



(12) 发明专利

(10) 授权公告号 CN 112926837 B

(45) 授权公告日 2023. 09. 12

(21) 申请号 202110153601.6

(22) 申请日 2021.02.04

(65) 同一申请的已公布的文献号
申请公布号 CN 112926837 A

(43) 申请公布日 2021.06.08

(73) 专利权人 郑州轻工业大学
地址 450002 河南省郑州市高新技术产业
开发区科学大道136号

(72) 发明人 乔东平 柏文通 段绿旗 王雅静
肖艳秋 文笑雨 李浩 李立伟
孙春亚 张玉彦 王昊琪

(74) 专利代理机构 重庆市信立达专利代理事务
所(普通合伙) 50230
专利代理师 陈炳萍

(51) Int. Cl.

G06Q 10/0631 (2023.01)

G06Q 50/04 (2012.01)

G06N 3/126 (2023.01)

(56) 对比文件

CN 111190401 A, 2020.05.22

CN 111325443 A, 2020.06.23

CN 108805403 A, 2018.11.13

US 2018356803 A1, 2018.12.13

CN 111079987 A, 2020.04.28

乔东平等. 基于关联规则的作业车间调度问题改进遗传算法研究. 《河南理工大学学报(自然科学版)》. 2022, 全文.

审查员 徐颖

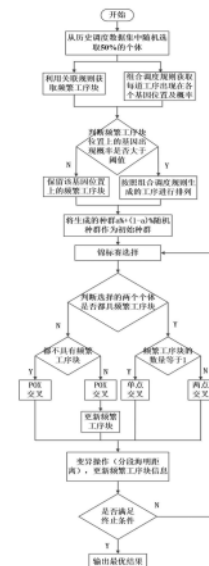
权利要求书2页 说明书7页 附图5页

(54) 发明名称

基于数据驱动改进遗传算法求解作业车间调度问题的方法

(57) 摘要

本发明提供一种数据驱动改进遗传算法 (AR-GA) 解决作业车间调度问题的方法, 有效的解决了传统遗传算法求解作业车间调度问题时, 受限于初始种群质量而存在收敛速度慢、易陷入局部最优解等问题。该算法的改进之处在于: 在遗传算法种群初始化阶段借助关联规则和组合调度规则获取基因序列中的频繁工序块以及每道工序出现各个基因位置及概率; 在交叉阶段根据频繁工序块在待交叉种群中的分布设计了三种不同的交叉方式; 变异过程中结合分段海明距离引导子代种群变异, 并且在每次迭代后更新频繁工序块的信息。本发明提出的改进算法不仅在求解质量上具有优势, 在提高求解效率、稳定性的作业车间调度问题也具有一定的优势。



CN 112926837 B

1. 基于数据驱动改进遗传算法求解作业车间调度问题的方法, 其特征在于, 包括以下步骤:

步骤一: 从历史调度数据集中随机选取若干个体;

步骤二: 利用关联规则获取频繁工序块, 同时组合调度规则获取每道工序出现在各个基因的位置及概率;

步骤三: 预先设置一阈值, 判断频繁工序块位置上的基因出现概率是否大于阈值, 如果是, 执行步骤四; 如果否, 执行步骤五;

步骤四: 按照组合调度规则生成的工序进行排列, 并执行步骤六;

步骤五: 保留该基因位置上的频繁工序块, 并执行步骤六;

步骤六: 将生成的种群 $a\% + (1-a)\%$ 随机种群作为初始种群;

步骤七: 进行锦标赛选择;

步骤八: 判断选择的两个个体是否都具频繁工序块, 如果是, 执行步骤九; 如果否, 执行步骤十;

步骤九: 判断待交叉个体中频繁工序块的数量是否等于1, 如果是, 执行单点交叉, 如果否, 执行两点交叉; 然后执行步骤十一;

步骤十: 判断两个个体是否都不具有频繁工序块, 如果是, 执行POX交叉; 如果否, 执行POX交叉, 并更新频繁工序块; 然后执行步骤十一;

步骤十一: 进行变异操作, 更新频繁工序块信息;

步骤十二: 判断是否满足终止条件, 如果是, 输出最优结果; 如果否, 重复步骤七至步骤十二; 终止条件为运算循环的次数达到迭代次数 N_{iter} 时停止;

所述步骤一中, 从历史调度数据集中随机选取40-60%的个体;

所述步骤二中, 调度规则包括基于工序的加工时间 $R_{a,i}$, 基于工件的剩余加工时间 $R_{b,i}$, 基于工件的剩余加工工序数 $R_{c,i}$ 和基于机器的加工时间 $R_{d,i}$, 调度规则的线性组合参数表现形式为Power;

基于工序的加工时间的具体调度规则如下:

$$R_{a,i} = 1 - \frac{t_{\max} - t_i^k}{t_{\max} - t_{\min}} \quad (1)$$

其中: t_{\max} 和 t_{\min} 分别表示工序的最大加工时间和最小加工时间, t_i^k 表示工件i的第k道工序的加工时间, $R_{a,i}$ 表示工件i在规则a下的优先值;

基于工件的剩余加工时间的具体调度规则如下:

$$R_{b,i} = 1 - \frac{f_{\max} - f_i}{f_{\max} - f_{\min}} \quad (2)$$

其中: f_{\max} 和 f_{\min} 分别表示待加工工件的最大剩余加工时间和最小剩余加工时间, f_i 表示工件i的剩余加工时间, $R_{b,i}$ 表示工件i在规则b下的优先值;

基于工件的剩余加工工序数的具体调度规则如下:

$$R_{c,i} = 1 - \frac{n_i - n_{\min}}{n_{\max} - n_{\min}} \quad (3)$$

其中： n_{\max} 和 n_{\min} 分别表示待加工工件中的最大剩余工序数和最小剩余工序数， n_i 表示工件i的剩余工序数， $R_{c,i}$ 表示工件i在规则c下的优先加工值；

