



(12) 发明专利申请

(10) 申请公布号 CN 101901425 A

(43) 申请公布日 2010. 12. 01

(21) 申请号 201010226408. 2

(22) 申请日 2010. 07. 15

(71) 申请人 华中科技大学

地址 430074 湖北省武汉市洪山区珞喻路
1037 号(72) 发明人 李新宇 高亮 邵新宇 张利平
王晓娟(74) 专利代理机构 华中科技大学专利中心
42201

代理人 朱仁玲

(51) Int. Cl.

G06Q 10/00 (2006. 01)

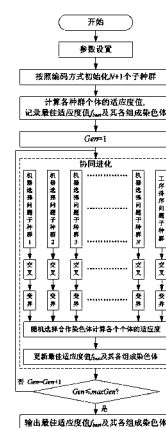
权利要求书 2 页 说明书 7 页 附图 2 页

(54) 发明名称

一种基于多种群协同进化的柔性作业车间调度方法

(57) 摘要

本发明提供了一种基于多种群协同进化的柔性作业车间调度方法,它属于车间调度领域,主要克服基于遗传算法的柔性作业车间调度方法性能得不到充分发挥的不足。其主要步骤包括:(1) 参数设置;(2) 按照设定的编码方式初始化种群;(3) 按照设定的方法计算各个种群中每个染色体的适应度值,并记录最佳适应度值及其各组成染色体;(4) 多种群协同进化:对各子种群中的染色体进行交叉和变异的进化操作,并对新染色体进行评价;(5) 判断方法终止准则是否达到,若满足,则方法结束,输出最佳适应度值及其各组成染色体;否则跳转到步骤4。本发明可以获得高质量的适合车间实际生产的调度方案,缩短生产时间,可用于车间生产过程的调度管理与优化。



1. 一种基于多种群协同进化的柔性作业车间调度方法,用于获得适合车间实际生产的调度方案,实现车间生产过程的调度管理与优化,该方法具体包括如下步骤:

(1) 假设工件个数为 N ,则设置 $N+1$ 个种群,其中包括 N 个机器选择子问题的种群和 1 个工序排序子问题的种群;

(2) 根据各种群的编码方式,随机初始化上述 $N+1$ 个种群;

(3) 评价上述经初始化后的种群,并令迭代次数 $Gen = 1$;

计算经初始化后的各个种群中每个染色体的适应度值,即针对每个种群的每个染色体,从 N 个合作种群的每一个中都随机选择一个染色体,这 $N+1$ 个染色体组成一个完整解,计算该完整解的适应度,以作为当前染色体的适应度值;进而,获得种群中染色体的最佳适应度值 f_{best} 及其对应的组成完整解的染色体组;

(4) 设种群编号为 q ,初始化 $q = 1$,对上述 $N+1$ 个种群进行多种群协同进化,具体过程为:

(4.1) 基于锦标赛选择算子在种群 q 中选择两个染色体作为父代执行交叉操作,生成两个子代染色体;其中;

(4.2) 计算上述父代和子代染色体的适应度值,其中父代和子代染色体具有相同的合作染色体,从这四个染色体中选择两个适应度值较优的染色体进入下一代种群中;

(4.3) 在种群 q 中随机选择一个父代染色体执行变异操作,生成一个子代,用该子代取代该父代进入下一代种群中;

(4.4) 计算种群 q 的染色体适应度值

针对种群 q 的每个染色体,从该种群 q 的 N 个合作种群的每一个中都随机选择一个染色体,与种群 q 的该染色体组成一个完整解,计算该完整解的适应度,以作为当前染色体的适应度值;进而,从种群 q 的多个染色体获得的多个适应度中选择最佳的适应度值以更新所述最佳适应度值 f_{best} ,同时更新该适应度值所对应的组成完整解的染色体组;

(4.5) 令 $q = q+1$,若 $q \leq N+1$,则跳转到步骤 (4.1) 进行下一代种群的处理,否则进入步骤 (5);

(5) 判断迭代次数 Gen 是否达到规定的阈值 $maxGen$,若是,则算法结束,输出最后得到的最佳适应度值 f_{best} 及其对应的组成完整解的染色体组,否则令 $Gen = Gen+1$,并跳转到步骤 (4) 继续处理。

