



(12) 发明专利申请

(10) 申请公布号 CN 102566431 A

(43) 申请公布日 2012. 07. 11

(21) 申请号 201210012197. 1

(22) 申请日 2012. 01. 16

(71) 申请人 浙江工业大学

地址 310014 浙江省杭州市下城区朝晖六区

(72) 发明人 陈勇 盛家君 邱晓杰 吴云翔
潘益菁

(74) 专利代理机构 杭州天正专利事务所有限公
司 33201

代理人 王兵 王利强

(51) Int. Cl.

G05B 13/04 (2006. 01)

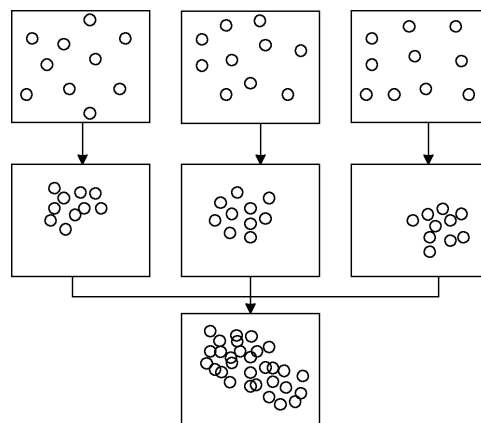
权利要求书 2 页 说明书 9 页 附图 1 页

(54) 发明名称

一种基于改进遗传算法的动态扰动下的模糊化参数作业车间调度控制方法

(57) 摘要

一种基于改进遗传算法的解决复杂生产环境下作业车间调度控制方法,包括以下步骤:1)、设定模糊化参数的作业车间调度模型,以最大化所有产品满意度和最大化最小满意度为目标函数;2)、采用改进的遗传算法求解所述目标函数,(2.1) 采用基于工序的编码;(2.2) 多次运行 G&T 算法产生一个初始种群;(2.3) 采用目标函数作为适应度函数对个体进行评价;(2.4) 选择和交叉操作;(2.5) 采用反转变异;(2.6) 当每个组的个体均收敛到某一个度时,将各组种群合并继续进化直到收敛;(2.7) 以预先设定的最大进化代数 Nmax 作为停止条件,将目前为止最好的解作为最优解,得到解决复杂生产环境下作业车间调度方案。本发明精确性良好、可靠性较好、实用性强。



1. 一种基于改进遗传算法的动态扰动下的模糊化参数作业车间调度控制方法,其特征
在于:所述控制方法包括以下步骤:

1)、确定模糊化参数作业车间调度的目标函数

模糊作业车间调度问题的目标函数为:求一个满足 z^* 的工件加工顺序,使得:

$$z^* = \max(z') \quad (4)$$

$$\left\{ \begin{array}{l} z' = y_1 z_1 + y_2 z_2 \\ \max z_1 = \sum_{i=1}^n w_i AI_i \\ \max z_2 = AI_{\min} = \min_{i=1, \Delta n} AI_i \\ \sum_{i=1}^n w_i = 1 \\ y_1 + y_2 = 1 \end{array} \right. \quad (4-1)$$

2)、采用改进的遗传算法求解所述目标函数,具体过程如下:

(2.1) 编码:对调度窗口中的工件采用基于工序的编码;

(2.2) 初始种群的生成:多次运行 G&T 算法产生一个初始种群;

(2.3) 采用目标函数作为适应度函数对个体进行评价;

(2.4) 选择和交叉操作:

采用交叉操作从两个父代中产生一个子代,具体如下:

步骤 2.4.1. 选择所有工件的第 1 道操作,加入集合 C;假定机器在同一时刻可以加工任意多个作业,计算集合 C 中各操作 $O_{ijk} \in C$ 的模糊完成时间,记作 $(EC_{ijk}^1, EC_{ijk}^2, EC_{ijk}^3)$;根据三角模糊数的比较准则,确定集合 C 中最小模糊完成时间的操作 $O_{i^*j^*k^*}$,将与操作 $O_{i^*j^*k^*}$ 使用同一台机器 M_{k^*} 的操作 $O_{ijk} \in C$ 构成集合 G。由于操作 O_{ijk} 与操作 $O_{i^*j^*k^*}$ 的操作时间上是有重叠的,集合 G 为冲突集合;

步骤 2.4.2. 以 50% 的相同概率从两个父代个体中任选一个个体,在冲突集合 G 中选择具有最小模糊完成时间的操作,用选出的父代个体代替并表示为 O_{i_s, j_s, k^*} ;

步骤 2.4.3. 以步骤 2.4.2. 中所选择的操作为基准,通过模糊数的取大操作依次更新冲突集合中其他操作的 EC^1, EC^2 和 EC^3 值,从集合 C 中移去操作 O_{i_p, j_p, k^*} ,将其下一道工序加入集合 C,并计算其相应模糊完成时间;