基于机器的加工时间的具体调度规则如下：

$$R_{d,i} = \frac{M_i^k}{M_{\max}} \quad (4)$$

其中： M_{\max} 表示所有机器开工时间的最大值， M_i^k 表示在加工工件i的第k道工序所在机器的总加工时间， $R_{d,i}$ 表示工件i在规则d下的优先加工值；

所述步骤二中，所述频繁工序块的获取方法具体包括以下步骤：

S1：对历史调度数据集中随机选取若干个体，依次计算每个父代种群个体在第一基因点位置上各个工序号出现的概率，如果不满足 $\text{Support} \geq S_1$ ，则计算下一个基因位置上的各个工序是否满足最小支持度；如果所有的基因点位上的工序都不满足最小支持度，则重新随机生成调度数据样本；如果满足 $\text{Support} \geq S_1$ ，执行S2；

S2：计算在该工序发生的前提下，其相邻基因位置上的工序出现的概率是否满足 $\text{Confidence} \geq C_1$ ，如果不满足，则重新计算该基因位置上的工序号出现的频率是否满足最小支持度，并重复S1步骤；如果满足最小置信度，则对满足最小支持度的工序和最小置信度的相邻工序进行记录；

S3：将S2中记录满足最小支持度与最小置信度的相邻工序作为一个频繁工序块；

S4：计算S4得到的频繁工序块出现的频率是否满足最小支持度；如果不满足 $\text{Support} \geq S_1$ ，则停止操作，将该频繁工序块输出，并重复S1步骤；如果满足 $\text{Support} \geq S_1$ ，则再次频繁工序块发生的前提下，计算下一相邻基因位置上的工序出现的概率是否满足 $\text{Confidence} \geq C_1$ ，并执行S5；

S5：如果不满足 $\text{Confidence} \geq C_1$ ，则输出S3过程中得到的频繁工序块，并重复S1步骤；如果满足 $\text{Confidence} \geq C_1$ ，则将该道工序添加到频繁工序块中；

S6：重复S1至S5步骤，直至满足终止条件，输出所有满足条件的频繁工序块；

所述步骤六中，初始种群的产生具体包括以下步骤：

T1：将步骤五得到的频繁工序块赋予一个权重因子h；

T2：利用提取规则对父代种群的分析，得到每道工序可能出现在各个基因位置上的概率；

T3：判断频繁工序块所在基因位置上被分配的工序出现的概率是否大于h，如果该道工序出现的概率大于h，则按照提取规则分配的工序进行排列，否则保留该基因位置上频繁工序块；

T4：重复T1至T3步骤，直至达到终止条件；

所述步骤十一具体包括以下步骤：

取选取基因串中从第一个频繁工序块中的首块工序位置到最后一个频繁工序块中的尾块工序位置，这一段基因片段进行变异操作；在种群中选取适应度值最高的一个种群个体与其他个体进行分段式海明距离计算；如果被选中的种群个体的分段式海明距离为零，则该个体不再进行变异操作；如果被选中的种群个体的分段海明距离不为零，则随机选择其中几个相异点进行基因互换。

基于数据驱动改进遗传算法求解作业车间调度问题的方法

技术领域

[0001] 本发明涉及车间调度技术领域,更具体地说,它涉及一种基于数据驱动改进遗传算法求解作业车间调度问题的方法。

背景技术

[0002] 新一轮工业革命的到来,促使信息技术迅速融入到工业制造过程中,从而企业和工厂都积累了海量的生产调度数据,这些数据呈现出不规则性和多样性等特性。而传统作业车间调度问题研究往往忽略了工业生产中历史调度数据的潜在规律。因此,研究数据驱动在求解作业车间调度问题的应用研究方面影响具有重要的意义。

发明内容

[0003] 为解决上述技术问题,本发明提供一种基于数据驱动改进遗传算法求解作业车间调度问题的方法,由于遗传算法不受限制性条件的约束,因此在作业车间调度问题上得到广泛应用,然而采用遗传算法求解作业车间调度问题,初始种群的质量对遗传算法求解效率有很大的影响,初始种群的质量越高,算法的起点就越高,从而可以提高遗传算法的求解效率和质量,可以更有效的求解车间调度问题。

[0004] 本发明的上述技术目的是通过以下技术方案得以实现的:

[0005] 基于数据驱动改进遗传算法求解作业车间调度问题的方法,包括以下步骤:

[0006] 步骤一:从历史调度数据集中随机选取若干个体;

[0007] 步骤二:利用关联规则获取频繁工序块,同时组合调度规则获取每道工序出现在各个基因的位置及概率;

[0008] 步骤三:预先设置一阈值,判断频繁工序块位置上的基因出现概率是否大于阈值,如果是,执行步骤四;如果不是,执行步骤五;

[0009] 步骤四:按照组合调度规则生成的工序进行排列,并执行步骤六;

[0010] 步骤五:保留该基因位置上的频繁工序块,并执行步骤六;

[0011] 步骤六:将生成的种群 $a\%+(1-a)\%$ 随机种群作为初始种群;

[0012] 步骤七:进行锦标赛选择;

[0013] 步骤八:判断选择的两个个体是否都具频繁工序块,如果是,执行步骤九;如果不是,执行步骤十;

[0014] 步骤九:判断待交叉个体中频繁工序块的数量是否等于1,如果是,执行单点交叉,如果不是,执行两点交叉;然后执行步骤十一;

[0015] 步骤十:判断两个个体是否都不具有频繁工序块,如果是,执行POX交叉;如果不是,执行POX交叉,并更新频繁工序块;然后执行步骤十一;

[0016] 步骤十一:进行变异操作,更新频繁工序块信息;

[0017] 步骤十二:判断是否满足终止条件,如果是,输出最优结果;如果不是,重复步骤七至步骤十二;终止条件为运算循环的次数达到迭代次数 N_{iter} 时停止。

