 1, 5, 2, 6)$ 时,则工序对应的加工顺序为 $O_{2,3} > O_{3,1} > O_{1,2} > O_{3,2} > O_{1,3} > O_{3,3}$,即首先安排工序 $O_{2,3}$ 加工,然后安排工序 $O_{3,1}$,后面依次类推。

表 1 工序染色体编码表

Tab.1 Process of chromosome coding

工序	$O_{1,2}$	$O_{1,3}$	$O_{2,3}$	$O_{3,1}$	$O_{3,2}$	$O_{3,3}$
染色体向量 S 对应元素	1	2	3	4	5	6

用整数序列 $R = (r_1, r_2, \dots, r_d)$ 表示机器染色体,其中, $r_j (j = 1, 2, \dots, d)$ 表示第 j 个工件选择其可选机器集中第 r_j 个机器安排加工。

表 1 所示问题的一个机器分配序列如图 3 所示。由图 3 可知,工序 $O_{3,2}$ 选择的是第 2 个可加工机器,即机器 M_3 。设整数序列 $L = (l_1, l_2, \dots, l_d)$,其中, l_j 表示工序 j 的可选机器数,按照下式

进行转换,确定每道工序的加工机器:

$$r_j = \text{round} \left(\frac{(l_j - 1)(x_j + \delta)}{2\delta} + 1 \right) \quad (14)$$

其中, $\text{round}(\cdot)$ 为取整函数, x_j 为实数序列 X_1 中的第 j 个实数。 $l_j = 1$ 表示工序 j 只有一个机器可供加工,此时不论 x_j 取何值, r_j 始终等于 1。

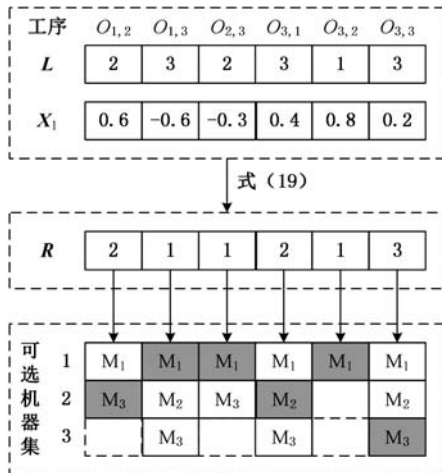


图 3 机器染色体示意图

Fig.3 Schematic diagram of machine chromosome

3.2.2 染色体解码

设工序 O_i 的开工时间、加工时间分别为 t_{si} 、 p_i ,则工序 O_i 的完工时间为 $t_{si} + p_i$;设工序 O_i 的工件紧前工序、机器紧前工序分别为 O_{ji} 和 O_{mi} (如果它们存在)。在重调度时刻 t_1 ,某些工件正在被加工,由于加工的延续性,正在加工这些工件的机器只能等该工件加工结束后才能参与重调度,因此在解码过程中应对其分开进行考虑。

对于正在加工工件,该工序在机器上的后续工序的开始加工时间取该工序的工件前任工序 O_{ji} 和机器前任工序 O_{mi} 的完工时间中的较大值:

$$t_{si} = \max(t_{sjp_i} + p_{jp_i}, t_{smip_i} + p_{mp_i}) \quad (15)$$

对于未加工工件,其第一道工序的开始加工时间取重调度时刻 t_1 和该工序的机器前任工序 O_{mi} 的完工时间中的较大值:

$$t_{si} = \max(t_1, t_{smip_i} + p_{mp_i}) \quad (16)$$

对于正在加工工件和未加工工件的非第一道工序,其开始加工时间仍依据式(8)、式(9)计算。

为保证解码产生的调度为主动调度,本文采用插入式贪婪解码算法,将每道工序插入到所分配机器上的最佳可加工时刻加工。

需要说明的是,基于实数的编码中,按实数序列降序排列后得到的工序染色体 S 中可能存在同一工件的相邻工序间加工顺序相互冲突的情况,如 $S = (5, 1, 3, 6, 4, 2)$,序号 5、6、4 分别表示工件 3 的第 2 道、第 3 道和第 1 道工序,此时无法解码出

可行调度解。当出现这种情况时,应对 S 做相应的调整,保证各工件相邻工序间的加工顺序约束条件得到满足^[14]。

3.3 Pareto 排序

本文采用一种快速排序方法构造非支配集,该方法在进化个体之间定义一种新的关系: $\forall X, Y \in P$, 如果 X 支配 Y 或者 X 和 Y 不相关,则 $X \succ_d Y$ 。快速排序方法每次找单个个体 X 作为比较对象,按照支配关系“ \succ_d ”进行判断,接着进行逐轮排序。