通过以上的操作得到了一个新的子代个体,进行 c 次以上的操作得到 c 个新子代个体。

为保留由 c 个子代和 2 个父代组成的 (c+2) 个个体中具有优良性状的个体,用以下的方法选择两个在下一次遗传时保留下来的个体:

a. 在 c 个子代个体中,选择具有最大目标函数值的个体,也即局部排名选择;

b. 在剩下的 (c+1) 个个体中,选择具有最大目标函数值的个体;

(2.5) 变异操作:采用反转变异;

(2.6) 种群构造:当每个组的个体均收敛到某一个度时,将各组种群合并继续进化直到收敛;

(2.7) 以预先设定的最大进化代数 N_{\max} 作为停止条件,将目前为止最好的解作为最优解;

3) 在动态事件发生时进行再调度,具体步骤如下:

步骤 1. 初始化调度次数 $n = 0$,开始调度的时间 $t_0 = 0$,给定窗口中工件的最大数目 L_w ,将待加工工件放入窗口内,设工件窗口为 N_w ,已完成工件集为 N_c 。

步骤 2. 对当前窗口内的工件执行改进遗传算法,选择使目标函数最优的染色体作为调度结果,产生调度方案;

步骤 3. 当有以下动态事件发生时,执行一次再调度,设动态事件发生时刻为 t_c ;

①当有插单工件到来时启动一次再调度,转步骤 4;

②当工件到期时间改变时启动一次再调度,转步骤 4;

③当机器损坏事件发生时启动一次再调度,转步骤 4;

步骤 4. 修改可利用设备项,将已完成的工序从工件窗口 N_w 迁入已完成工件集 N_c ,产生已完成工序的调度方案;未完成工序的顺序提前,将剩余工序作为未调度的新工序:

①将插单工件加入工件窗口 N_w ,和未完成工序组成新的待加工工序,执行再调度。若插单工件到来时,某机器正在加工某道工序,则自动延迟一定时间,该道工序完成后执行再调度;

②工件到期时间改变时,更新工件窗口内的工件信息,执行再调度;

③机器损坏时,将正在该机器上加工的工件移出工件窗口,对剩下的工件执行再调度;当仿真时钟运行到剩余工件需在该机器加工的工序时,将此工件移出工件窗口,对剩余工件执行再调度;机器修复时,将移出工件窗口的工件重新迁入,执行再调度;

按照再调度事件重复进行步骤 4,直到所有的工件加工完成,产生最终调度方案。

一种基于改进遗传算法的动态扰动下的模糊化参数作业车间调度控制方法

技术领域

[0001] 本发明涉及作业车间调度控制技术领域,尤其是一种解决复杂生产环境下作业车间调度控制方法。

背景技术

[0002] 近年来对 JSP(作业车间调度问题(Job-Shop Scheduling Problem, JSP))的研究,已经突破了原始的运筹学范畴,在管理科学、控制论、人工智能、工业工程、系统工程等领域都有学者用各自的领域知识开展了卓有成效的研究工作,从而推动各种优化算法的发展和融合。随着先进制造技术的发展,作业车间调度问题的含义有所拓展,增加了随机性、动态性、不确定性、约束性、多目标性等属性,其与实际生产情况更为接近。

[0003] 目前对作业车间调度问题的研究大多存在以下问题:

[0004] (1) 将生产过程中的各项参数看做确定性的精确值。过去的研究工作多数集中于确定型作业车间调度问题,即工件的加工时间和交货期都是已知的确定量。而事实上,由于各种随机因素,如机器故障、操作工人的熟练程度、环境参数等影响,很少能获得精确的加工时间和交货期,人们只能估计加工时间或交货期的变化范围。因此,调度中生产参数用非精确数来表示更符合生产实际,即模糊作业车间调度问题。

[0005] (2) 将生产系统当作一个静态的系统,忽略实际生产过程中各种突发状况的出现。由于待加工的工件连续不断的到来、机床设备突然损坏或被修复、工人生病、紧急订单等各种状况的出现而产生的原调度中断失效情况,调度中应加以考虑,即动态的作业车间调度。

[0006] 已有的作业车间调度研究往往将生产模型大大简化,与实际生产状况距离较远,不完全适合生产条件变化频繁、工况变化大的制造企业。

发明内容

[0007] 为了克服已有作业车间调度技术的精确性较差、可靠性较差、实用性较差的不足,本发明提供一种精确性良好、可靠性较好、实用性强的基于改进遗传算法的动态扰动下的模糊化参数作业车间调度控制方法。

[0008] 本发明解决其技术问题所采用的技术方案是:

[0009] 一种基于改进遗传算法的动态扰动下的模糊化参数作业车间调度控制方法,所述控制方法包括以下步骤:

[0010] 1)、确定模糊化参数作业车间调度的目标函数

[0011] 模糊作业车间调度问题的目标函数为:求一个满足 z^* 的工件加工顺序,使得:

