Journal of Computer Applications
计算机应用,2021,41(8):2242 - 2248
ISSN 1001-9081
CODEN JYIIDU
2021-08-10
http://www.joca.cn

文章编号:1001-9081(2021)08-2242-07
DOI:10.11772/j.issn.1001-9081.2020101664

基于改进帝国竞争算法的柔性作业车间机器故障重调度

张国辉*,陆熙熙,胡一凡,孙靖贺
(郑州航空工业管理学院 管理工程学院,郑州 450015)
(*通信作者电子邮箱zgh_hust@qq.com)

摘　要:针对机器故障下的柔性作业车间重调度问题,提出了一种改进的帝国竞争算法(ICA)。首先,以最大完工时间、机器能耗和总延迟时间为目标函数建立柔性作业车间动态重调度模型,并对三个目标采用线性加权法;然后提出了改进的ICA来把优良的信息保留到下一代,即在传统ICA的同化和革命步骤后加入一个轮盘赌的选择机制,使初始帝国中的优秀基因得以保留,并且更新后的帝国质量更优,更加贴近最优解;最后,在机器发生故障后,采用事件驱动的重调度策略对故障点后未加工的工序进行重新调度。通过生产实例,对假设的三种机器故障情景进行仿真实验,并把所提算法与改进遗传算法(GA)和遗传算法与模拟退火混合算法(GASA)这两种算法进行比较。实验结果表明了提出的改进ICA是有效且可行的。

关键词:帝国竞争算法;柔性作业车间调度问题;动态调度;机器故障;轮盘赌

中图分类号:TP183　　**文献标志码:**A

Machine breakdown rescheduling of flexible job shop based on improved imperialist competitive algorithm

ZHANG Guohui*, LU Xixi, HU Yifan, SUN Jinghe
(*College of Management Engineering, Zhengzhou University of Aeronautics, Zhengzhou Henan* 450015, *China*)

Abstract: For the flexible job shop rescheduling problem with machine breakdown, an improved Imperialist Competition Algorithm (ICA) was proposed. Firstly, a flexible job shop dynamic rescheduling model was established with the maximum completion time, machine energy consumption and total delay time as the objective functions, and linear weighting method was applied to three objectives. Then, the improved ICA was proposed to retain the excellent information for the next generation. A roulette selection mechanism was added after the assimilation and revolutionary steps of the general ICA, so that the excellent genes in the initial empire were able to be retained, and the updated empire quality was better and closer to the optimal solution. Finally, after the machine breakdown, the event-driven rescheduling strategy was adopted to reschedule the unprocessed job procedures after the breakdown point. Through production examples, simulation experiments were carried out on three hypothetical machine breakdown scenarios, and the proposed algorithm was compared with improved Genetic Algorithm (GA) and Genetic and Simulated Annealing Algorithm (GASA). Experimental results show that the proposed improved ICA is effective and feasible.

Key words: Imperialist Competitive Algorithm (ICA); Flexible Job-Shop Scheduling Problem (FJSP); dynamic scheduling; machine breakdown; roulette

0　引言

作业车间调度问题一直是制造业中的重点关注对象,车间调度的主要目的是节省产品的生产时间,让生产资源的利用率发挥最大作用,最后通过对资源的恰当分配让柔性作业车间的生产效率更高。柔性作业车间调度问题(Flexible Job-shop Scheduling Problem, FJSP)是经典车间调度的延伸,扩展了每道工序的机器柔性,相对经典车间调度来说较为繁杂。在实际的生产制造过程中,车间环境是动态变化的,会出现各种各样的问题,比如紧急插单、机器故障等不确定性因素,要充分考虑到当前工作环境及系统的状态,及时作出正确合理的调度方案,因此就出现了柔性作业车间动态重调度问题。在1977年,Nelson等[1]对于动态调度问题,提出了将动态问题转化为多个连续的静态区间,对每个区间进行优化,以达到总体最优。柴永生等[2]根据动态调度问题提出了免疫遗传算法,即对进化种群进行免疫操作,有效地解决了搜索效率低的问题。孟冠军等[3]在求解多目标柔性作业车间调度问题上设计了改进混合人工蜂群算法,即在不同的阶段采用不同的搜索机制,并且为了提高得到最优解的概率,将禁忌搜索与改进的混合人工蜂群算法结合。Shahrabi等[4]针对考虑随机作业到达和机器故障的动态作业车间调度问题,提出了一种基于

收稿日期:2020-10-27;**修回日期:**2021-01-26;**录用日期:**2021-02-01。　**基金项目:**国家自然科学基金资助项目(U1904167, 71871204, 51905494);教育部人文社科研究规划基金资助项目(18YJAZH125);河南省高校科技创新团队支持计划(21IRTSTHN018)。

作者简介:张国辉(1980—),男,河南新乡人,教授,博士,主要研究方向:智能调度;　陆熙熙(1998—),女,河南商丘人,硕士研究生,主要研究方向:智能调度;　胡一凡(1997—),男,河南驻马店人,硕士研究生,主要研究方向:智能优化算法、车间调度;　孙靖贺(1995—),男,河南沈丘人,硕士研究生,主要研究方向:智能调度、智能优化算法。