3.4 变异操作

在 XUE 等^[15] 提出的多目标差分算法的变异操作中,参与变异的 2 个个体是在种群中随机选择的 2 个互不相同的个体,该方法在查找全局非支配解时具有随机性。本文依据变异个体的稳定性和鲁棒性,选择稳定性指标较高的个体参与变异操作。

测试发现,依据两种评价指标选择的个体进行变异操作时不仅可以在一定程度上提高解的质量,还能提高算法找到全局非支配解的速度。

3.5 交叉操作

YUAN 等^[16] 提出了一种基于实数编码的交叉操作并成功运用于 FJSP 的求解。具体的交叉过程如图 4 所示,经过变异操作后得到变异个体 V_i 的实数染色体 $X'_i = (0.4, -0.1, 0.2, 0.6, 0.3, 0.7, -0.3)$, 机器染色体 $R' = (2, 1, 1, 4, 2, 3, 1)$, 产生 0、1 之间的随机数 α 及随机基因位 q , 若 $\alpha \leq P_c$ (P_c 为交叉概率) 或 $q = j$, 则基因位 j 取变异个体中对应基因位上的值, 否则, 该基因位不变。交叉操作如下:

$$U_{j,i,G} = \begin{cases} v_{j,i,G} & (\alpha \leq P_c) \cup (q = j) \\ x_{j,i,G} & \text{其他} \end{cases}$$

式中, $v_{j,i,G}$ 表示变异操作后产生的新个体, $x_{j,i,G}$ 表示原始种群中的个体。

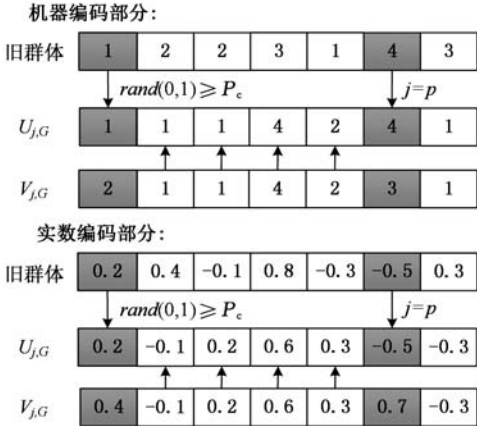


图 4 交叉操作示意图

Fig.4 Schematic diagram of crossover operation

3.6 局部搜索

局部搜索可以有效地提高算法解的质量和算法的收敛速度。本文针对 FJSP 特性提出两类邻域结构: 机器部分的邻域结构 N_r 和工件部分的邻域结构 N_s 。

机器部分的邻域结构 N_r 仅对机器染色体进行邻域更新^[13]。如图 5 所示, 邻域结构 N_{r1} 表示随机选择机器染色体上的基因 1, 则与该机器染色体对应的工序染色体中, 基因 1 所对应工序的可选机器集中的机器码 2、3 均为此基因位的邻域, 随机选择机器码 2 代替机器码 1 形成新的邻域结构 N_{r1} ^[17]。邻域结构 N_{r2} 主要考虑机器负荷平衡, 为平衡机器负荷, 从待加工工序的所有可选机器集中, 选择加工时间最短的机器 3 替换机器 1, 得到邻域结构 N_{r2} 。该方法可以有效减小机器总负荷和关键机器负荷, 有助于提高调度稳定性^[5]。

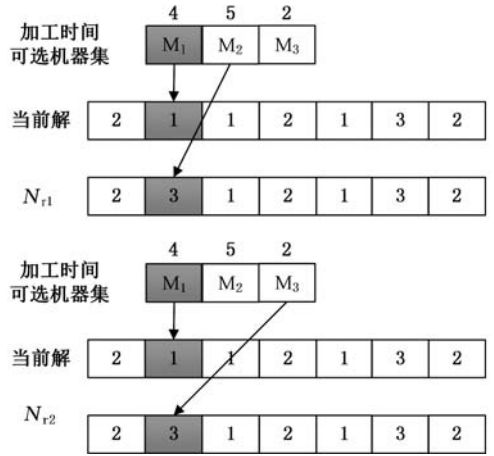


图 5 邻域结构 N_{r1} 、 N_{r2} 示意图

Fig.5 Schematic diagram of neighborhood structure N_{r1} and N_{r2}

工件部分的邻域结构 N_{s1} 、 N_{s2} 、 N_{s3} 是通过工序染色体进行位置调整产生的。三种邻域结构分别通过对工序码中随机选择的基因进行交换、反转和插入操作获得^[9], 图 6 为这三种邻域结构的示意图。