[0012] $z^* = \max(z')$ (4)

$$\begin{cases}
 z' = y_1 z_1 + y_2 z_2 \\
 \max z_1 = \sum_{i=1}^n w_i AI_i \\
 \max z_2 = AI_{\min} = \min_{i=1, \dots, n} AI_i \\
 \sum_{i=1}^n w_i = 1 \\
 y_1 + y_2 = 1
 \end{cases} \quad (4-1)$$

[0014] 2)、采用改进的遗传算法求解所述目标函数,具体过程如下:

[0015] (2.1) 编码:对调度窗口中的工件采用基于工序的编码;

[0016] (2.2) 初始种群的生成:多次运行 G&T 算法产生一个初始种群;

[0017] (2.3) 采用目标函数作为适应度函数对个体进行评价;

[0018] (2.4) 选择和交叉操作:

[0019] 采用交叉操作从两个父代中产生一个子代,具体如下:

[0020] 步骤 2.4.1. 选择所有工件的第 1 道操作,加入集合 C;假定机器在同一时刻可以加工任意多个作业,计算集合 C 中各操作 $O_{ijk} \in C$ 的模糊完成时间,记作 $(EC_{ijk}^1, EC_{ijk}^2, EC_{ijk}^3)$;根据三角模糊数的比较准则,确定集合 C 中最小模糊完成时间的操作 $O_{i^*j^*k^*}$,将与操作 $O_{i^*j^*k^*}$ 使用同一台机器 M_{k^*} 的操作 $O_{ijk^*} \in C$ 构成集合 G。由于操作 O_{ijk^*} 与操作 $O_{i^*j^*k^*}$ 的操作时间上是有重叠的,集合 G 为冲突集合;

[0021] 步骤 2.4.2. 以 50% 的相同概率从两个父代个体中任选一个个体,在冲突集合 G 中选择具有最小模糊完成时间的操作,用选出的父代个体代替并表示为 O_{i_s, j_s, k^*} ;

[0022] 步骤 2.4.3. 以步骤 2.4.2. 中所选择的操作为基准,通过模糊数的取大操作依次更新冲突集合中其他操作的 EC^1, EC^2 和 EC^3 值,从集合 C 中移去操作 $O_{i_p j_p k^*}$,将其下一道工序加入集合 C,并计算其相应模糊完成时间;

[0023] 通过以上的操作得到了一个新的子代个体,进行 c 次以上的操作得到 c 个新子代个体。

[0024] 为保留由 c 个子代和 2 个父代组成的 (c+2) 个个体中具有优良性状的个体,用以下的方法选择两个在下一次遗传时保留下来的个体:

[0025] a. 在 c 个子代个体中,选择具有最大目标函数值的个体,也即局部排名选择;

[0026] b. 在剩下的 (c+1) 个个体中,选择具有最大目标函数值的个体;

[0027] (2.5) 变异操作:采用反转变异;

[0028] (2.6) 种群构造:当每个组的个体均收敛到某一个度时,将各组种群合并继续进化直到收敛;

[0029] (2.7) 以预先设定的最大进化代数 N_{\max} 作为停止条件,将目前为止最好的解作为最优解;

[0030] 3) 在动态事件发生时进行再调度,具体步骤如下:

[0031] 步骤 1. 初始化调度次数 $n = 0$,开始调度的时间 $t_0 = 0$,给定窗口中工件的最大

数目 L_w , 将待加工工件放入窗口内, 设工件窗口为 N_w , 已完成工件集为 N_c 。

[0032] 步骤 2. 对当前窗口内的工件执行改进遗传算法, 选择使目标函数最优的染色体作为调度结果, 产生调度方案;

[0033] 步骤 3. 当有以下动态事件发生时, 执行一次再调度, 设动态事件发生时刻为 t_c ;

[0034] ①当有插单工件到来时启动一次再调度, 转步骤 4;

[0035] ②当工件到期时间改变时启动一次再调度, 转步骤 4;

[0036] ③当机器损坏事件发生时启动一次再调度, 转步骤 4;

[0037] 步骤 4. 修改可利用设备项, 将已完成的工序从工件窗口 N_w 迁入已完成工件集 N_c , 产生已完成工序的调度方案; 未完成工序的顺序提前, 将剩余工序作为未调度的新工序;

[0038] ①将插单工件加入工件窗口 N_w , 和未完成工序组成新的待加工工序, 执行再调度。若插单工件到来时, 某机器正在加工某道工序, 则自动延迟一定时间, 该道工序完成后执行再调度;

[0039] ②工件到期时间改变时, 更新工件窗口内的工件信息, 执行再调度;

[0040] ③机器损坏时, 将正在该机器上加工的工件移出工件窗口, 对剩下的工件执行再调度; 当仿真时钟运行到剩余工件需在该机器加工的工序时, 将此工件移出工件窗口, 对剩余工件执行再调度; 机器修复时, 将移出工件窗口的工件重新迁入, 执行再调度;