变量邻域搜索的调度方法,结合事件驱动策略,为了在任意重调度点获得合适的参数,采用了基于Q因子算法的强化学习算法,并通过模拟作业车间实验结果表明该方法的性能明显优于常用调度规则和一般可变邻域搜索。He等[5]在研究机器故障状态下的动态调度问题时,为了提高调度的鲁棒性和稳定性,首先提出了一种新的空闲时间插入策略,其次提出了加工路线变更策略和右移策略相结合的策略,在这两种策略的基础上,提出了一种多策略调度算法。邹攀等[6]提出一种基于分层蚁群遗传算法的车间资源驱动的多目标调度方法,适用于车间资源变化、任务执行情况变化、紧急任务插入等不确定情况下的动态调度。吴正佳等[7]在研究机器发生故障的柔性作业车间动态调度问题时,提出了两种重调度策略即插入重调度和完全重调度,并结合改进遗传算法(Genetic Algorithm,GA)进行求解,将机器故障产生的影响降到最低。孙丽珍等[8]采用改进遗传算法结合启发式规则求解柔性作业车间问题;针对工件加入和机器故障的动态调度,提出了加入多约束的贪婪插入空隙法的解码方式的改进遗传算法。Nouiri等[9]以最小最大完工时间和鲁棒性为目标,在求解机器故障下柔性作业车间调度问题上,提出了一种两阶段粒子群优化算法,实验部分与Al-hinai等[10]提出的混合遗传算法进行比较,结果表明该优化算法具有更好的鲁棒性和稳定性。近年来,随着可持续发展战略的提出,越来越多的生产企业开始转向绿色制造,降低能源消耗已经成了焦点问题。Nouiri等[11]在解决机器故障下的动态柔性作业车间调度问题时,以最大完工时间最小、全局能耗最小为目标,提出了一种基于粒子群优化方法的绿色重调度方法,可以最大限度地减少时间和降低能耗。包哲人等[12]面向考虑能耗的柔性作业车间调度问题提出了改进离散蝙蝠算法,为避免算法早熟收敛设计了一种具有记忆能力的粒子变异操作,同时为了算法的搜索能力加入惯性权重策略,通过生产实例验证了算法的有效性。陈超等[13]以平均流经时间和能耗为目标建立柔性作业车间模型,针对此模型提出了遗传算法与模拟退火混合算法(Genetic and Simulated Annealing Algorithm, GASA),采用滚动窗口技术结合GASA来求解机器故障的动态调度问题并验证了算法的有效性。曹庆奎等[14]针对在机器故障情况下,同时考虑能耗和交货期的柔性作业车间动态调度问题,采用GASA进行求解,以最大完工时间最小、能耗最小、客户满意度最大为目标验证了其算法的有效性。李明等[15]针对考虑准备时间的低碳FJSP,提出了新型帝国竞争算法(Imperialist Competitive Algorithm, ICA),采用一种全新的同化方法和归一化总成本新定义,在优化关键目标的同时持续改进非关键目标总能耗。吕聪等[16]提出协作混合帝国算法,改进了自适应参数,提高了算法的收敛速度,并对不同阶段采用多变异改革策略以提高算法的局部搜索效率,以及协作交流机制来提高算法的全局搜索能力,表明了协作混合帝国算法在求解柔性车间调度问题上的稳定性和优越性。以上针对动态调度问题的研究,在国内外学者的努力下研究成果丰富,但是针对机器故障重调度,并同时考虑能耗的研究相对较少。机器故障在实际车间生产过程中是很常见的动态事件,在发生故障后找到合适的重调度方案是非常重要的。本文主要研究机器故障的柔性作业车间重调度问题,对帝国竞争算法进行改进,在同化和革命操作之后加入一个轮盘赌的选择机制,对同化和革命后的个体重新进行选择,并结合事件驱动策略对此类重调度问题进行求解,验证算法的有效性。

1　问题描述

柔性作业车间调度问题可以描述为:将1到n个工件合理安排到m台机器上进行加工,每个工件有多道工序,其中每道工序都可以选择不同的机器进行加工,各工序所选机器不同,所需加工时长也不相同。在实际生产过程中存在一些不确定性因素,导致会出现各种动态事件,本文考虑的是机器故障的动态事件。柔性作业车间动态重调度研究存在以下几个假设条件:

1)同一个工件在同一时刻只能由一台机器进行加工,一台机器在同一时刻只允许有一个工件在该机器上加工。

2)每一道工序可以选择由可加工机器集中的机器进行加工,但是同一时刻只允许在一台机器上加工。

3)每一个工件对于每道工序的加工有先后顺序要求,只有某工件的前道工序加工完成后才能对该道工序进行加工。

4)当原调度已经对工件的前几道工序加工完成,再调度时只需加工该工件的后续工序。

柔性作业车间机器故障重调度问题是在给出相关变量和决策变量、满足约束条件和目标函数的情况下,生成一个初始调度,在机器发生故障后,初始调度方案便不可行,此时应重新调整调度方案,结合重调度策略生成重调度方案。

1.1　变量定义

本文的相关变量符号及定义如表1所示。

表1　变量描述

Tab. 1　Variables description

变量	定义
i	工件数量,i=1,2,…,n
j	工序号,j=1,2,3,…
k	机器序号,k=1,2,…,m
O_{ij}	工件J_i的第j道工序
m_{ij}	工序O_{ij}可选择的加工机器数
C_i	工件J_i的完工时间
E	加工总能耗
E_{ijk}	工件J_i在机器k上单位时间内的加工能耗
S_{ijk}	工件J_i在机器k上的开始加工时间
P_{ijk}	工件J_i在机器k上的加工时间
C_{ijk}	工件J_i在机器k上的加工结束时间
∂_{ijk}	工件J_i的第j道工序在机器k上进行加工
$C_{\max}^1$	机器故障前调度的最大完工时间
$C_{\max}^2$	机器故障后调度的最大完工时间
M	无穷大的正数