2. 根据权利要求 1 所述的一种基于多种群协同进化的柔性作业车间调度方法,其特征在于,所述的机器选择子问题的种群编码方式为:该种群的染色体长度为对应工件的工序总数,每个基因位用整数表示,依次按照该工件工序的顺序进行排列,每个整数表示当前工序选择的加工机器在可选机器集中的顺序编号。

3. 根据权利要求 1 或 2 所述的一种基于多种群协同进化的柔性作业车间调度方法,其特征在于,所述工序排序子问题的种群编码方式基于工序的编码方法实现,具体为:该种群的染色体长度等于所有工件的工序数之和,每一个基因用工件号直接编码,工件号出现的顺序表示该工件工序间的先后加工顺序,并且工件号的出现次数等于该工件的工序总数。

4. 根据权利要求 1-3 之一所述的一种基于多种群协同进化的柔性作业车间调度方法,其特征在于,针对所述机器选择子问题的交叉操作为均匀交叉操作。

5. 根据权利要求 1-4 之一所述的一种基于多种群协同进化的柔性作业车间调度方法,

其特征在于,针对工序排序子问题的种群的交叉操作采用基于工序编码的交叉算子 POX,具体为:首先将工件集 $\{J_1, J_2, \dots, J_n\}$ 随机划分为两个工件集 Jobset1 和 Jobset2;然后复制父代染色体 Parent1 和 Parent2 中包含在工件集 Jobset1 和 Jobset2 中的工件到 Offspring1 和 Offspring2,保持它们的位置和顺序;最后将 Parent1 和 Parent2 中不包含在工件集 Jobset1 和 Jobset2 中的工件复制到 Offspring2 和 Offspring1,保持它们的顺序不变。

6. 根据权利要求 1-5 之一所述的一种基于多种群协同进化的柔性作业车间调度方法,其特征在于,针对机器选择子问题的种群的变异操作采用如下的变异算子,即首先在变异染色体中随机选择多个位置,依次选择每一个位置,对每一个位置的机器设置为当前工序可选机器集中加工时间最短的机器。

7. 根据权利要求 1-6 之一所述的一种基于多种群协同进化的柔性作业车间调度方法,其特征在于,针对工序排序子问题的种群的变异操作,采用基于邻域搜索变异操作,即首先在变异染色体中随机选择多个不同基因,并生成其排序的所有邻域,然后评价所有邻域的适应值,选出最佳染色体作为子代。

一种基于多种群协同进化的柔性作业车间调度方法

技术领域

[0001] 本发明属于车间调度领域,具体涉及一种柔性作业车间调度方法,用于车间生产过程的调度管理与优化。

背景技术

[0002] 车间生产调度是制造系统的基础,生产调度的优化是先进制造技术和现代管理技术的核心。有关资料表明,制造过程中 95% 的时间消耗在非切削过程中,因此科学地制定生产调度方案,对于缩短产品生产周期,控制车间的在制品库存,提高产品交货期满足率和提高企业生产率起着至关重要的作用,对于实现先进制造企业的现代化具有重要的理论价值和实际意义。

[0003] 传统车间调度模型中,假设工序加工所需要的资源是不具有柔性的,但是随着大批量连续生产时代正逐渐被适应市场动态变化的多品种、小批量离散生产所替代,一个制造企业的生存能力和竞争能力在很大程度上取决于它是否能在较短的生产周期内,生产出低成本、高质量的多品种产品的能力。随着柔性制造系统和数控加工中心等一些柔性生产系统的出现。以往作业车间调度的研究成果,无法适应新的制造模式。柔性作业车间调度问题也就成为了研究重点,它是作业车间调度问题的一种扩展,工件的每道工序可以在多台机器上进行加工,加工时间不一定相同。工件的每道工序可以在多台机器上加工的情况在实际生产中有明显的优点:(1) 提高设备的利用率,机床一旦空闲就可以安排工件进行加工,减少设备闲置和等待的时间;(2) 具有维持生产稳定的能力,当一台或多台机器发生故障时,工序可以绕过故障机器,在其它机器上进行加工,生产得以继续,保证生产的稳定;(3) 提高产品质量和缩短生产周期,同一工件的多个工序可以在同一台机床上连续进行加工,减少了中间装卸和搬运等造成时间的消耗。从以上的描述可以看出,柔性作业车间调度问题包含两个子问题:确定各工件的加工机器(机器选择子问题)和确定各个机器上的加工先后顺序(工序排序子问题)。另一方面,柔性作业车间调度问题是典型的 NP-hard 问题,一直吸引着学者的广泛关注,是制造系统领域的研究热点之一。