[0041] 按照再调度事件重复进行步骤 4, 直到所有的工件加工完成, 产生最终调度方案。

[0042] 本发明的技术构思为: 在目前国内外作业车间调度问题现状分析的基础上, 考虑实际生产调度过程中的非确定性加工参数及动态扰动因素, 将作业车间调度问题从严格限定的理想环境拓展到贴近现实的复杂生产环境中, 使其具有灵活性和实用性。

[0043] 本文将加工参数非确定性和动态干扰事件发生的作业车间调度定义为复杂生产环境下的作业车间调度研究。正是由于复杂生产环境下的车间调度更符合实际的生产情况, 因此研究复杂生产环境下的作业车间调度问题, 更具有重要的理论意义和实用价值。

[0044] 作业车间调度问题是典型的 NP-hard 难题, 已有的作业车间调度问题的研究大都存在以下问题: 一是将生产系统中各种加工参数看作确定性的精确值; 二是将生产系统当作一个静态的系统, 忽略了实际加工过程中的各种突发状况。本文从复杂生产环境下生产运作和管理的实际需求出发, 考虑到生产过程中的加工参数非确定性精确值、动态扰动等因素的影响, 基于改进遗传算法开展了复杂生产环境下的作业车间调度问题的研究工作。

[0045] 遗传算法是一种借鉴生物界自然选择和进化机制发展起来的高度并行、随机、自适应搜索算法, 在问题求解、优化和搜索、智能控制、模式识别以及人工生命等领域均获得了令人鼓舞的成就。特别是对于那些传统搜索算法难以解决的复杂的和非线性的问题尤其适用。目前, 进化计算与人工神经网络、模糊系统理论一起已成为新的研究方向——计算智能。同样地, 在车间作业调度领域, 基于遗传算法甚至计算智能的研究也成为了研究的热点。

[0046] 模糊集合理论基础: 集合论是由德国数学家乔·康托 (George Cantor) 于 19 世纪末首先创立的。所谓集合, 一般是指具有某种特定属性的对象的全体。将组成集合的对象 (或事物) 称为元素, 这些元素对应于在域上的任何一个对象, 与集合之间的关系只能是属于或不属于的关系。随着人类社会的进步和科学技术的发展, 人们已经充分认识到利用 Cantor 的集合论是不能完全精确的描述复杂多样的现实世界, 这就要求将 Cantor 的集

合论扩充,即将一个对象是否属于某个集合的特征函数的取值不仅仅限制为 0 和 1 两个数,而应该是可以取 0 和 1 之间的任何一个值。即将仅在 $\{0,1\}$ 中取值的特征函数扩充为可在 $[0,1]$ 中取值的隶属函数。

[0047] G&T 算法 :G&T 算法的具体步骤如下 :

[0048] 步骤 1. 选择所有工件的第 1 道操作 (工序), 加入集合 C ;

[0049] 步骤 2. 假定机器在同一时刻可以加工任意多个作业, 计算集合 C 中各操作 $O_{ijk} \in C$ 的模糊完成时间, 记作 $(EC_{ijk}^1, EC_{ijk}^2, EC_{ijk}^3)$;

[0050] 步骤 3. 根据三角模糊数的比较准则, 确定集合 C 中最小模糊完成时间的操作 O_{i*j*k*} , 将与操作 O_{i*j*k*} 使用同一台机器 M_{k*} 的操作 $O_{ijk} \in C$ 构成集合 G。由于操作 O_{ijk} 与操作 O_{i*j*k*} 的操作时间上是有重叠的, 因此集合 G 叫做冲突集合 ;

[0051] 步骤 4. 从集合 G 中随机选择一个操作 $O_{ipjp*k*}$;

[0052] 步骤 5. 以上一步中所选择的操作为基准, 通过模糊数的取大操作 (对作业的加工开始时间的调整) 和模糊数的加法操作 (对作业的加工完成时间的调整) 依次更新冲突集合中其他操作的 EC^1, EC^2 和 EC^3 值 (因为每台机器同一时刻只能加工一个工件, 所以必须对步骤 2 所确定的模糊时间作调整)。从集合 C 中移去操作 $O_{ipjp*k*}$, 将其下一道工序加入集合 C, 并计算其相应模糊完成时间。

[0053] 步骤 6. 重复步骤 3-5, 直到所有的操作加工完毕。

[0054] 通过以上的步骤这样就可得到一条染色体, 执行 n 次 G&T 算法, 就可得到 n 条染色体。以此作为遗传算法的初始种群可以将问题的搜索空间限于活跃调度集, 从而可以提高遗传算法的运算效率。

[0055] 模拟退火算法 : 这里采用的模拟退火算法求解过程如下所述 :

[0056] 步骤 1. 在 G&T 算法的第 4 步随机选择产生一个调度方案 (当前解), 用 X^c 表示, 计算目标函数值 $F(x^c)$ 。设定初始温度为 T_0 。

[0057] 步骤 2. 用相应的矩阵表示当前解 X^c 的每台机器的工件加工顺序, 随机选择一台机器 M_k 。随机选择这台机器上的两个工件并进行互换。例如, 3×3 的调度问题中, 当机器 2 的第 1 个工件 (J_2) 和第 3 个工件 (J_3) 被选择时, 互换后的结果就如图 4(b) 所示。