[0018] 作为一种优选方案,步骤一中,从历史调度数据集中随机选取40-60%的个体。

[0019] 作为一种优选方案,步骤二中,调度规则包括基于工序的加工时间 $R_{a,i}$,基于工件的剩余加工时间 $R_{b,i}$,基于工件的剩余加工工序数 $R_{c,i}$ 和基于机器的加工时间 $R_{d,i}$,调度规则的线性组合参数表现形式为Power。

[0020] 作为一种优选方案,基于工序的加工时间的具体调度规则如下:

$$[0021] \quad R_{a,i} = 1 - \frac{t_{\max} - t_i^k}{t_{\max} - t_{\min}} \quad (1)$$

[0022] 其中: t_{\max} 和 t_{\min} 分别表示工序的最大加工时间和最小加工时间, t_i^k 表示工件i的第k道工序的加工时间, $R_{a,i}$ 表示工件i在规则a下的优先值;

[0023] 基于工件的剩余加工时间的具体调度规则如下:

$$[0024] \quad R_{b,i} = 1 - \frac{f_{\max} - f_i}{f_{\max} - f_{\min}} \quad (2)$$

[0025] 其中: f_{\max} 和 f_{\min} 分别表示待加工工件的最大剩余加工时间和最小剩余加工时间, f_i 表示工件i的剩余加工时间, $R_{b,i}$ 表示工件i在规则b下的优先值;

[0026] 基于工件的剩余加工工序数的具体调度规则如下:

$$[0027] \quad R_{c,i} = 1 - \frac{n_i - n_{\min}}{n_{\max} - n_{\min}} \quad (3)$$

[0028] 其中: n_{\max} 和 n_{\min} 分别表示待加工工件中的最大剩余工序数和最小剩余工序数, n_i 表示工件i的剩余工序数, $R_{c,i}$ 表示工件i在规则c下的优先加工值;

[0029] 基于机器的加工时间的具体调度规则如下:

$$[0030] \quad R_{d,i} = \frac{M_i^k}{M_{\max}} \quad (4)$$

[0031] 其中: M_{\max} 表示所有机器开工时间的最大值, M_i^k 表示在加工工件i的第k道工序所在机器的总加工时间, $R_{d,i}$ 表示工件i在规则d下的优先加工值。

[0032] 作为一种优选方案,步骤二中,所述频繁工序块的获取方法具体包括以下步骤:

[0033] S1:对历史调度数据集中随机选取若干个体,依次计算每个父代种群个体在第一基因点位置上各个工序号出现的概率,如果不满足 $\text{Support} \geq S_1$,则计算下一个基因位置上的各个工序是否满足最小支持度;如果所有的基因点位上的工序都不满足最小支持度,则重新随机生成调度数据样本;如果满足 $\text{Support} \geq S_1$,执行S2;

[0034] S2:计算在该工序发生的前提下,其相邻基因位置上的工序出现的概率是否满足 $\text{Confidence} \geq C_1$,如果不满足,则重新计算该基因位置上的工序号出现的频率是否满足最小支持度,并重复S1步骤;如果满足最小置信度,则对满足最小支持度的工序和最小置信度的相邻工序进行记录;

[0035] S3:将S2中记录满足最小支持度与最小置信度的相邻工序作为一个频繁工序块;

[0036] S4:计算S4得到的频繁工序块出现的频率是否满足最小支持度;如果不满足 $\text{Support} \geq S_1$,则停止操作,将该频繁工序块输出,并重复S1步骤;如果满足 $\text{Support} \geq S_1$,则

再次频繁工序块发生的前提下,计算下一相邻基因位置上的工序出现的概率是否满足 $\text{Confidence} \geq C_1$, 并执行S5;

[0037] S5: 如果不满足 $\text{Confidence} \geq C_1$, 则输出S3过程中得到的频繁工序块, 并重复S1步骤; 如果满足 $\text{Confidence} \geq C_1$, 则将该道工序添加到频繁工序块中;

[0038] S6: 重复S1至S5步骤, 直至满足终止条件, 输出所有满足条件的频繁工序块。

[0039] 作为一种优选方案, 步骤六中, 初始种群的产生具体包括以下步骤:

[0040] T1: 将步骤五得到的频繁工序块赋予一个权重因子h;

[0041] T2: 利用提取规则对父代种群的分析, 得到每道工序可能出现在各个基因位置上的概率;

[0042] T3: 判断频繁工序块所在基因位置上被分配的工序出现的概率是否大于h, 如果该道工序出现的概率大于h, 则按照提取规则分配的工序进行排列, 否则保留该基因位置上频繁工序块;

[0043] T4: 重复T1至T3步骤, 直至达到终止条件。

[0044] 作为一种优选方案, 步骤十一具体包括以下步骤:

[0045] 选取基因串中从第一个频繁工序块中的首块工序位置到最后一个频繁工序块中的尾块工序位置, 这一段基因片段进行变异操作; 在种群中选取适应度值最高的一个种群个体与其他个体进行分段式海明距离计算; 如果被选中的种群个体的分段式海明距离为零, 则该个体不再进行变异操作; 如果被选中的种群个体的分段海明距离不为零, 则随机选择其中几个相异点进行基因互换。

[0046] 步骤1: 参数设置, 包括种群规模 N_{pop} 、迭代次数 N_{iter} 、交叉概率 P_c 、变异概率 P_m 、最小支持度 S_1 、最小置信度 C_1 、四种调度规则: 基于工序的加工时间 $R_{a,i}$ 、基于工件的剩余加工时间 $R_{b,i}$ 、基于工件的剩余加工工序数 $R_{c,i}$ 和基于机器的加工时间 $R_{d,i}$, 调度规则的线性组合参数表现形式 Power。

[0047] 本发明算法的终止条件为: 运算循环的次数达到最大迭代 N_{iter} 时停止。

[0048] 步骤2: 通过关联规则和组合调度规则对历史调度数据集进行数据挖掘。

[0049] 首先随机选取历史调度数据集规模50%的种群个体作为调度数据样本; 然后, 根据作业车间调度问题的特性, 对关联规则中的最小支持度、最小置信度阈值和组合调度规则进行定义。