本文的决策变量有∂_{ijk}和δ_{ijhdk},其中,∂_{ijk}用于确定工序与机床的选择关系;δ_{ijhdk}用于确定工序在机器上的优先加工顺序。

$$\partial_{ijk}=\begin{cases}1, & \text{工序}O_{ij}\text{在机器}k\text{上加工}\\0, & \text{工序}O_{ij}\text{不在机器}k\text{上加工}\end{cases}$$

$$\delta_{ijhdk}=\begin{cases}1, & \text{工序}O_{ij}\text{在机器}k\text{上优先于}O_{hd}\text{加工}\\0, & \text{工序}O_{ij}\text{在机器}k\text{上不优先于}O_{hd}\text{加工}\end{cases}$$

1.2　约束条件

柔性作业车间调度问题要满足工件约束和机器约束:

$$\sum_{k=1}^{m_{ij}} \partial_{ijk} = 1 \tag{1}$$

$$C_{i(j+1)h} - p_{i(j+1)h} + M\left(1 - \partial_{i(j+1)h}\right) \geqslant C_{ijk} \tag{2}$$

$$S_{ijk} + p_{ijk} \leqslant S_{hdk} + M\left(1 - \delta_{ijhdk}\right) \tag{3}$$

式(1)表示一道工序只能在一台机器上加工;式(2)表示同一工件只能在前一道工序加工完才能加工后一道工序;式(3)表示一台机器在同一时刻只能加工一道工序。

1.3 目标函数

本文主要研究的调度目标为最小化最大完工时间(makespan)、最小能耗以及最小总延迟时间,如下所示:

1)最小化最大完工时间。

$$\min C_{\max} = \min\left[\max\left(\sum_{i=1}^{n} C_i\right)\right] \tag{4}$$

2)最小能耗。

采用文献[13]中的加工能耗E的表示方法,每台机器上的第一个工件的开始加工时刻就是该台机器的开工时刻,机器一直保持着耗能状态,直到该机器上所有工件都加工完毕,因此能耗的数学模型为式(5)所示:

$$\min E = \sum_{k=1}^{m} \sum_{i=1}^{n} E_{ijk} \cdot P_{ijk} \cdot \partial_{ijk} \tag{5}$$

3)最小总延迟时间。

本文总延迟时间主要是计算机器故障前后,两次调度的最小化最大完工时间差,总延迟时间的数学模型如式(6)所示:

$$\min D = C_{\max}^2 - C_{\max}^1 \tag{6}$$

由于三个目标的量纲不统一,需要进行归一化处理,将各目标数据转换到[0,1]内,计算过程如式(7)所示:

$$x^* = \left|\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right| \tag{7}$$

其中:x^*表示归一化之后的目标值,x为当前目标值,$x_{\max}$为全局目标最大值,$x_{\min}$为全局目标最小值。

对本文的三个目标进行归一化处理后,再采用线性加权和法,将三个目标优化问题转变成单目标优化问题,如式(8)所示:

$$\min \theta = aC_{\max}^* + bE^* + cD^* \tag{8}$$

其中:$C_{\max}^*$、E^*、D^*分别为归一化之后的最大完工时间、能耗和总延迟时间的数值,a、b、c分别表示最大完工时间、能耗、总延迟时间系数。

2 FJSP动态重调度策略

动态重调度策略,是指在系统生产过程突发情况下将采取应对措施来解决发生的异常事件。一般用到的策略方法有三种,分为事件驱动、周期驱动和周期与事件的混合驱动,本文采用事件驱动策略。

当机器出现故障时,确定机器修复时间T,故障点前已经加工过的工序保持不动,找出故障点后未加工的工序,并对其用帝国竞争算法进行重新调度,故障机器的开工时间为修复后的时间,即故障时刻加上故障时长,其余机器的开工时间为当前正在加工工序的完成时间。机器发生故障后,在该机器上未进行加工的工件将被分配到其他机器上进行加工。机器故障重调度的具体流程如图1所示。

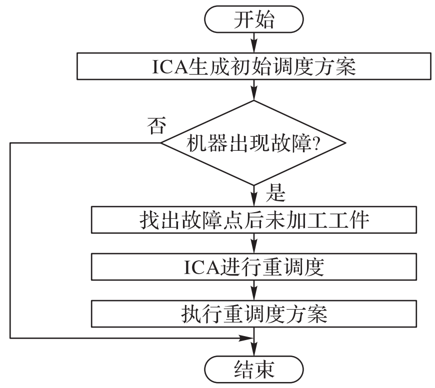

图1 机器故障重调度流程

Fig. 1 Flow of rescheduling with machine breakdown

3 算法设计

3.1 基本帝国竞争算法

帝国竞争算法是一种受帝国竞争行为启发的新的智能优化算法,它与粒子群优化(Particle Swarm Optimization, PSO)算法、蚁群优化(Ant Colony Optimization, ACO)算法等一样,都属于基于群体的随机优化搜索算法。帝国竞争算法比PSO和GA收敛速度快,具有较强的全局收敛性,并且同化和革命步骤又保证了算法的局部搜索能力,因此在求解柔性作业车间调度问题上有一定的优势。帝国竞争算法在解决各种实际的优化问题上具有广泛应用,本文将其应用在解决机器故障重调度问题上。