[0058] 步骤 3. 通过上述工件交换后, 得到新的工件加工顺序, G&T 算法的第 4 步的个体冲突将得到解散, 并产生一个新的调度方案 (解)。若获得的解与工件互换前的解不一样, 则设当前解为一个解 X, 并进入下一步 ; 若否, 返回第 2 步选择一个新的交换对 ;

[0059] 步骤 4. 若当前解的目标函数值得到优化, 则接受互换, 并令 $X^c = X$ 。若否, 用下面的子步骤决定是否接受交换 ;

[0060] 1. 计算当前目标函数值 $F(x)$ 与目标函数值 $F(x^c)$ 的差值 Δf , 计算概率值 $\exp(-\Delta f/T)$;

[0061] 2. 生成 (0, 1) 之间的随机数, 并与 $\exp(-\Delta f/T)$ 相比 ;

[0062] 3. 若 $\exp(-\Delta f/T) > \text{random}(0, 1)$, 则接受互换, 令 $X^c = X$; 否则, 互换将不被接受 ;

[0063] 当互换接受时进入下一步。否则返回第 2 步寻找下一个交换对。

[0064] 步骤 5. 进行平衡态检验。检查经过一定次数交换后获得的目标函数值的变化是

否足够小。平衡态检验的次数定义为世代。按下面的四个步骤进行平衡态检验。

[0065] 1. 重复步骤 2 至 4, 直到交换达到世代的数量。然后进行以下操作;

[0066] 2. 计算当前代的目标函数值的平均值 \bar{f}_e 和经过交换后的目标函数值的平均值 \bar{f}'_e ;

[0067] 3. 比较 \bar{f}_e 和 \bar{f}'_e 的相对误差 $\frac{|\bar{f}_e - \bar{f}'_e|}{\bar{f}_e}$ 与给定公差值 ε 的大小;

[0068] 4. 当相对误差小于公差值时, 即 $\frac{|\bar{f}_e - \bar{f}'_e|}{\bar{f}_e} < \varepsilon$, 则认为在这个温度达到平衡态, 进入下一步。否则, 清除世代的计算, 回到第 2 步重复工件交换流程;

[0069] 步骤 6. 确定模拟退火的衰减温度, $T_{\text{new}} = \alpha \times T_{\text{old}}$, (α 为模拟退火衰减系数, 一般取 0.8-0.9 左右)

[0070] 步骤 7. 如果交换对数达到预定数量, 算法结束, 得到调度问题近似解。

[0071] 重复以上操作直到算法终止, 在获得的解中选择最优目标函数值的解。

[0072] 本发明的有益效果主要表现在: (1) 针对复杂的生产环境, 用模糊数来表示非确定性的加工参数。基于满意度概念, 以单个工件最小满意度和所有产品满意度为目标函数, 建立了模糊化参数的作业车间调度模型。

[0073] (2) 将 G&T 算法运用于遗传算法初始染色体的生成, 对所建立的模糊化参数模型进行了求解, 通过与模拟退火算法的比较, 验证了算法的有效性和稳定性, 对模型有较好的求解效果。

[0074] (3) 建立了基于模糊化参数的动态调度模型, 详细讨论了紧急插单工件的到来、机器损坏与修复、工件到期时间改变等三种动态扰动事件, 用改进遗传算法对模型进行验证, 并比较了动态事件发生后实行再调度和未实行再调度的调度方案优劣。

[0075] (4) 以 TYDQ 机加工车间为例, 结合车间实际生产情况, 建立了模糊化参数的动态调度模型, 通过与原始调度方案的比较, 获得了较好的结果。

附图说明

[0076] 图 1 是相似度计算示意图。

[0077] 图 2 是翻转变异算子的示意图。

[0078] 图 3 是种群构造的示意图。

[0079] 图 4 是 3×3 调度问题的工件加工顺序和互换的示意图, 其中该, (a) 为工件交换前; (b) 为工件交换后。

具体实施方式

[0080] 下面结合附图对本发明作进一步描述。

[0081] 参照图 1 ~ 图 4, 一种基于改进遗传算法的解动态扰动下的模糊化参数作业车间调度控制方法, 所述控制方法包括以下步骤:

[0082] 1)、确定模糊化参数作业车间调度的目标函数

[0083] 为了更贴切地反应作业车间调度最大化满足顾客需求和顾客对不同产品具有不同的满意度的情况, 提出模糊化参数的作业车间调度模型, 以最大化所有产品满意度和最大化最小满意度为目标函数, 即

$$[0084] \quad \max z_1 = \sum_{i=1}^n w_i AI_i \quad (1)$$

$$[0085] \quad \max z_2 = AI_{\min} = \min_{i=1, \Lambda, n} AI_i \quad (2)$$

[0086] 其中, $W = (w_1, w_2, \Lambda, w_n)$ 为各产品交货期满意度的权重系数, w_1 表示顾客对产品 i 交货期要求的重要程度, $\sum_{i=1}^n w_i = 1$ 。目标函数 z_1, z_2 分别表示所有产品满意度和最小满意度。