[0050] 步骤3: 将频繁工序块和组合调度规则获得的规则集进行“结合”, 产生新的子代种群, 并对该子代种群进行适应度值大小排列, 选取前80%的子代个体和20%随机策略生成的个体作为遗传算法求解作业车间调度问题的初始种群。

[0051] 步骤4: 遗传算法中的选择操作作为选择策略。

[0052] 步骤5: 在进行交叉操作时, 通过判断具有频繁工序块的待交叉操个体的数量, 来决定进行交叉操作的方式。

[0053] 如果两个个体都具有频繁工序块, 则进行两点交叉操作, 如果基因序列中只有一个频繁工序块, 则进行单点交叉操作; 如果只有一个个体具有频繁工序块进行POX交叉操作, 并比较父代、子代个体适应度值大小, 如果子代个体优于父代个体, 则对频繁工序块所处基因位置上的工序信息进行更新; 如果两个个体都不具有频繁工序块, 则也进行POX交叉操作并且不再更新频繁工序块信息。

[0054] 步骤6:执行变异操作;将“分段式海明距离”概念引入变异操作过程中,分段式海明距离表示为:只计算两个基因串中的某一部分基因片段进行相异的基因位置数目称作分段式海明距离,简称H。

[0055] 挑选种群中适应度值最大者来计算分段式海明距离,这里取选取基因串中从第一个频繁工序块的首块工序位置到最后一个频繁工序块中尾块位置一段基因片段进行变异操作。

[0056] 如果被选中的种群个体的分段式海明距离为零,则该个体不进行变异操作。

[0057] 如果被选中的种群个体的分段海明距离不为零,则随机选择其中几个(偶数)相异点进行基因互换,来完成遗传算法的变异操作。

[0058] 步骤7:更新当前最优解,直至满足终止条件。

[0059] 综上所述,本发明具有以下有益效果:

[0060] 本发明将传统的遗传算法经过改进以此来解决遗传算法求解作业车间调度问题时对初始种群质量的依赖性问题。在该发明中采用两种方式进行优化遗传算法初始种群,首先利用关联规则找出基因串中具有关联性的优势相邻工序组合,其次利用组合调度规则获得工序之间的排列顺序关系,两者结合产生“新的种群”作为遗传算法求解作业车间调度问题的初始种群;两种优化初始种群方式相混合产生初始种群的方法提高了种群整体初始解的质量以及种群个体之间的关联性;并且在遗传操作过程中引入“分段式海明距离”概念,在降低种群个体之间的差异的同时,还能减少迭代后期因变异操作对优良基因的破坏。

附图说明

[0061] 图1本发明实施例中的AR-GA算法的流程图;

[0062] 图2本发明实施例中的AR-GA算法测试案例示意图;

[0063] 图3本发明实施例中的第一种两点交叉操作示意图;

[0064] 图4本发明实施例中的第二种POX交叉操作示意图;

[0065] 图5本发明实施例中的第三种POX交叉操作示意图;

[0066] 图6本发明实施例中的变异操作示意图;

[0067] 图7本发明实施例中的基准案例运算结果曲线图。

具体实施方式

[0068] 本说明书及权利要求并不以名称的差异作为区分组件的方式,而是以组件在功能上的差异作为区分的准则。如在通篇说明书及权利要求当中所提及的“包括”为一开放式用语,故应解释成“包括但不限于”。“大致”是指在可接受的误差范围内,本领域技术人员能够在一定误差范围内解决所述技术问题,基本达到所述技术效果。

[0069] 本说明书及权利要求的上下左右等方向名词,是结合附图以便于进一步说明,使得本申请更加方便理解,并不对本申请做出限定,在不同的场景中,上下、左右、里外均是相对而言。

[0070] 以下结合附图对本发明作进一步详细说明。

[0071] 本发明为一种数据驱动改进遗传算法求解作业车间调度问题的方法,包括如下步骤:

[0072] 步骤1:参数设置:种群规模 N_{POP} 、迭代次数 N_{iter} 、交叉概率 P_C 、变异概率 P_M 、最小支持度 S_1 、最小置信度 C_1 、四种调度规则:基于工序的加工时间 $R_{a,i}$ 、基于工件的剩余加工时间 $R_{b,i}$ 、基于工件的剩余加工工序数 $R_{c,i}$ 和基于机器的加工时间 $R_{d,i}$,调度规则的线性组合参数表现形式Power。

[0073] 本文发明的算法的终止条件为:运算循环的次数达到迭代次数 N_{iter} 时停止。

[0074] 关于变量取值的说明:种群规模 N_{POP} 取值为300和500两种不同规模进行比较说明。

[0075] 迭代次数本发明算法选取100代。

[0076] 交叉概率 P_C , $0 < P_C < 1$,一般在0.7左右,本发明取值为0.8。

[0077] 变异概率 P_M , $0 < P_M < 1$,一般取值在0.1左右,本发明取值为0.1。

[0078] 步骤2:通过关联规则和组合调度规则进行数据挖掘。

[0079] 频繁工序块的提取:

[0080] ①、随机生成的海量调度数据样本种群个体并进行适应度值计算,然后选取适应度值前50%种群个体进行数据挖掘。

[0081] ②、依次计算每个父代种群个体在第一个基因点位置上各个工序号出现的概率。

[0082] ③、如果不满足 $Support \geq S_1$ 时,则计算下一个基因位置上的各个工序是否满足最小支持度。如果所有的基因点位上的工序都不满足最小支持度时,则重新随机生成调度数据样本。

[0083] ④、如果满足 $Support \geq S_1$,则计算在该工序发生的前提下,其相邻基因位置上的工序出现的概率是否满足 $Confidence \geq C_1$ 。

[0084] ⑤、如果不满足 $Confidence \geq C_1$,则重新计算该基因位置上的工序号出现的频率是否满足最小支持度,以下操作流程跳转到③。如果满足最小置信度时,则对满足最小支持度的工序和最小置信度的相邻工序进行记录。