3.7 Pareto 解决策

在调度区间中, 采用基于 Pareto 优化的方法进行求解后最终会得到一个非支配解集, 而每次调度只能选出一个可行方案安排加工, 因此, 需从当前的非支配集合中选择一个满意的调度方案作为下一个调度区间的基准。在对非支配解进行决策选择时, 本文采用一种有效的 TOPSIS 方法, 该方法依据众多评价对象与理想化目标的接近程度对解进行排序, 使最终解在一定程度上逼近于理想解。

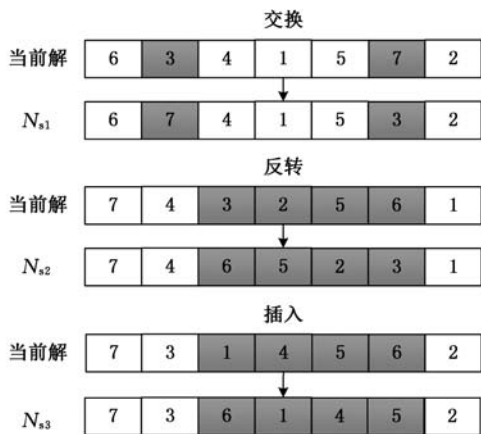


图 6 邻域结构 N_{s1} 、 N_{s2} 、 N_{s3} 示意图

Fig.6 Schematic diagram of neighborhood structure N_{s1} , N_{s2} and N_{s3}

4 实例结果与分析

本文算法用 C++ 语言编程,运行环境为 P4 CPU,主频 3.2G Hz,内存 3 GB。实例测试分为两部分,第一部分为了评估和比较提出算法的性能,对相同问题采用不同调度算法和调度规则进行测试;第二部分通过模拟实际生产车间数据的实例进行测试,得出工件到达率和工件到达数量对调度性能的影响。经测试,本文多目标差分进化算法运行的参数设置如下:种群规模 $N = 100$,交叉概率 $P_c = 0.3$,缩放因子 $F = 0.1$,贪婪算子 $\lambda = 0.5$,局部搜索概率 $P_l = 0.5$,局部搜索迭代次数 $g = 20$,最大进化代数 $G = 80$ 。由于每个重调度时刻参与重调度的工件数可能不一样(新工件到达,旧工件加工完成),为了测试方便,各个调度区间的种群规模和进化代数均取相同值。

4.1 算法性能分析

为了验证本文提出的重调度方法和策略在求解动态 FJSP 上的可行性与有效性,使用文献中通用的一些调度规则^[18]和方法求解本文的动态调度问题,并将求解结果进行对比分析。本文对比的重调度方法包括最短加工时间(SPT)规则、最长加工时间(LPT)规则、人工蜂群(ABC)算法^[5]、先进先出(FIFO)规则、混合粒子群(HPSO)算法^[19]、后进先出(LIFO)规则、多目标进化算法(MOEA)^[20]、MODE_NL 算法。MODE_NL 与 MODE_L 的不同之处在于没有考虑局部搜索。

假设所有的机器具有相同的 t_{MTBF} 和 t_{MTTR} 。车间机器故障水平 $A_g = t_{MTTR} / (t_{MTBF} + t_{MTTR})$,当 $A_g = 0.05$, $t_{MTTR} = 5$ 时, $t_{MTBF} = 45$,表示每台机器故障间隔时间为 45 个时间单位,修复时间为 5 个时间单位。

定义参数 γ 为每单位时间到达的新工件数量(工件到达率),且是服从泊松分布的随机变量。 $\gamma = 1/4$ 表示每隔 4 个单位时间就有一个新工件到达。

本文研究了 4 种不同的工件到达率和 6 种不同的新工件数量 J_n 对调度性能的影响情况。为了保证实验对比的公正性,本文的调度周期和机器故障水平等均不变,加工机器数量根据工件规模的不同分别取不同值。机器平均故障时间 t_{MTTR} 和平均修复时间 t_{MTBF} 均服从指数分布,机器和工件的到达时间间隔均服从指数分布,新工件加工时间服从均匀分布。调度的性能指标为 makespan、完工时间偏差、开始时间偏差和机器偏差。表 2 给出了具体的实验数据。