[0087] 令

$$[0088] \quad z' = y_1 z_1 + y_2 z_2 \quad (3)$$

[0089] $Y = (y_1, y_2)$ 为 z_1 和 z_2 的权重系数, $y_1 + y_2 = 1$ 。

[0090] 这样, 模糊作业车间调度问题的目标函数为: 求一个满足 z^* 的工件加工顺序, 使得:

$$[0091] \quad z^* = \max(z') \quad (4)$$

$$[0092] \quad \begin{cases} z' = y_1 z_1 + y_2 z_2 \\ \max z_1 = \sum_{i=1}^n w_i AI_i \\ \max z_2 = AI_{\min} = \min_{i=1, \Lambda, n} AI_i \\ \sum_{i=1}^n w_i = 1 \\ y_1 + y_2 = 1 \end{cases} \quad (4-1)$$

[0093] 2)、改进的遗传算法

[0094] (2.1) 编码

[0095] 合理地设计编码机制对遗传算法的质量和效益有很大影响, 在进行遗传算法编码时, 必须考虑染色体的合法性、可行性、有效性以及对问题解空间表征的完整性。

[0096] 这里采用基于工序的编码, 例如 3 工件 3 机器的问题, 假定染色体为 [1 2 3 11 3 2 2 3], 其中 1 代表工件 J_1 , 2 代表工件 J_2 , 3 代表工件 J_3 。由于每个工件有三道工序, 所以每个工件在染色体中出现三次。工件的工序对应工件在染色体中出现的顺序。如第一个 1 代表工件 J_1 的第 1 道工序, 第二个 1 代表工件 J_1 的第 2 道工序。每个工序的加工时间用三角模糊数表示。

[0097] 编码和解码时, 因动态车间调度采用了滚动窗口的思想, 故这里的编码和解码都是针对调度窗口中的工件进行。

[0098] (2.2) 初始种群的生成

[0099] 多次运行 G&T 算法可以产生一个初始种群。由于其 G&T 第 4 步是随机选择, 为保证初始种群的多样性, 要求初始种群的相似度不能太大。通过 Sakawa 的研究, 当个体相似度在 0.8 或以下的时候, 产生的初始种群有最大的稳定性。这里要求初始种群的个体相似度不能大于 0.8。

[0100] 以 4×4 的 JSP 为例介绍个体相似度的计算方法。个体 1 和个体 2 的每台机器上的工件加工序列表如图 3-4 所示。对于个体 1, 在机器 1 上, 工件的优先权分别是 1、2、4、3。

对于个体 2, 机器 1 上的工件优先权是 2、1、4、3。对比两个个体同机器上的工件顺序, 机器 1 上与工件 1 拥有相同的优先关系的工件是 3 和 4, 因此赋值为 2。同理, 机器 1 上与工件 2 拥有相同的优先关系的工件是 3 和 4, 因此赋值 2。机器 1 上与工件 3 拥有相同优先关系的工件有 1、2 和 4, 因此赋值 3, 机器 1 上与工件 4 拥有相同优先关系的工件是 1、2 和 3, 赋值 3。这样机器 1 上的工件赋值相加为 $2+2+3+3=10$ 。同样的方法处理另外机器, 计算过程如图 1 所示。通过这个方法, 可以得到个体 1 和个体 2 的相似度为 0.917。

[0101] (2.3) 适应度函数设计

[0102] 适应度函数用于对个体进行评价, 同时也是优化过程发展的依据。这里采用目标函数作为适应度函数对个体进行评价。

[0103] (2.4) 选择和交叉操作

[0104] 选择操作用于避免有效基因的丢失, 使得适应度大的个体得以更大的概率存活至下一代, 从而提高全局收敛性和计算效率。交叉操作用于产生一个新个体, 并保证子代尽量继承父代的优良性状。基于 G&T 算法, 采用下面的交叉操作从两个父代中产生一个子代。

[0105] 步骤 2.4.1. 执行 G&T 算法的第 1 至 3 步, 获得集合 C 和冲突集合 G;

[0106] 步骤 2.4.2. 以 50% 的相同概率从两个父代个体中任选一个个体。在冲突集合 G 中选择具有最小模糊完成时间的操作, 用选出的父代个体代替并表示为 O_{i_x, j_y, k^*} ;

[0107] 步骤 2.4.3. 执行 G&T 算法的第 5 和第 6 步。

[0108] 通过以上的操作得到了一个新的子代个体, 进行 c 次以上的操作就可以得到 c 个新子代个体。

[0109] 为保留由 c 个子代和 2 个父代组成的 (c+2) 个个体中具有优良性状的个体, 用以下的方法选择两个在下一次遗传时保留下来的个体。

[0110] a. 在 c 个子代个体中, 选择具有最大目标函数值的个体, 也即局部排名选择;