[0085] ⑥、将⑤中记录满足最小支持度与最小置信度的相邻工序作为一个频繁工序块;计算该频繁工序块出现的频率是否满足最小支持度;如果不满足 $Support \geq S_1$,则停止操作,将该频繁工序块输出。跳转到③。

[0086] ⑦、如果⑥中的频繁工序块出现的频率满足 $Support \geq S_1$,则在此频繁工序块发生的前提下,计算下一个相邻基因位置上的工序出现的概率是否满足 $Confidence \geq C_1$ 。

[0087] ⑧、如果不满足 $Confidence \geq C_1$,则输出⑤得到的频繁工序块,并跳转到③;如果满足 $Confidence \geq C_1$,则将该道工序添加到频繁工序块中。

[0088] ⑨、以此类推,直至满足终止条件,输出所有满足条件的频繁工序块。

[0089] 组合调度规则的挖掘:本发明定义四种简单调度规则如下所示。

[0090] 基于工序的加工时间:由于每一道工序的加工时间不同,不同的加工时间的排序也影响着解决方案的优劣。其中较短加工时间的工序具有容易插空、易排序的特点,而较长加工时间的工序的排序,影响着总加工时间的长度,所以较长加工时间的工序具有一定的优先加工权,公式如下。其中, t_{max} 和 t_{min} 分别表示工序的最大加工时间和最小加工时间, t_i^k 表示工件i的第k道工序的加工时间, $R_{a,i}$ 表示工件i在规则a下的优先值。

$$[0091] \quad R_{a,i} = 1 - \frac{t_{max} - t_i^k}{t_{max} - t_{min}}$$

[0092] 基于工件的剩余加工时间: 剩余加工时间是指在加工工件j的工序p之后, 完成工件j还需要多少加工时间。由于各个工件的剩余加工时间的大小决定着工件的优先加工等级, 所以本文较长剩余加工时间的工件具有优先加工权。公式如下, 其中, f_{\max} 和 f_{\min} 分别表示待加工工件的最大剩余加工时间和最小剩余加工时间, f_i 表示工件i的剩余加工时间, $R_{b,i}$ 表示工件i在规则b下的优先值。

$$[0093] \quad R_{b,i} = 1 - \frac{f_{\max} - f_i}{f_{\max} - f_{\min}}$$

[0094] 基于工件的剩余加工工序数: 在考虑剩余加工时间时, 也需要考虑剩余加工工序数目对加工的影响。剩余加工工序数目就是是指在加工工件j的工序p之后, 还需要加工几道工序才能完成对工件j的加工。在求解作业车间调度问题时在合理分配工件之间的加工顺序的同时加快完成较短加工时间的工序, 有利于降低总的完工时间, 所以剩余加工工序数目较少的工件具有优先加工权。公式如下, 其中, n_{\max} 和 n_{\min} 分别表示待加工工件中的最大剩余工序数和最小剩余工序数, n_i 表示工件i的剩余工序数, $R_{c,i}$ 表示工件i在规则c下的优先加工值。

$$[0095] \quad R_{c,i} = 1 - \frac{n_i - n_{\min}}{n_{\max} - n_{\min}}$$

[0096] 基于机器的加工时间: 机器的加工时间是指加工完该机器上面分配的工件所需要的时间。由于每台机器的工作时间长度差异较大, 合理调配各个机器的加工时间将较短完工时间的机器上的工序后移较长完工时间机器上的工序先加工, 使每台机器的完工时间相近, 这样有利于降低工件总的完工时间, 所以分配在较长工作时间机器上的工件具有优先加工权。公式如下, 其中, M_{\max} 表示所有机器开工时间的最大值, M_i^k 表示在加工工件i的第k道工序所在机器的总加工时间, $R_{d,i}$ 表示工件i在规则d下的优先加工值。

$$[0097] \quad R_{d,i} = \frac{M_i^k}{M_{\max}}$$

[0098] 步骤3: 初始种群的产生。

[0099] 将所得到的频繁工序块赋予一个权重因子, 该权重因子大小为h ($0 < h < 1$)。

[0100] 利用提取规则对父代种群的分析, 可以得到每道工序可能出现在各个基因位置上的概率。

[0101] 判断频繁工序块所在基因位置上被分配的工序出现的概率大于h; 如果该道工序出现的概率大于h, 则按照提取规则分配的工序进行排列, 否则, 保留该基因位置上频繁工序块。

[0102] 重复此操作, 直至达到终止条件。

[0103] 为了确保初始种群的多样性, 需要将得到的子代种群进行适应度值计算, 并只保留适应度值前a%的种群个体与(1-a)%的随机生成的种群个体, 作为遗传算法求解作业车间调度问题的初始种群。

[0104] 步骤4: 遗传算法中的选择操作为锦标赛选择策略。

[0105] 步骤5: 交叉操作会出现三种情况:

[0106] 当出现选择的两个种群个体都具有频繁工序块时, 采用两点交叉操作如图3(其中

深色标记的为频繁工序块)所示,即对基因串上的每一个频繁工序块中第一个工序位置进行标记,然后随机选择其中两个标记点进行基因片段交换,如果基因串中只有一个频繁工序块,就进行单点交叉方式,以此来降低交叉操作对频繁工序块的破坏程度。

[0107] 当出现二种情况时,采用POX交叉操作如图4示(其中深色标记为频繁工序块),因交叉操作产生的Child3、Child4子代个体,进行子代适应度值大小计算。如果子代个体的适应度值小于父代个体适应度值,则产生子代个体不含频繁工序块;反之,则Child3,Child4个体基因串对应父代频繁工序块的基因位置上的相邻工序作为新的频繁工序块,并且更新进行频繁工序块信息,通过产生新的频繁工序块来增加算法的局部搜索能力。