[0004] 柔性作业车间调度问题在 1990 年被首次提出,典型的求解方法主要有:整数规划、混合整数规划、拉格朗日松弛法、分解方法、分枝定界法和启发式规则等。这些方法只能求解小规模柔性作业车间调度问题,无法应用于实际生产。近年来随着计算智能方法的发展,在该领域出现了如禁忌搜索、模拟退火、蚁群算法、遗传算法、粒子群优化算法、人工免疫算法等新方法,这些新方法能够较好地兼顾求解速度和质量,能够在合理的时间内找到问题尽可能好的解。

[0005] 基于遗传算法的柔性作业车间调度方法是模拟达尔文主义的自然选择和生物遗传机制构造的随机搜索方法。遗传算法具有较强的全局搜索能力,然而在求解柔性作业车间调度问题时还存在如下问题:柔性作业车间调度问题包含两个子问题(机器选择子问题和工序排序子问题),这两个子问题具有不同的特性,如果将两者一起优化,问题的解空间较复杂,算法的性能无法得到充分的发挥,也难以找到高质量的调度方案。

发明内容

[0006] 本发明的目的在于克服基于遗传算法的柔性作业车间调度方法性能得不到充分发挥的不足,提供一种基于多种群协同进化的柔性作业车间调度方法,以获得高质量的适合车间实际生产的调度方案,缩短生产时间。

[0007] 为实现上述目的,本发明的技术方案具体为:

[0008] 一种基于多种群协同进化的柔性作业车间调度方法,用于获得适合车间实际生产的调度方案,实现车间生产过程的调度管理与优化,该方法具体包括如下步骤:

[0009] (1) 假设工件个数为 N ,则设置 $N+1$ 个种群,其中包括 N 个机器选择子问题的种群和 1 个工序排序子问题的种群;

[0010] (2) 根据各种群的编码方式,随机初始化上述 $N+1$ 个种群;

[0011] (3) 评价上述经初始化后的种群,并令迭代次数 $Gen = 1$;

[0012] 计算经初始化后的各个种群中每个染色体的适应度值,即针对每个种群的每个染色体,从 N 个合作种群的每一个中都随机选择一个染色体,这 $N+1$ 个染色体组成一个完整解,计算该完整解的适应度,以作为当前染色体的适应度值;进而,获得种群中染色体的最佳适应度值 f_{best} 及其对应的组成完整解的染色体组;

[0013] (4) 设种群编号为 q ,初始化 $q = 1$,对上述 $N+1$ 个种群进行多种群协同进化,具体过程为:

[0014] (4.1) 基于锦标赛选择算子在种群 q 中选择两个染色体作为父代执行交叉操作,生成两个子代染色体;其中;

[0015] (4.2) 计算上述父代和子代染色体的适应度值,其中父代和子代染色体具有相同的合作染色体,从这四个染色体中选择两个适应度值较优的染色体进入下一代种群中;

[0016] (4.3) 在种群 q 中随机选择一个父代染色体执行变异操作,生成一个子代,用该子代取代该父代进入下一代种群中;

[0017] (4.4) 计算种群 q 的染色体适应度值

[0018] 针对种群 q 的每个染色体,从该种群 q 的 N 个合作种群的每一个中都随机选择一个染色体,与种群 q 的该染色体组成一个完整解,计算该完整解的适应度,以作为当前染色体的适应度值;进而,从种群 q 的多个染色体获得的多个适应度中选择最佳的适应度值以更新所述最佳适应度值 f_{best} ,同时更新该适应度值所对应的组成完整解的染色体组;

[0019] (4.5) 令 $q = q+1$,若 $q \leq N+1$,则跳转到步骤 (4.1) 进行下一代种群的处理,否则进入步骤 (5);

[0020] (5) 判断迭代次数 Gen 是否达到规定的阈值 $maxGen$,若是,则算法结束,输出最后得到的最佳适应度值 f_{best} 及其对应的组成完整解的染色体组,否则令 $Gen = Gen+1$,并跳转到步骤 (4) 继续处理。

[0021] 作为本发明的进一步改进,所述的机器选择子问题的种群编码方式为:该种群的染色体长度为对应工件的工序总数,每个基因位用整数表示,依次按照该工件工序的顺序进行排列,每个整数表示当前工序选择的加工机器在可选机器集中的顺序编号。