表 2 实验数据

Tab.2 Experimental data

元素	特征	规格
机器	数量	4, 6, 10
	机器故障水平	0.05
	t_{MTTR}	5
	工件到达率	1/8, 1/4, 1/2, 1
	新工件数量	20, 40, 60, 80, 100, 200
	调度周期	20

表 3 给出了不同工件到达率下不同方法的 makespan,其中新工件数量取 20、40、60、80、100、200, γ 取 1/8、1/4、1/2、1, S、M、L 分别表示新工件为小规模、中规模、大规模,不同的新工件规模对应不同的加工机器数。表 3 中的粗体数据表示求解同类问题时得到的最优解。分析表 3 数据可知,调度算法比两种调度规则产生的解较好。在所有调度算法中,MODE_NL 算法解的质量最差,MODE_L 算法解最优,VNS 和 MOEA 算法次之。此外,VNS 算法在工件稀疏到达($\gamma = 1/8, 1/4$)时的求解性能与 MODE_L 算法比较接近,优于 MOEA 算法;工件密集到达($\gamma = 1/2, 1$)时,VNS 算法的求解结果明显差于 MODE_L 算法,但与 MOEA 算法接近。此外,随着新工件数量的增加,4 种调度规则的 makespan 迅速增大,GA 算法得到的最大完工时间随着新工件数量增加有所增长,但其增长速度较为缓慢,说明结果鲁棒性较好。将 MOEA 算法与 MODE_L 算法进行对比,发现 MODE_L 算法所得结果与 MOEA 算法所得结果相近。以上分析可以说明 MODE_L 算法能有效求解动态 FJSP。

在调度问题求解过程中,算法的运行时间直接影响着动态调度性能的优劣。表 4 列出了所有方法求解同一动态调度问题所需的 CPU 时间(取每个调度周期的平均值)。

表 3 不同工件到达率下对应的 makespan
Tab.3 Makespan under different arrival rate of jobs

γ	J_n	MODE_L	MODE_NL	VSN	MOEA	SPT	LPT	ABC	HPSO	
1/8	S	20	44	56	44	52	74	52	135	164
		40	92	113	95	110	107	113	164	277
	M	60	152	178	163	190	182	178	274	452
		80	252	286	252	278	263	286	624	2342
	L	100	328	356	328	328	368	356	758	1640
		200	614	651	615	614	635	651	1236	4048
1/4	S	20	50	58	53	50	56	58	152	171
		40	94	107	108	102	126	107	162	263
	M	60	142	156	146	147	173	156	425	776
		80	231	246	241	243	230	246	654	828
	L	100	461	487	465	478	511	487	1532	1714
		200	830	858	835	853	857	838	1768	1967
1/2	S	20	1041	1067	1052	1054	1058	1047	2179	2398
		40	2060	2469	2061	2060	2065	2069	4158	5186
	M	60	4365	4388	4397	4387	4379	4388	12 241	13 274
		80	6983	6994	6996	6990	7102	6994	12 320	15 328
	L	100	6902	7121	7193	7104	7269	7121	12 368	14 451
		200	11 324	11 743	30 136	11 712	22 149	11 743	118 046	321 874
1	S	20	2142	2643	2547	2156	2344	2243	2768	5642
		40	4063	4163	4124	4065	4166	4163	8143	10 484
	M	60	5068	5078	5108	5068	5214	5078	11 176	23 432
		80	8492	32 291	8689	8498	8684	32 291	32 626	94 168
	L	100	16 742	17 107	16 978	16 765	16 783	17 107	41 325	77 210
		200	22 143	90 136	41 134	24 231	82 131	90 136	92 683	144 498