[0108] 当出现第三种情况时,经过POX交叉操作产生的子代基因串不含频繁工序块,因此可以增加种群的多样性,如图5所示。

[0109] 步骤6:为使基因串中的频繁工序块尽可能地保留下来,引入“分段式海明距离”的概念:只取两个基因串中的某一部分基因片段进行相异的基因位置个数计算称作分段式海明距离,简称H,如图所示6。

[0110] 这里取选取基因串中从第一个频繁工序块中的首块工序位置到最后一个频繁工序块中的尾块工序位置,这一段基因片段进行变异操作。

[0111] 在种群中选取适应度值最高的一个种群个体与其他个体进行分段式海明距离计算。

[0112] 如果被选中的种群个体的分段式海明距离为零,则该个体不再进行变异操作;如果被选中的种群个体的分段海明距离不为零,则随机选择其中几个(偶数)相异点进行基因互换,以此降低两个种群个体之间的分段式海明距离,并且更新进行频繁工序块信息。

[0113] 步骤7:判断算法是否终止。

[0114] 依据参数中对算法终止条件的设定,判断算法是否终止,若算法终止,则输出当前最优解;反之,算法未终止,则返回步骤4。

[0115] 本发明将传统的遗传算法经过改进以此来解决遗传算法求解作业车间调度问题时对初始种群质量的依赖性问题。在该发明中采用两种方式进行优化遗传算法初始种群,首先利用关联规则找出基因串中具有关联性的工序组合,其次利用组合调度规则获得工序之间的排列顺序关系,两者结合产生“新的种群”作为遗传算法求解作业车间调度问题的初始种群;两种优化初始种群方式相混合产生初始种群的方法提高了种群整体初始解的质量以及种群个体之间的关联性;并且在遗传操作过程中引入“分段式海明距离”概念,在降低种群个体之间的差异的同时,还能减少迭代后期因变异操作对优良基因的破坏。

[0116] 本具体实施例仅仅是对本发明的解释,其并不是对本发明的限制,本领域技术人员在阅读完本说明书后可以根据需要对本实施例做出没有创造性贡献的修改,但只要在本发明的权利要求范围内都受到专利法的保护。

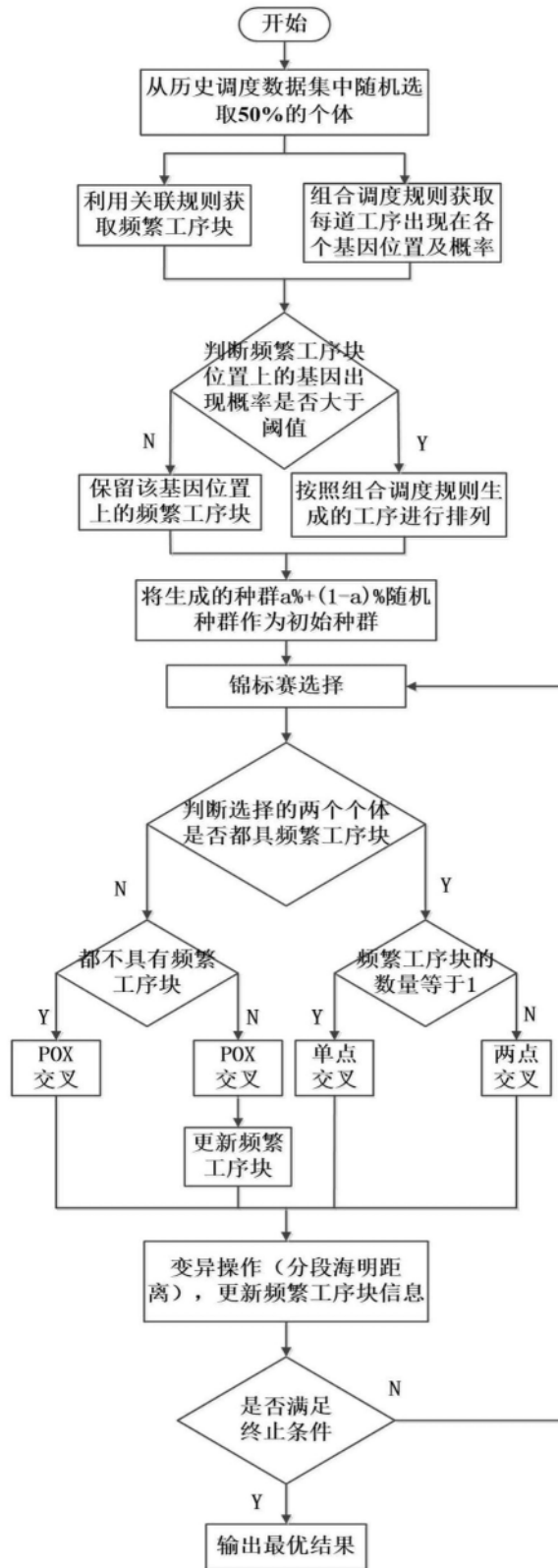


图1

	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
J1	9.14	5.75	4.12	3.38	1.76	6.97	10.12	2.29	8.44	7.66
J2	6.38	4.82	3.85	5.58	7.87	10.89	1.43	2.80	8.69	9.92
J3	10.5	2.84	1.43	7.48	5.8	8.7	4.41	6.61	9.66	3.14
J4	3.42	2.8	1.96	6.19	5.59	8.97	10.73	9.43	4.74	7.41
J5	7.55	3.70	4.75	9.42	5.37	8.23	2.48	5.5	10.38	1.7
J6	9.9	3.72	8.31	1.79	6.73	4.95	5.25	7.43	10.60	2.56
J7	1.97	3.64	4.78	6.21	5.94	10.31	9.53	7.16	8.86	2.7
J8	4.86	8.85	10.63	1.61	3.65	5.30	6.32	2.33	9.44	7.59
J9	3.44	4.16	5.11	7.45	2.30	10.84	9.93	1.60	6.61	8.90
J10	8.36	9.31	5.47	7.52	1.32	6.11	3.28	10.35	4.20	2.49
J11	9.20	7.49	8.74	5.10	6.17	4.34	1.85	3.77	10.68	2.84
J12	2.85	6.7	9.71	7.59	5.76	1.17	4.29	3.17	8.48	10.13
J13	3.15	7.87	8.11	2.39	5.39	9.43	1.19	4.32	10.16	6.64
J14	7.32	3.92	6.33	9.82	2.83	8.57	10.99	5.91	4.99	1.8
J15	5.88	8.7	9.27	2.38	4.91	3.69	7.21	10.62	6.39	1.48