ICA的基本过程为:产生初始帝国、同化、革命、帝国竞争和帝国消亡。ICA的流程如图2所示,具体步骤为:

步骤1 将随机产生的N_{pop}个初始解作为初始国家,并计算每个国家的成本c_n。

步骤2 构建初始帝国。选择成本较小的N_{imp}个国家作为殖民国家$\boldsymbol{imp}$,剩余的作为殖民地$\boldsymbol{Col}$,殖民地的个数N_{Col}与殖民国家个数N_{imp}之和等于初始解的个数N_{pop}。确定殖民国家所能分到的殖民地数量,随机分配殖民地给各殖民国家,此时各殖民国家与其殖民地组成初始帝国$\boldsymbol{PEmp}$。殖民国家所拥有的殖民地数量与其势力成正比,因此要先计算每个殖民国家的标准化成本C_n,如式(9)所示:

$$C_n = c_n - \max_i \{c_i\} \tag{9}$$

殖民国家的成本越小,势力就越大,每个殖民国家的标准化势力p_n如式(10)所示:

$$p_n = \left|C_n \Big/ \sum_{i=1}^{N_{imp}} C_i\right| \tag{10}$$

第n个殖民国家所拥有的初始殖民地数量NC_n如式(11)所示:

$$NC_n = \mathrm{round}\left\{p_n \times N_{Col}\right\} \tag{11}$$

其中round是将小数四舍五入成整数的函数。

步骤3 同化和革命。将殖民地向其所在殖民国家移动,即殖民地都趋向于殖民国家这个最优解;革命的目的是防止同化后的算法过早收敛。同化操作分为机器的选择和工序排序两部分,机器的选择部分采用多点交叉变异的方法,具体操作为:随机选取r个位置,将殖民国家的r个位置的信息替换到殖民地中的对应位置,其余位置信息保持不变;工序排序部分采用工件交换的方式,将殖民地的加工工序部分与殖民国家的加工工序部分进行信息交换,更新殖民地的加工工序

部分。革命操作同样分为机器的选择部分和工序排序部分，在机器选择部分，随机选取某个加工机器，然后选择对应机器集中的其他任一机器代替它；在工序排序部分，随机选择两个位置上的工序进行信息交换。同化和革命后的新帝国为 ***SEmp***。

步骤 4　殖民国家更新。当殖民地经过同化和革命后，将殖民国家和所有殖民地的成本进行比较：若有殖民地的成本比殖民国家要小，则该殖民地替代殖民国家成为该帝国内新的殖民国家；否则，殖民国家不发生变化，更新后的帝国为 ***Emp***。

步骤 5　帝国竞争。计算所有帝国的总成本，较弱帝国需选出一个最弱殖民地，较强帝国可以来争夺殖民地，帝国越强获得该殖民地的概率越大。每个帝国的总成本 TC_n 计算如式(12)所示，将每个帝国的总成本进行标准化处理，如式(13)所示，每个帝国占有弱殖民地的概率 P_n 计算如式(14)所示：

$$TC_n = f\left(\boldsymbol{imp}_n\right) + \alpha \times \frac{\sum_{i=1}^{NC_n} f\left(\boldsymbol{Col}_i\right)}{NC_n} \tag{12}$$

$$NTC_n = TC_n - \max_i \{TC_i\} \tag{13}$$

$$P_n = \left| NTC_n \Big/ \sum_{i=1}^{N_{\mathrm{imp}}} NTC_i \right| \tag{14}$$

其中：$f(\boldsymbol{imp}_n)$ 表示第 n 个帝国的殖民国家的成本，$f(\boldsymbol{Col}_i)$ 表示第 i 个殖民地的成本，α 为殖民地影响因子，$0<\alpha<1$，NTC_n 表示第 n 个帝国的标准化成本。

步骤 6　帝国消亡。当某个帝国没有任何殖民地时，此时该帝国消亡。

步骤 7　若终止条件成立，则算法结束；否则回到步骤 3。

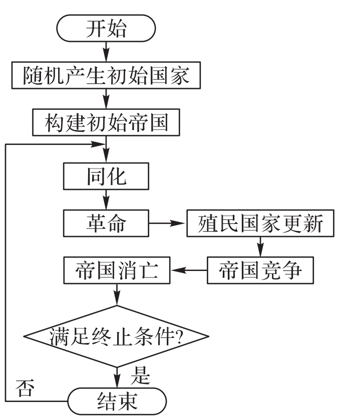

图 2　ICA 流程

Fig. 2　Flowchart of ICA

3.2　编码和解码

在 ICA 中，每一个国家代表一组柔性作业车间动态调度问题的可行方案，结合柔性作业车间调度问题特点设计两段式实数编码方式对国家个体进行编辑。

以图 3 为例，每个个体为一串整数，总长度为 $2L$，L 为所有工序的个数，O_i 表示工件 i 的工序数。左半部分为加工机器的选择，按照工件 1、工件 2 和工件 3 的工序顺序进行排列，每个整数表示对应工序可选加工机器的排列序号，第一个整数 3 为工序 O_{11} 可选机器集 $\{M_2, M_3, M_4\}$ 的第 3 个机器即 M_4。右半部分为加工工序的排序，每个整数代表工件号，第 1 次出现表示该工件的第 1 道工序，第 2 次出现表示该工件的第 2 道工序，以此类推，图 3 中加工工序所表示的工序顺序为 $(O_{21}, O_{11}, O_{31}, O_{12}, O_{32}, O_{22})$。