[0111] b. 在剩下的 (c+1) 个个体中 (2 个父代和 (c-1) 个子代), 选择具有最大目标函数值的个体。

[0112] 上述的交叉和选择操作, 由于更大的 c 值意味着更大的子代群体, 优良子代的产生概率将变得比较高。然而, 由于这些子代是由相同的父代产生的, 子代间的相似度也将变得比较高。因此在本算法中将 c 设定为 5。

[0113] (2.5) 变异操作

[0114] 当交叉操作产生的后代适应度不再进化, 不向好的方向发展, 就意味着算法的早熟收敛。变异操作就是在一定程度上克服这种情况。这里, 采用比较通用的反转变异 (Inversion Mutation), 即随机选择两个基因位置, 将此位置的基因相互交换。例如, 对于父代个体 parent1 选取基因位置为 1 和 6 的基因进行交换以达到变异的目的, 从而得到变异子代个体 son1, 如图 2 所示。

[0115] (2.6) 种群构造

[0116] 在遗传算法运行中, 必须防止进入局部最优解。为防止收敛到局部解, 采用 Sakawa 提出的种群构造方法^[71, 72]。假设有 3 组种群, 每组基于相似度产生 10 个个体。当每个组的个体均收敛到某一个度时, 将各组种群合并继续进化直到收敛, 如图 3 所示。

[0117] (2.7) 算法终止条件

[0118] 由于在实际问题中使用遗传算法, 是绝不允许让其无休止搜索下去的, 同时问题

的最优解也通常未知,因此必须设计一些近似收敛准则来终止算法进程。这里以预先设定的最大进化代数 N_{\max} 作为停止条件,将目前为止最好的解作为最优解。

[0119] 3) 在动态事件发生时进行再调度:

[0120] 步骤 1. 初始化调度次数 $n = 0$, 开始调度的时间 $t_0 = 0$, 给定窗口中工件的最大数目 L_w , 将待加工工件放入窗口内, 设工件窗口为 N_w , 已完成工件集为 N_c 。

[0121] 步骤 2. 对当前窗口内的工件执行改进遗传算法, 选择使目标函数最优的染色体作为调度结果, 产生调度方案;

[0122] 步骤 3. 当有以下动态事件发生时, 执行一次再调度, 设动态事件发生时刻为 t_c ;

[0123] ①当有插单工件到来时启动一次再调度, 转步骤 4;

[0124] ②当工件到期时间改变时启动一次再调度, 转步骤 4;

[0125] ③当机器损坏事件发生时启动一次再调度, 转步骤 4;

[0126] 步骤 4. 修改可利用设备项, 将已完成的工序从工件窗口 N_w 迁入已完成工件集 N_c , 产生已完成工序的调度方案; 未完成工序的顺序提前, 将剩余工序作为未调度的新工序。

[0127] ①将插单工件加入工件窗口 N_w , 和未完成工序组成新的待加工工序, 执行再调度。若插单工件到来时, 某机器正在加工某道工序, 则自动延迟一定时间, 该道工序完成后执行再调度;

[0128] ②工件到期时间改变时, 更新工件窗口内的工件信息, 执行再调度;

[0129] ③机器损坏时, 将正在该机器上加工的工件移出工件窗口, 对剩下的工件执行再调度。当仿真时钟运行到剩余工件需在该机器加工的工序时, 将此工件移出工件窗口, 对剩余工件执行再调度。机器修复时, 将移出工件窗口的工件重新迁入, 执行再调度。

[0130] 按照再调度事件重复进行步骤 4, 直到所有的工件加工完成, 产生最终调度方案。

[0131] 本方案在上一方案-模糊化参数调度模型的基础上, 针对实际作业车间的动态扰动情况, 考虑了工件到期时间改变、插单工件到来和机器损坏等三个动态事件, 建立模糊化参数的动态调度模型, 用改进遗传算法来进行求解, 最后用算例演算和验证, 并与未再调度的方案进行比较, 结果表明模型能体现车间实际的动态事件, 并产生较好的调度方案。

[0132] 该方法的实际运用:

[0133] 步骤 1. 别用 M_1, M_2, \dots, M_{11} 表示各种生产机器。

[0134] 步骤 2. 用模糊数表示不同产品在划线工作站的时间。

[0135] 步骤 3. 用模糊数表示不同产品的安装和拆卸时间。

[0136] 步骤 4. 用模糊数表示不同产品对应机械的加工时间。

[0137] 步骤 5. 用模糊数表示不同产品的油封装箱时间。

[0138] 步骤 6. 计算产品完工时间。

[0139] 步骤 7. 得到产品交货期(天改成小时)。

[0140] 步骤 8. 调度模型的建立

[0141] 建立实例调度模型前, 先确定一些基本参数:

[0142] t_{ih} - 工件划线工作站时间;

[0143] t_{ip} - 工件加工时间(包含安装时间和拆卸时间);