图2

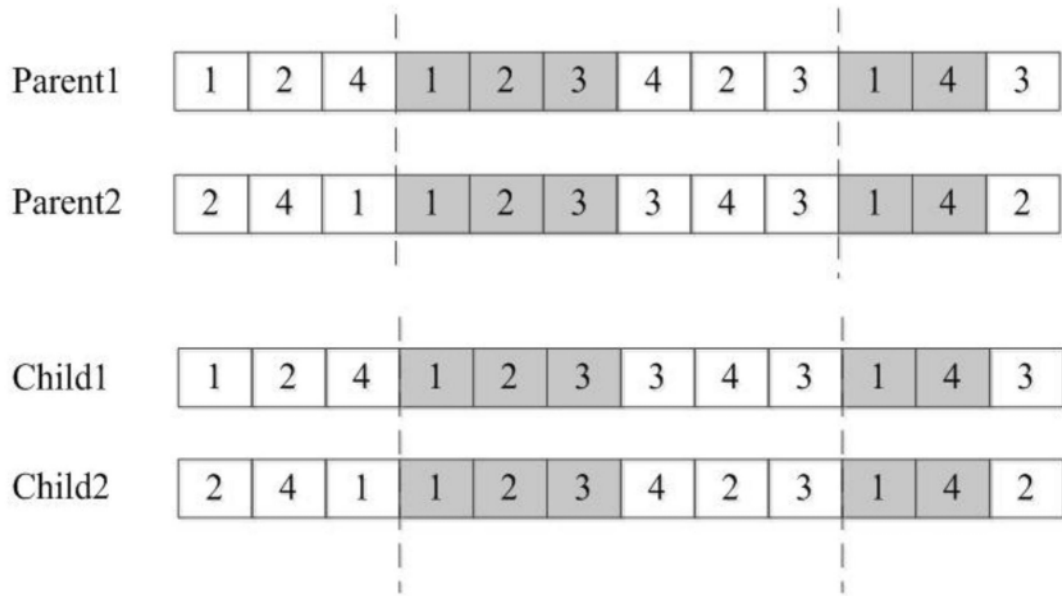


图3

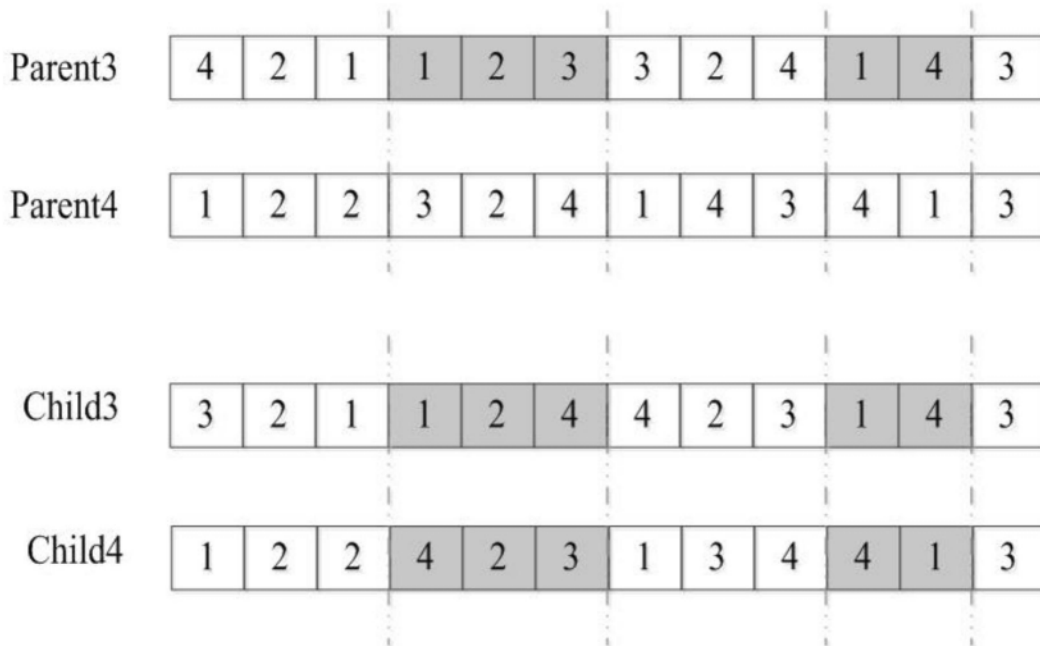


图4