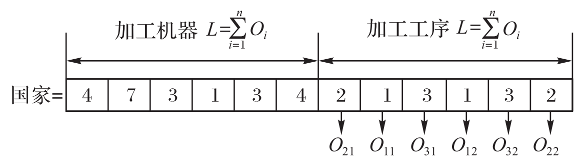

图 3　编码表达示意图

Fig. 3　Schematic diagram of coding expression

对国家个体进行解码操作时，首先对个体的前半部分即机器部分从左到右依次进行解码，转成对应的机器并确定各工序在对应机器上的加工时间，然后对个体的后半部分即工序部分进行解码，解码成活动调度。

3.3　改进 ICA

帝国竞争算法中的同化是通过殖民地向殖民国家的移动来实现，相当于是一个寻优过程增强了算法的局部搜索能力，但是在传统的帝国竞争算法中并没有物竞天择的进化规律，在 ICA 求解调度时为了能够提高求解质量、提高算法的寻优能力，因此本文参照遗传算法中的遗传和变异对帝国竞争算法进行改进，在初始帝国进行同化和革命之后增加一个选择机制的步骤，可以使初始帝国中的优秀基因得以保留，更新后的帝国质量更优，更加贴近最优解。选择机制的具体操作如下：

步骤 1　首先将初始帝国 ***PEmp*** 中的第一个帝国 ***PEmp***(1)的所有个体与其同化和改革后的 ***SEmp*** 中对应的第一个帝国 ***SEmp***(1)中的所有个体放在一起。

步骤 2　采用轮盘赌的方法从合并的帝国中选出一半数量的个体，将其作为一个新的帝国 ***Emp***(1)。本文采用轮盘赌的个体概率值为各个国家的成本值的倒数，成本值越小则被选到的概率越大，否则概率越小。

步骤 3　从新帝国中选出成本值最低的国家作为殖民国家，剩余的国家则沦为该殖民国家的殖民地。

步骤 4　接着转到步骤 1 进行初始帝国 ***PEmp*** 中的第二个帝国 ***PEmp***(2)和同化和改革后的 ***SEmp*** 中对应的第二个帝国 ***SEmp***(2)的合并，以此类推，直到所有帝国全部选择完成，最终生成新的帝国 ***Emp***，此时选择机制结束。

改进帝国竞争算法选择机制的示意图如图 4 所示。

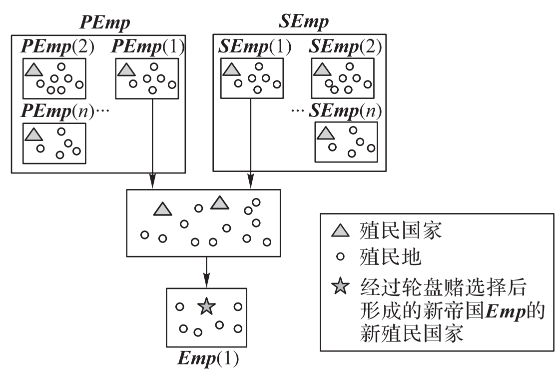

图 4　选择机制示意图

Fig. 4　Schematic diagram of selection mechanism

改进后的帝国竞争算法的流程如图5所示。

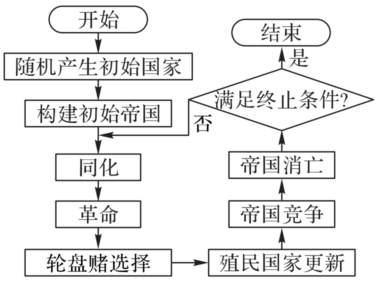

图5 改进ICA流程

Fig. 5 Flow of improved ICA

4 仿真结果与分析

采用改进ICA对柔性作业车间机器故障重调度算例进行仿真实验，并随机假设三种机器发生故障的场景，得出最佳调度方案，验证该算法求解柔性作业车间机器故障重调度问题的有效性。

4.1 参数设置

本文的实验算例来自洛阳某钢具厂生产车间，经过简化为8×8的柔性作业调度问题，如表2所示，有8个工件，8台机器。表3为每台机器的加工能耗，表4为假设的三种机器故障场景，这三种场景分别独立发生。