[0022] 作为本发明的进一步改进,所述工序排序子问题的种群编码方式基于工序的编码方法实现,具体为:该种群的染色体长度等于所有工件的工序数之和,每一个基因用工件号

直接编码,工件号出现的顺序表示该工件工序间的先后加工顺序,并且工件号的出现次数等于该工件的工序总数。

[0023] 作为本发明的进一步改进,针对所述机器选择子问题的交叉操作为均匀交叉操作。

[0024] 作为本发明的进一步改进,针对工序排序子问题的种群的交叉操作采用基于工序编码的交叉算子 POX,具体为:首先将工件集 $\{J_1, J_2, \dots, J_n\}$ 随机划分为两个工件集 Jobset1 和 Jobset2;然后复制父代染色体 Parent1 和 Parent2 中包含在工件集 Jobset1 和 Jobset2 中的工件到 Offspring1 和 Offspring2,保持它们的位置和顺序;最后将 Parent1 和 Parent2 中不包含在工件集 Jobset1 和 Jobset2 中的工件复制到 Offspring2 和 Offspring1,保持它们的顺序不变。

[0025] 作为本发明的进一步改进,针对机器选择子问题的种群的变异操作采用如下的变异算子,即首先在变异染色体中随机选择多个位置,依次选择每一个位置,对每一个位置的机器设置为当前工序可选机器集中加工时间最短的机器。

[0026] 作为本发明的进一步改进,针对工序排序子问题的种群的变异操作,采用基于邻域搜索变异操作,即首先在变异染色体中随机选择多个不同基因,并生成其排序的所有邻域,然后评价所有邻域的适应值,选出最佳染色体作为子代。

[0027] 本发明具有下列区别于传统方法的显著优势:

[0028] (1) 本发明由于引入多种群协同进化,使得柔性作业车间调度问题的两个子问题(机器选择子问题和工序排序子问题)能够进行分开优化,这避免了将两者集成优化,问题解空间过于复杂,算法性能无法得到充分的发挥的缺点。本发明将柔性作业车间调度问题分解成两个子问题进行多种群协同优化,每个种群所处理问题的解空间较简单,算法更容易找到高质量的解。

[0029] (2) 本发明由于引入了记忆染色体,并在算法运行过程中各种群的染色体在寻找其他各种群的合作染色体时,采用的是随机选择的方法,这样能很好的保持各种群的多样性,使得算法能更好地寻找到全局最优解。

附图说明

[0030] 图 1 本发明的实现流程图;

[0031] 图 2 本发明的多种群协同进化图。

具体实施方式

[0032] 下面结合附图和具体实施例对本发明作进一步说明。

[0033] 参照图 1 和图 2,一种基于多种群协同进化的柔性作业车间调度方法,包括如下步骤:

[0034] 步骤 1,参数设置:

[0035] 假设工件个数为 N ,则本方法包含 $N+1$ 个子种群,其中 N 个机器选择子问题的种群,1 个工序排序子问题的种群规模。设置算法循环迭代的次数阈值 maxGen,机器选择子问题的种群规模 P_1 ,锦标赛选择因子 k_1 ,交叉概率为 P_{c1} ,变异概率为 P_{m1} ,锦标赛选择因子 k_2 ,工序排序子问题的种群规模为 P_2 ,交叉概率 P_{c2} ,变异概率 P_{m2} 。

[0036] 步骤 2, 初始化种群 :

[0037] 按照设定的编码方式, 随机初始化 N 个规模为 P_1 的机器选择子问题的种群, 初始化 1 个规模为 P_2 的工序排序子问题的种群 ;

[0038] 本实施例中, 针对机器选择子问题的编码方式为 : 该部分染色体长度为对应工件的工序总数。每个基因位用整数表示, 依次按照该工件工序的顺序进行排列, 每个整数表示当前工序选择的加工机器在可选机器集中的顺序编号 (注意 : 这个编号不是对应的机器编号)。该编码方式首先能保证后续交叉、变异等操作后产生的解仍然是可行解, 其次对工序可选机器个数的多少没有要求, 长度确定, 操作方便 ;