表 4 不同工件到达率下不同算法的 CPU 时间

Tab.4 CPU time of different algorithms under different jobs arrival rates

γ	J_n	MODE_L	MODE_NL	MOEA	SPT	LPT	ABC	HPSO	
1/8	S	20	0.97	0.53	0.53	0.002	0.008	0.006	0.004
		40	0.58	0.52	0.52	0.006	0.003	0.009	0.007
	M	60	0.61	0.55	0.55	0.002	0.006	0.007	0.004
		80	0.71	0.58	0.58	0.008	0.009	0.006	0.009
	L	100	0.76	0.62	0.62	0.002	0.005	0.007	0.005
		200	0.61	0.67	0.67	0.002	0.006	0.012	0.006
1/4	S	20	1.89	0.65	0.65	0.004	0.015	0.005	0.007
		40	1.91	0.74	0.74	0.007	0.012	0.006	0.002
	M	60	1.46	0.86	0.86	0.001	0.007	0.004	0.006
		80	1.39	0.93	0.93	0.002	0.003	0.005	0.008
	L	100	1.61	0.90	0.90	0.008	0.020	0.015	0.014
		200	1.41	1.18	1.18	0.008	0.011	0.017	0.019
1/2	S	20	4.46	4.85	0.85	0.005	0.003	0.002	0.004
		40	9.1	8.03	1.03	0.005	0.002	0.004	0.001
	M	60	20.0	18.37	1.37	0.007	0.004	0.003	0.002
		80	39.1	27.88	1.88	0.006	0.006	0.003	0.005
	L	100	31.9	16.93	2.93	0.007	0.003	0.006	0.007
		200	155	163.62	3.62	0.009	0.010	0.018	0.013
1	S	20	7.3	8.94	0.94	0.002	0.002	0.006	0.006
		40	23.1	24.46	1.46	0.001	0.004	0.008	0.004
	M	60	39.4	45.84	1.84	0.005	0.005	0.007	0.003
		80	64	32.49	2.49	0.004	0.003	0.002	0.006
	L	100	88.1	52.78	2.78	0.004	0.007	0.005	0.010
		200	295	224.14	4.89	0.003	0.014	0.019	0.013

分析表 4 中数据可知,4 种调度规则所需的 CPU 时间远远小于 GA 算法、MODE_NL 算法和 MODE_L 算法所需 CPU 时间。同时,表 4 中数

据显示,调度规则虽大大节省了 CPU 时间,但其最大完工时间较大,在一定程度上降低了调度的鲁棒性和稳定性,尤其在工件密集到达($\gamma = 1/2$,

1) 和大规模工件到达 ($J_n = 100, 200$) 情况下。MODE_NL 算法没有考虑局部搜索, 所需 CPU 时间与 GA 算法接近, 均小于 MODE_L 算法。对比 4 种调度规则和其他调度算法可知, MODE_L 算法的 CPU 时间较长, 但该算法在提高调度鲁棒性和稳定性上具有明显优势。综合考虑可知, 在实际制造系统中, MODE_L 的 CPU 时间是可接受的。

4.2 工件到达率和新工件数量对调度性能的影响

为分析工件到达率和新工件数量对调度性能的影响, 参考文献[13], 首先采用方差分析 (analysis of variance, ANOVA) 方法判断以上 2 个因素是否对调度性能产生显著影响。ANOVA 用于统计显著性实验, 研究诸多控制变量中的哪些变量是对观测变量有显著影响的变量, 一般情况下, P-value 小于 0.05 表示影响是显著的。表 5、表 6 所示为统计显著性分析结果, 其中, 因素 A 表示工件到达率, B 表示新工件数量, AB 表示二者之间存在相互影响, SS 表示方差, MS 表示均方差, F 表示显著性差异的水平, P-value 表示检验水平。

表 5 调度鲁棒性的双因素方差分析

Tab.5 Two-factor variance analysis of scheduling robustness

因素	SS	MS	F	P-value
A	1159.15	10 811.31	386.95	<0.01
B	21 833.55	1560.75	55.86	<0.01
AB	21 622.63	606.12	21.69	<0.01

表 6 调度稳定性双因素方差分析

Tab.6 Two-factor variance analysis of scheduling stability

因素	SS	MS	F	P-value
A	13 122.6	6561.3	642.83	<0.01
B	621.99	310.99	30.47	<0.01
AB	1125.89	281.47	27.58	<0.01

表 5、表 6 表明, 各因素分析的 P-value 均小于 0.05, 说明工件到达率和新工件数量对调度鲁棒性和调度稳定性存在显著影响, 验证了本文前面的设想。

图 7、图 8 所示为不同工件到达率对调度稳定性和调度鲁棒性的影响。

稳定性目标为各个调度区间中参与重调度工件的开始加工时间偏差和机器偏差的累积和, 随着工件到达率 γ 的增大, 单位时间内参与重调度的工件增加, 开始加工时间偏差累积和增大, 机器偏差的累积和也增大, 从而使得调度稳定性值随着增大, 经过简单计算得出, 调度稳定性增幅在 2%~15% 之间, 说明增幅不是很明显。由图 8 可