表2 仿真实例数据

Tab. 2 Data of simulation examples

工件序号	工序序号	可选机器集	加工时间/min
J_1	O_{11}	1,2,3,4,5,7,8	5,3,5,3,3,10,9
	O_{12}	1,3,4,5,6,7,8	10,5,8,3,9,9,6
	O_{13}	2,4,5,6,7,8	10,5,6,2,4,5
J_2	O_{21}	1,2,3,4,5,7	5,7,3,9,8,9
	O_{22}	2,3,4,5,6,7,8	8,5,2,6,7,10,9
	O_{23}	2,4,5,6,7,8	10,5,6,4,1,7
	O_{24}	1,2,3,4,6,8	1,4,5,6,10,7
J_3	O_{31}	1,4,5,6,7,8	10,7,6,5,2,4
	O_{32}	2,3,4,5,6,7	10,6,4,8,9,10
	O_{33}	1,2,3,4,6,8	1,4,5,6,10,7
J_4	O_{41}	1,2,3,4,5,6,7,8	3,1,6,5,9,7,8,4
	O_{42}	1,2,3,4,5,6,7,8	12,11,7,8,10,5,6,9
	O_{43}	1,2,3,4,5,6,7,8	4,6,2,10,3,9,5,7
J_5	O_{51}	1,2,3,4,5,7	3,6,7,8,9,10
	O_{52}	1,3,4,5,6,7	10,7,4,9,8,6
	O_{53}	2,3,4,5,6,7	9,8,7,4,2,7
	O_{54}	1,2,4,5,6,7,8	11,9,6,7,5,3,6
J_6	O_{61}	1,2,3,4,5,6,8	6,7,1,4,6,9,10
	O_{62}	1,3,4,5,6,7,8	11,9,9,9,7,6,4
	O_{63}	1,2,3,4,5,7	10,5,9,10,11,10
J_7	O_{71}	1,2,3,4,5,7	5,4,2,6,7,10
	O_{72}	2,3,4,5,6,8	9,9,11,9,10,5
	O_{73}	2,3,4,5,6,8	8,9,3,8,6,10
J_8	O_{81}	1,2,3,4,6,8	2,8,5,9,4,10
	O_{82}	1,2,3,4,5,7	7,4,7,8,9,10
	O_{83}	1,2,4,5,6,7,8	9,9,8,5,6,7,1
	O_{84}	1,3,4,5,6,7	9,3,7,1,5,8

使用Matlab 2017b对该算例进行仿真实验，本文提出的算法参数设定为：最大迭代次数为50，国家个数为100，帝国个数为10，a、b、c分别设为0.3、0.3、0.4。针对同化系数、革命概率和竞争概率三个参数，采用田口方法以及Minitab软件进行多因子正交实验，正交表如表5所示，每个参数有3个水平，共有9种不同的组合，对该正交表进行田口分析得到的均值主效应图如图6所示，最后得出同化系数、革命概率和竞争概率的参数分别为0.7、0.05、1。

表3 机器能耗

Tab. 3 Machine energy consumption

机器	能耗/(J·min^{-1})	机器	能耗/(J·min^{-1})
1	4.5	5	1.2
2	1.6	6	5.4
3	7.5	7	6.6
4	1.7	8	9.7

表4 三种机器故障场景

Tab. 4 Three scenarios of machine breakdowns

场景	故障机器	故障时刻$t1$	修复时刻$t2$
1	3	0	15
2	6	5	10
3	7	10	25

表5 正交表

Tab. 5 Orthogonal table

序号	同化系数	革命概率	竞争概率	目标值
1	0.5	0.05	0.2	0.1262
2	0.5	0.10	0.5	0.1448
3	0.5	0.15	1.0	0.1249
4	0.7	0.05	0.5	0.1185
5	0.7	0.10	1.0	0.1178
6	0.7	0.15	0.2	0.1366
7	0.9	0.05	1.0	0.1141
8	0.9	0.10	0.2	0.1461
9	0.9	0.15	0.5	0.1471

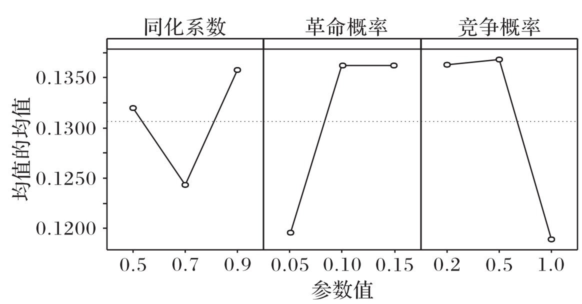

图6 参数均值主效应图

Fig. 6 Main effect plots of parameter means

4.2 实验结果

首先在初始时刻采用改进ICA对问题进行调度，得到初始方案，如图7所示为迭代最优解的甘特图，最大完工时间为26 min，能耗为624.3 J。图7中相同颜色代表同一工件，其中颜色块上的数字表示为：第一个数字是工件号，后两位数字表示工序号，如“101”表示工件1的第一道工序，“402”表示工件4的第二道工序。

场景1是机器3在0时刻发生故障，故障修复时刻为15，进行重调度后的甘特图如图8所示，最大完工时间为27 min，总延迟时间为1 min，能耗为722.8 J。图8中数字表示工件号，相同颜色代表同一工件，机器3上，[0,15]时间段为故障区间。

场景2是机器6在5时刻发生故障,故障修复时刻为10,进行重调度后的甘特图如图9所示,最大完工时间为27 min,总延迟时间为1 min,能耗为757.6 J。图9中标有数字的白色部分表示机器发生故障时刻之前的工序保持原调度不变,有颜色的部分表示机器发生故障后重新进行调度的工序。