[0144] t_{if} - 油封装箱时间;

[0145] 这样建立机加工车间的模糊化参数下的动态调度模型如下:

$$[0146] \quad z^* = \max(z') \quad (1)$$

$$[0147] \quad \left\{ \begin{array}{l} z' = y_1 z_1 + y_2 z_2 \\ \max z_1 = \sum_{i=1}^n w_i AI_i \\ \max z_2 = AI_{\min} = \min_{i=1, \Delta n} AI_i \\ AI_i = \frac{area(\tilde{C}_i \cap \tilde{D}_i)}{area \tilde{C}_i} \\ \tilde{C}_i = t_{ih} + t_{ip} + t_{if} \\ \sum_{i=1}^n w_i = 1 \\ y_1 + y_2 = 1 \end{array} \right. \quad (4-2)$$

[0148] 在该模型中, 其中, $\tilde{C}_i = t_{ih} + t_{ip} + t_{if}$ 与前面模型中通用的加工完成时间 \tilde{C}_i 是一致的。在实例模型中是将加工完成时间进行了具体化处理, 分为划线时间、机加工时间和油封装箱时间。模型中, y_1 设为 0.7, y_3 设为 0.3, 表示企业着重追求整体产品的满意度最大。

[0149] 步骤 9. 确定客户对各产品的满意度评价值。

[0150] 步骤 10. 确定各产品满意度权重系数。

[0151] 步骤 11. 动态事件描述

[0152] 11.1 确定机器的损坏和修复模糊化时间

[0153] 11.2 紧急插单工件到来时间模糊化

[0154] 步骤 13. 模型的仿真及结果分析。

[0155] 当系统发生插单工件到来动态事件时, 若未作再调度则最优目标函数值为 0.058, 而再调度后最优目标函数值为 0.181。

[0156] 当系统发生机器损坏与修复动态事件时, 若未作再调度则最优目标函数值为 0.093, 再调度后最优目标函数值为 0.234。

[0157] 本实施例的模型在较高的目标函数值的情况下能够获得良好的体现动态特性的调度方案。机器损坏和修复、插单工件到来都能在调度方案中考虑进去, 符合了企业的生产实际。

	Job						Job				
Machine1	1	2	4	3	2+2+3+3=10		Machine1	2	1	4	3
Machine2	3	1	4	2	3+3+3+3=12		Machine2	3	1	4	2
Machine3	4	3	1	2	3+3+3+3=12		Machine3	4	3	1	2
Machine4	3	2	1	4	3+3+2+2=10		Machine4	3	2	4	1

$$\frac{10+12+12+10}{12*4} = \frac{44}{48} = 0.917$$

图 1

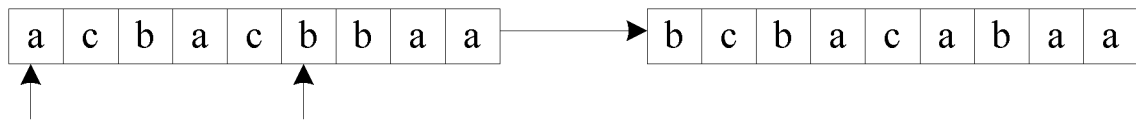


图 2

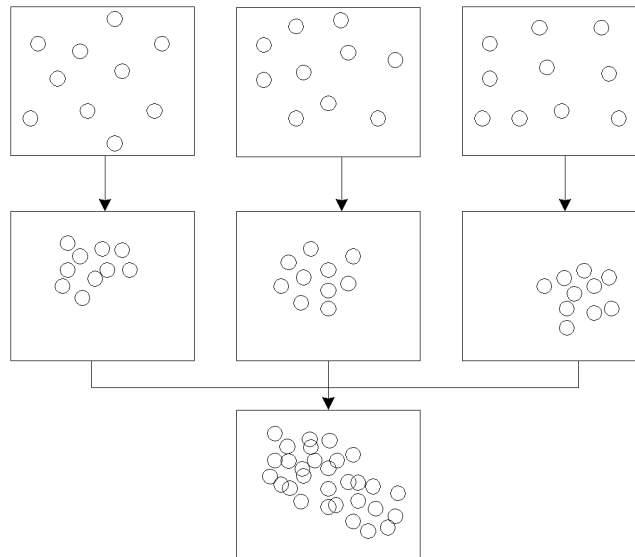


图 3

$$\begin{matrix} M_1 \\ M_2 \\ M_3 \end{matrix} \begin{pmatrix} J_1 & J_2 & J_3 \\ J_2 & J_1 & J_3 \\ J_2 & J_3 & J_1 \end{pmatrix} \Rightarrow \begin{matrix} M_1 \\ M_2 \\ M_3 \end{matrix} \begin{pmatrix} J_1 & J_2 & J_3 \\ J_3 & J_1 & J_2 \\ J_2 & J_3 & J_1 \end{pmatrix}$$

(a) 工件交换前 (b) 工件交换后

图 4