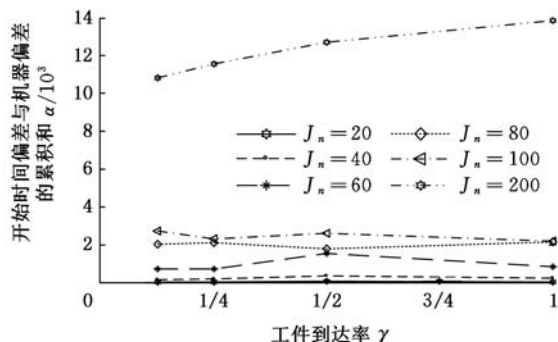


图 7 不同工件到达率下的调度稳定性结果

Fig.7 Result of schedule stability under different arrival rates of jobs

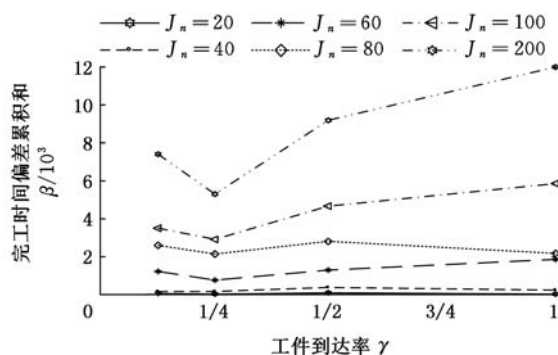


图 8 不同工件到达率下的调度鲁棒性结果

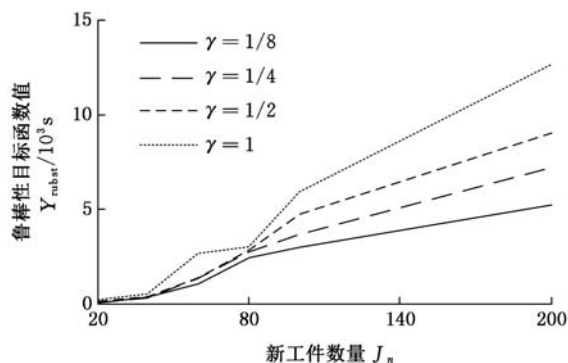
Fig.8 Result of schedule robustness under different arrival rates of jobs

知, $\gamma < 1/4$ 时, 工件较少, 此时新工件导致的扰动对调度的影响较小, 完工时间变动值呈递减趋势, 调度方案鲁棒程度增大; $\gamma > 1/4$ 时, 工件较多, 完工时间变动量成递增趋势, 调度方案鲁棒性减小。

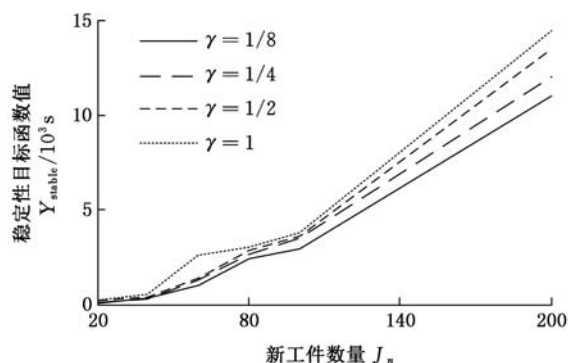
不同的工件到达率下, 新工件数量与调度性能之间的关系如图 9 所示。调度稳定性和调度鲁棒性曲线均随着新工件数量的增加近似呈直线增长。对于调度稳定性, 当新工件数量从 20 增加到 40 时, 调度稳定性以 3% 的增幅增长, 增长十分缓慢; 当新工件数量从 40 增加到 100 时, 曲线斜率变化比较明显, 其增大率近似为新工件数量 $J_n = 40$ 时的 26%; 新工件数量从 100 增加到 200 时, 由于工件密集到达, 分配到各机器上加工的工件增加, 机器负荷增大, 调度稳定性和鲁棒性降低, 曲线斜率增大。以上分析说明新工件数量的增大对调度鲁棒性和稳定性有较大影响。

5 结语

提出了柔性作业车间动态调度问题的框架, 为了保证车间生产效率与生产稳定, 建立了基于调度稳健性的动态调度模型, 结合完全重调度策略和混合周期调度与关键事件驱动重调度方法, 提出了差分变领域算法和动态事件发生器来对多



(a)鲁棒性目标函数值-新工件数量曲线



(b)稳定性目标函数值-新工件数量曲线

图9 不同新工件数下的调度稳定性与鲁棒性结果

Fig.9 Result of schedule stability and robustness under different number of new jobs