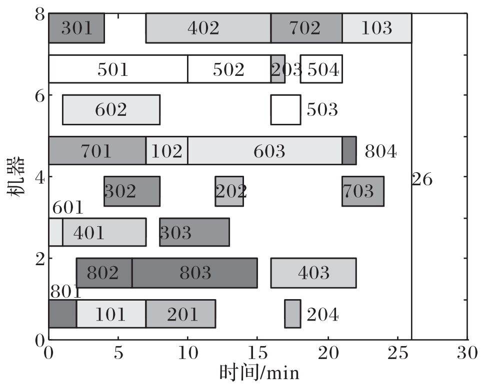

图7　初始调度方案甘特图

Fig. 7　Gantt chart of initial scheduling scheme

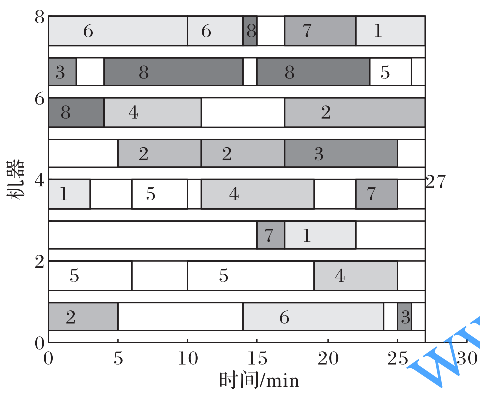

图8　机器3发生故障后的重调度甘特图

Fig. 8　Gantt chart of rescheduling after machine 3 breakdown

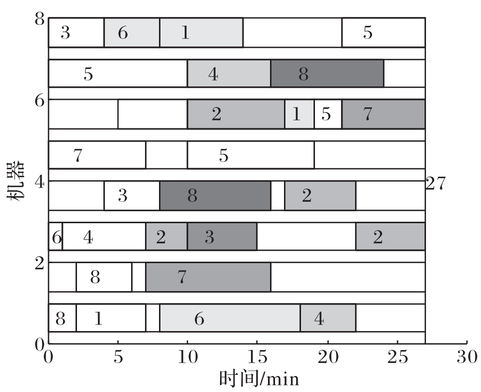

图9　机器6发生故障后的重调度甘特图

Fig. 9　Gantt chart of rescheduling after machine 6 breakdown

场景3是机器7在10时刻发生故障,故障修复时刻为25,重调度后的甘特图如图10所示,最大完工时间为27 min,总延迟时间为1 min,能耗为707.3 J。

为进一步验证改进ICA的有效性,假设机器2、机器5和机器6分别在0时刻发生故障,并且短时间内无法修复,针对该三种机器故障情况采用GASA[13]、改进GA[8]和改进ICA分别进行求解并对比,三种算法求出的三种优化目标数据如表6所示,从表中可以看出该改进ICA与其他两种算法相比实验结果更好,验证了该算法的有效性。

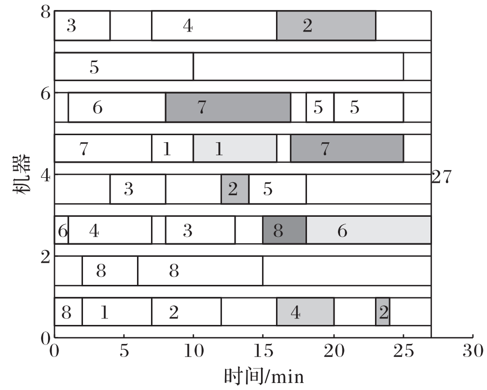

图10　机器7发生故障后的重调度甘特图

Fig. 10　Gantt chart of rescheduling after machine 7 breakdown

表6　实验数据对比

Tab. 6　Comparison of experimental data

故障机器	算法	$C_{\max}$/min	E/J	D/min
2	GASA	28	786.1	2
	改进GA	31	792.5	5
	改进ICA	27	712.2	1
5	GASA	30	807.9	4
	改进GA	30	833.9	4
	改进ICA	27	740.6	1
6	GASA	29	780.4	3
	改进GA	32	817.9	6
	改进ICA	29	723.6	3

为了验证算法求解的稳定性,分别采用三种算法对上述机器2发生故障的情况进行求解,表7为三种算法运行10次的各优化目标的平均值和方差,从表7中可以看出,改进ICA与GASA、改进GA相比各优化目标的方差最小,表明该改进ICA求解的稳定性最好。

表7　三种算法的性能对比

Tab. 7　Performance comparison of three algorithms

算法	数据类型	$C_{\max}$/min	E/J	D/min	运行时间/s
GASA	平均值	31.70	798.89	5.70	10.301 1
	方差	6.01	3 114.15	6.01	2.440 5
改进GA	平均值	32.70	795.63	6.70	1.380 2
	方差	4.81	3 252.06	4.81	0.098 7
改进ICA	平均值	29.80	758.82	3.80	1.977 0
	方差	1.76	1 508.90	1.76	0.005 6

针对上述假设的机器故障情况的实验研究以及与遗传算法进行比较,可以得出的结论是,故障后的重调度方案的最大完工时间、总延迟时间以及能耗都控制在一个较好的范围之内,符合实际生产需要。

5　结语

本文针对柔性作业车间机器故障的动态事件,对帝国竞争算法进行改进,加入选择机制操作,结合事件驱动策略以及最大完工时间、能耗和总延迟时间三个目标函数,对该类动态问题进行求解。对提出的三种不同机器故障的场景进行实验

仿真分析,得出的结果较为符合实际生产需要,并将改进ICA与两种不同的算法进行比较,验证了该算法的有效性和可行性。

柔性作业车间生产过程中容易受到各种复杂因素的影响,下一步将继续对帝国竞争算法进行改进,并运用到其他类型的动态调度中,如插入紧急订单、工件延误等动态事件。

参考文献(References)

[1] NELSON R T, HOLLOWAY C A, WONG R M L. Centralized scheduling and priority implementation heuristics for a dynamic job shop model with due dates and variable processing time [J]. AIIE Transactions, 1977, 9(1): 95-102.