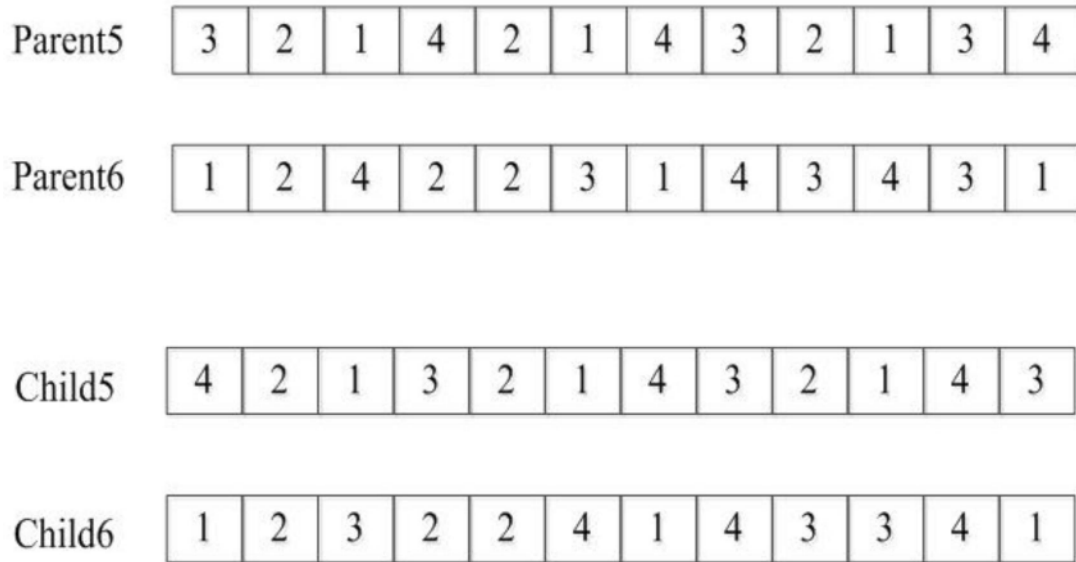


图5

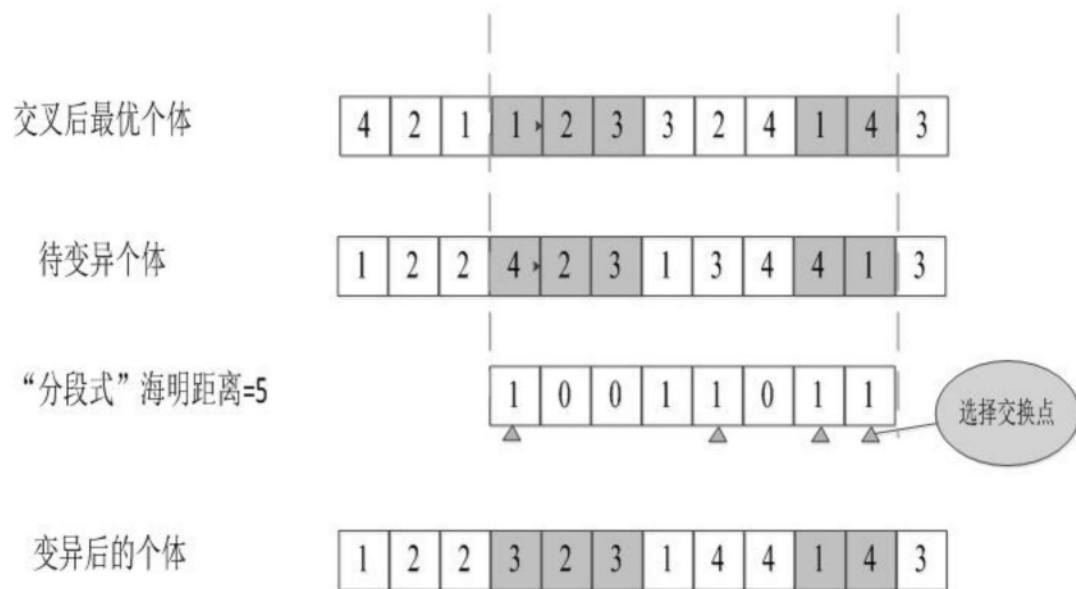


图6

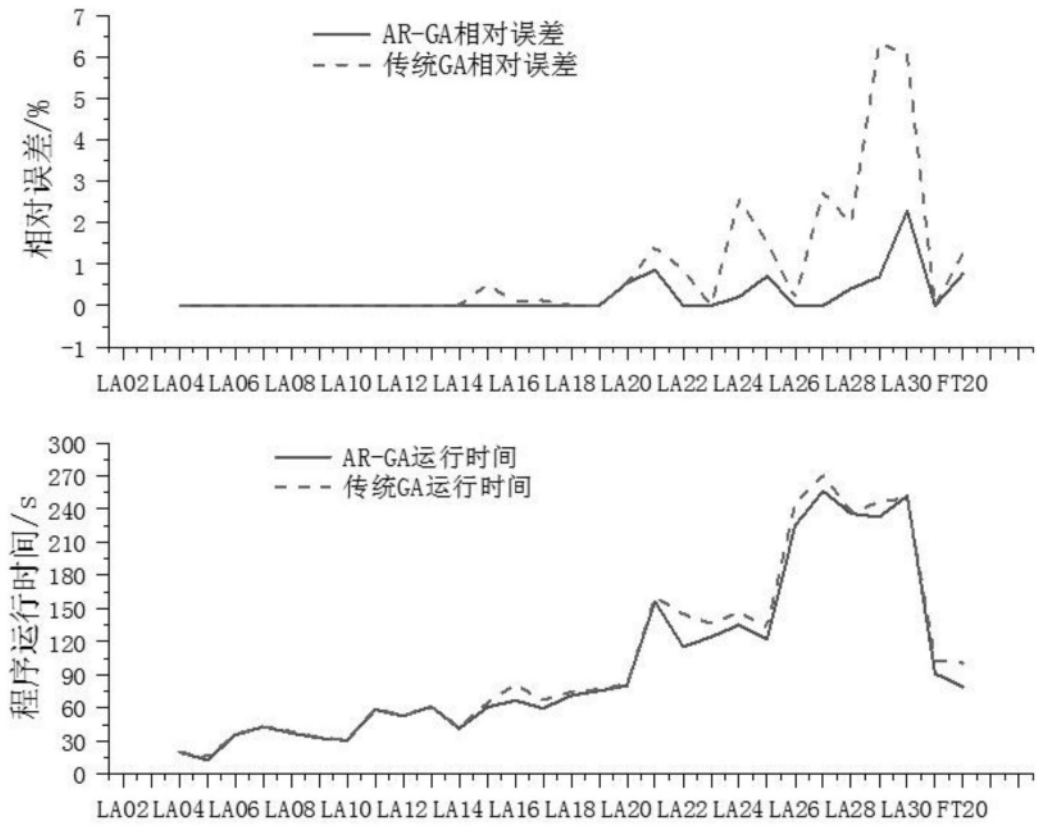


图7