目标柔性作业车间动态调度问题进行求解。实例验证了本文提出的模型与求解方法的有效性 with 优越性。

值得注意的是,当工件密集到达时,较小的调度周期可以适当提高调度稳定性;当工件稀疏到达时,较长的调度周期在一定程度上能增强调度鲁棒性。因此在整个调度过程中采用相同的调度周期势必会影响调度性能的优劣,以后的工作是进一步研究不同调度周期对调度鲁棒性和调度稳定性的影响。

参考文献:

- [1] 潘全科,朱剑英. 作业车间动态调度研究[J]. 南京航空航天大学学报,2005,37(2):262-268.
PAN Quanke, ZHU Jianying. Rolling Time Horizon Job-Shop Scheduling Strategy in Dynamic Environment[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2005, 37(2): 262-268.
- [2] ADIBI M A, ZANDIEH M, AMIRI M. Multi-objective Scheduling of Dynamic Job Shop Using Variable Neighborhood Search[J]. Expert Systems with Applications, 2010, 37(1):282-287.
- [3] HE W, SUN D H. Scheduling Flexible Job Shop Problem Subject to Machine Breakdown with Route

Changing and Right-shift Strategies[J]. Int. J. Adv. Manuf. Technol., 2013, 66 (1):501-514.

- [4] FANG J, XI Y. A Rolling Horizon Job Shop Rescheduling Strategy in the Dynamic Environment [J]. Int. J. Adv. Manuf. Technol., 1997, 13(3): 227-232.
- [5] LI J Q, PAN Q K, FATIH TASGETIREN M. A Discrete Artificial Bee Colony Algorithm for Multi-objective Flexible Job Shop Scheduling Problems with Maintenance Activities[J]. Applied Mathematical Modelling, 2014, 38: 1111-1132.
- [6] FATTAHI P, FALLAHI A. Dynamic Scheduling in Flexible Job Shop Systems by Considering Simultaneously Efficiency and Stability[J]. CIRP Journal of Manufacturing Science and Technology, 2010, 2 (2): 114-123.
- [7] XIONG J, XING L N, CHEN Y W. Robust Scheduling for Multi-objective Flexible Job-shop Problems with Random Machine Breakdowns[J]. Int. J. Production Economics, 2013, 141: 112-126.
- [8] AL-HINAI N, ELMKAWY T Y. Robust and Stable Flexible Job Shop Scheduling with Random Machine Breakdowns Using a Hybrid Genetic Algorithm [J]. Int. J. Production Economics, 2011, 132: 279-291.
- [9] ZHANG L P, GAO L, LI X Y. A Hybrid Genetic Algorithm and Tabu Search for a Multi-objective Dynamic Job Shop Scheduling Problem[J]. International Journal of Production Research, 2013, 51 (12): 3516-3531.
- [10] CHURCH L K, UZSOY R. Analysis of Periodic and Event-driven Rescheduling Policies in Dynamic Shops[J]. International Journal of Computer Integrated Manufacturing, 1992, 5(3): 153-163.
- [11] 张超勇,李新宇,王晓娟,等.基于滚动窗口的多目标动态调度优化研究[J].中国机械工程,2009, 20 (18): 2190-2197.
ZHANG Chaoyong, LI Xinyu, WANG Xiaojuan, et al. Multi-objective Dynamic Scheduling Optimization Strategy Based on Rolling-horizon Procedure [J]. China Mechanical Engineering, 2009, 20 (18): 2190-2197.
- [12] 张利平. 作业车间预反应式动态调度理论与方法研究[D].武汉:华中科技大学,2013.
ZHANG Liping. Research on Theories and Methods for Dynamic Job Shop Scheduling Based on Predictive-reactive Scheduling [D]. Wuhan: Huazhong University of Science and Technology, 2013.
- [13] LI J Q, PAN Q K, SUGANTHAN P N, et al. A Hybrid Tabu Search Algorithm with an Efficient Neighborhood Structure for the Flexible Job Shop