[2] 柴永生,孙树栋,余建军,等. 基于免疫遗传算法的车间动态调度[J]. 机械工程学报, 2005, 41(10): 23-27. (CHAI Y S, SUN S D, YU J J, et al. Dynamic shop scheduling based on immune genetic algorithm[J]. Journal of Mechanical Engineering, 2005, 41(10): 23-27.)

[3] 孟冠军,杨大春,陶细佩. 基于混合人工蜂群算法的多目标柔性作业车间调度问题研究[J]. 计算机应用研究, 2019, 36(4): 972-974, 979. (MENG G J, YANG D C, TAO X P. Study on multi-objective flexible job-shop scheduling problem based on hybrid artificial bee colony algorithm [J]. Application Research of Computers, 2019, 36(4): 972-974, 979.)

[4] SHAHRABI J, ADIBI M A, MAHOOTCHI M. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling[J]. Computers and Industrial Engineering, 2017, 110: 75-82.

[5] HE W, SUN D H. Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies [J]. The International Journal of Advanced Manufacturing Technology, 2013, 66(1/2/3/4):501-514.

[6] 邹攀,李蓓智,杨建国,等. 基于分层蚁群遗传算法的多目标柔性作业车间调度方法[J]. 中国机械工程, 2015, 26(21): 2873-2879, 2884. (ZOU P, LI B Z, YANG J G, et al. Hierarchical ant-genetic algorithm-based multi-objective intelligent approach for flexible job shop scheduling[J]. China Mechanical Engineering, 2015, 26(21): 2873-2879, 2884.)

[7] 吴正佳,何海洋,黄灿超,等. 带机器故障的柔性作业车间动态调度[J]. 机械设计与研究, 2015, 31(3): 94-98. (WU Z J, HE H Y, HUANG C C, et al. Flexible job shop dynamic scheduling problem research with machine fault [J]. Machine Design and Research, 2015, 31(3): 94-98.)

[8] 孙丽珍,毕利. 多约束条件下的动态柔性作业车间调度研究[J]. 控制工程, 2020, 27(11): 1921-1929. (SUN L Z, BI L. Dynamic flexible job shop scheduling problem under multiple constraints[J]. Control Engineering of China, 2020, 27(11): 1921-1929.)

[9] NOUIRI M, BEKRAR A, JEMAI A, et al. Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns[J]. Computers and Industrial Engineering, 2017, 112: 595-606.

[10] Al-HINAI N, ElMEKKAWY T Y. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm [J]. International Journal of Production Economics, 2011, 132(2): 279-291.

[11] NOUIRI M, BEKRAR A, TRENTESAUX D. Towards energy efficient scheduling and rescheduling for dynamic flexible job shop problem[J]. IFAC-PapersOnLine, 2018, 51(11): 1275-1280.

[12] 包哲人,徐华. 面向能耗机制的多目标柔性作业车间调度[J]. 计算机应用研究, 2017, 34(12): 3617-3622. (BAO Z R, XU H. Multi-objective flexible Job-Shop scheduling based on energy-consumption mechanism[J]. Application Research of Computers, 2017, 34(12): 3617-3622.)

[13] 陈超,王艳,严大虎,等. 面向能耗的柔性车间动态调度研究[J]. 系统仿真学报, 2017, 29(9): 2168-2174, 2181. (CHEN C, WANG Y, YAN D H, et al. Research on dynamic flexible job shop scheduling problem for energy consumption[J]. Journal of System Simulation, 2017, 29(9): 2168-2174, 2181.)

[14] 曹庆奎,张晓丽,任向阳. 基于GASA的柔性作业车间动态调度研究[J]. 河北工程大学学报(自然科学版), 2019, 36(2): 91-96. (CAO Q K, ZHANG X L, REN X Y. Research on dynamic scheduling of flexible job shop based on GASA [J]. Journal of Hebei University of Engineering (Natural Science Edition), 2019, 36(2): 91-96.)

[15] 李明,雷德明. 考虑准备时间和关键目标的柔性作业车间低碳调度研究[J]. 机械工程学报, 2019, 55(21): 139-149. (LI M, LEI D M. Research on flexible job shop low carbon scheduling with setup times and key objectives [J]. Journal of Mechanical Engineering, 2019, 55(21): 139-149.)

[16] 吕聪,魏康林. 柔性车间调度问题的协作混合帝国算法[J]. 计算机应用, 2018, 38(7): 1882-1887. (LYU C, WEI K L. Cooperative hybrid imperialist competitive algorithm for flexible job-shop scheduling problem [J]. Journal of Computer Applications, 2018, 38(7): 1882-1887.)

This work is partially supported by the National Natural Science Foundation of China (U1904167, 71871204, 51905494), the Humanities and Social Sciences Planning Fund of Ministry of Education (18YJAZH125), the Supporting Project of the Innovative Research Team in Colleges and Universities of Henan Province (21IRTSTHN018).

ZHANG Guohui, born in 1980, Ph. D., professor. His research interests include intelligent scheduling.

LU Xixi, born in 1998, M. S. candidate. Her research interests include intelligent scheduling.

HU Yifan, born in 1997, M. S. candidate. His research interests include intelligent optimization algorithm, job-shop scheduling.

SUN Jinghe, born in 1995, M. S. candidate. His research interests include intelligent scheduling, intelligent optimization algorithm.