- Scheduling Problem[J]. Int. J. Adv. Manuf. Technol., 2011, 52: 683-697.
- [14] KARSITI M N, CRUZ J B, MULLIGAN J H. Simulation Studies of Mutilvel Dynamic Job Shop Scheduling Using Heuristic Dispatching Rules[J]. Journal of Manufacturing Systems, 1992, 11(5): 346-358.
- [15] XUE F, SANDERSON A C, GRAVES R J. Pareto-based Multi-objective Differential Evolution [C]//Proceedings of the Congress on Evolutionary Computation. Washington D C, 2003: 862-869.
- [16] YUAN Y, XU H. Flexible Job Shop Scheduling Using Hybrid Differential Algorithms[J]. Journal of Intelligent Manufacturing, 2007, 18(3): 331-342.
- [17] LI J Q, PAN Q K, LIANG Y C. An Effective Hybrid Tabu Search Algorithm for Multi-objective Flexible Job Shop Scheduling Problems[J]. Comput. Int. Eng., 2010, 59: 647-662.
- [18] 刘炜琪, 刘琼, 张超勇, 等. 基于混合粒子群算法求解多目标混流装配线排序[J]. 计算机集成制造系统, 2011, 17(12): 2590-2598.
LIU Weiqi, LIU Qiong, ZHANG Chaoyong, et al. Hybrid Particle Swarm Optimization for Multi-objective Sequencing Problem in Mixed Model Assembly Lines[J]. Computer Intergraded Manufacturing Systems, 2011, 17(12): 2590-2598.
- [19] ZHANG G H, SHAO X Y, LI P G, et al. An Effective Hybrid Particle Swarm Optimization Algorithm for Multi-objective Flexible Job-shop Scheduling Problem[J]. Computers & Industrial Engineering, 2009, 56: 1309-1318.
- [20] 张超勇, 董星, 王晓娟, 等. 基于改进非支配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 Chaoyong, DONG Xing, WANG Xiaojuan, et al. Improved NSGA-II for the Multi-objective Flexible Job Shop Scheduling Problem [J]. Journal of Mechanical Engineering, 2010, 46(11): 156-164.

(编辑 张 洋)

作者简介:朱传平,男,1971年生。湖北工业大学机械工程学院副教授。主要研究方向为制造信息、智能调度算法、决策分析等。E-mail:zcyj2579@126.com。邝文(通信作者),女,1989年生。湖北工业大学机械工程学院硕士研究生。张超勇,男,1972年生。华中科技大学机械科学与工程学院副教授。金亮亮,男,1986年生。华中科技大学机械科学与工程学院博士研究生。

2017 年度“中国机械工业科学技术奖”申报、推荐工作的通知 2017 年度中国机械工业科学技术奖采用网络推荐、申报项目的方式(网址: <http://www.cmiao.com.cn>)。现将有关事项通知如下:

一、奖励范围 1.机械行业发明类成果;2.机械行业技术进步类成果;3.机械行业软科学、标准和检测类成果及科技图书。

二、推荐、申报办法和要求各中国机械工业专业协会、各省(自治区、直辖市)机械工业协会(联合会)、中国机械工程学会专业分会和各省(自治区、直辖市)机械工程学会等为各行业、领域和地区的推荐单位。中国机械工业联合会会员单位、机械工业领域科研院所、勘察设计院、相关大专院校、出版单位可直接申报(仅限本单位的申报项目或作为第一完成单位申报的项目)。除“直接申报单位”外,原则上其他单位均需经过推荐单位推荐。

三、推荐材料《中国机械工业科学技术奖推荐书》是中国机械工业科学技术奖励评审的主要依据,推荐书包括主件部分(第一至第十一项)、附件部分(第十二项)二部分内容。

四、公报格式文件要求申报单位请认真按照申报平台公报格式文件要求填写,并上传申报平台。内容包括单位基本情况、项目简介和能体现项目内容的彩色图片 2 张。“项目简介”主要用于获奖项目的公开宣传,不得涉及保密内容,字数应控制在 800~1000 字以内;图片格式为 JPG 或者 PDF,单张图片大小不超过 3 MB。

五、申报的其他注意事项 1.涉及国防、国家安全领域的保密项目和汽车类项目不属于中国机械工业科学技术奖评审范围;2.申报项目必须是整体技术应用一年以上(2015 年 12 月 31 日前);3.2016 年度推荐申报未授奖的项目不再受理;4.推荐申报项目一经受理不得撤销。

六、推荐时间推荐截止时间为二〇一七年三月三十日,逾期网站项目申报平台关闭不再受理;纸质材料请于四月五日前报送或快递至中国机械工业科学技术奖励工作办公室,逾期将不再受理。

七、联系方式

申报网址: <http://www.cmiao.com.cn>

地 址:中国机械工业科学技术奖励工作办公室北京市西城区百万庄大街 22 号(100037)

联系人:席敏 牛芳

电 话:010-68351688 010-68326820

传 真:010-68351688

电子邮箱: chinajixiekj@163.com