[0039] 针对工序排序子问题的编码方式 : 从柔性作业车间调度问题的定义可知, 当所有工件的工序所对应的机器确定后, 其和作业车间调度问题的工序排序问题一样, 所以, 针对工序排序子问题的编码方式为基于工序的编码方法。其染色体长度等于所有工件的工序数之和, 每一个基因用工件号直接编码, 工件号出现的顺序表示该工件工序间的先后加工顺序, 即对染色体从左到右进行编译, 对于第 j 次出现的工件号, 表示该工件 i 的第 j 道工序 o_{ij} , 并且工件号的出现次数等于该工件的工序总数。该编码柔性很高, 可满足调度规模变化、工件工序数不定等各种复杂情况, 而且任意置换染色体中的顺序后总能得到可行调度, 在解码过程中可以产生活动调度。

[0040] 步骤 3, 评价初始化种群 :

[0041] 计算各个种群中每个染色体的适应度值 : 即从其他各合作种群中随机选择一个染色体来组成一个完整解并计算其适应度, 作为当前染色体的适应度值 (如图 2 所示), 并记录最佳适应度值 f_{best} (最佳适应度值即为当前种群中使得评价函数最优的值) 及其各组成染色体, 设置迭代次数 $Gen = 1$ 。

[0042] 步骤 4 : 多种群协同进化 (种群编号为 q , 取值范围内为从 1 到 $N+1$ 的整数, 初始化 $q = 1$, 如图 2 所示) :

[0043] 对各子种群中的染色体进化操作包括交叉操作和变异操作两个步骤, 由于机器选择子问题和工序排序子问题属于两个不同的问题, 在编码方式上也不一样, 所以在交叉和变异操作上也是不一样的。

[0044] 步骤 4.1 : 基于锦标赛选择算子在 q 种群中选择两个染色体作为父代执行交叉操作, 生成两个子代染色体 ;

[0045] 针对机器选择子问题的交叉操作 : 为了保证每位基因的先后顺序保持不变, 采用均匀交叉操作 ;

[0046] 针对工序排序子问题的交叉操作 : 为了得到较好的搜索效果, 采用一种基于工序编码的交叉算子 POX (Precedence Operation Crossover), 它能很好地继承父代优良特征。首先将工件集 $\{J_1, J_2, \dots, J_n\}$ 随机划分为两个工件集 Jobset1 和 Jobset2 ; 然后复制父代染色体 Parent1 和 Parent2 中包含在工件集 Jobset1 和 Jobset2 中的工件到 Offspring1 和 Offspring2, 保持它们的位置和顺序 ; 最后将 Parent1 和 Parent2 中不包含在工件集 Jobset1 和 Jobset2 中的工件复制到 Offspring2 和 Offspring1, 保持它们的顺序不变。

[0047] 步骤 4.2 : 计算父代和子代染色体的适应度值 (选择和父代染色体相同的合作染色体), 从这四个染色体 (包括两个父代染色体和两个子代染色体, 即 Parent1、Parent2、Offspring1 和 Offspring2) 中选择两个较优的染色体进入下一代种群中, 其中, 较优的染

染色体指适应度值较优的染色体,即能够使目标函数更优的适应度值。

[0048] 步骤 4.3:在 q 种群中随机选择一个父代染色体执行变异操作,生成一个子代,用子代取代父代进入下一代种群中;

[0049] 针对机器选择子问题的变异操作:为保证能够较好的保持优良染色体的信息和机器顺序不被破坏,设计如下的变异算子。首先在变异染色体中随机选择 r 个位置(本实例中 r 为当前工件工序数的一半取整);依次选择每一个位置,对每一个位置的机器设置为当前工序可选机器集中加工时间最短的机器;

[0050] 针对工序排序子问题的变异操作:采用基于邻域搜索变异操作,可以更好地通过局部范围内的搜索找到适合的工序排序,改善子代性能。首先在变异染色体中随机选择 1 个不同基因(一般的 $l = 3$),并生成其排序的所有邻域;然后评价所有邻域的适应值,选出最佳染色体作为子代。

[0051] 步骤 4.4:染色体适应度值的计算:从其他各合作种群中随机选择一个合作染色体来组成一个完整解并计算适应度,更新最佳适应度值 f_{best} 及其各组成染色体。

[0052] 步骤 4.5:设置 $q = q+1$;若 $q \leq N+1$,则跳转到第 4.1 步;否则跳转到第 5 步。

[0053] 步骤 5:判断方法结束准则是否已经达到($Gen \leq \max Gen$?),若满足,则算法结束,输出最佳适应度值 f_{best} 及其各组成染色体,否则 $Gen = Gen+1$,并跳转到第 4 步。

[0054] 本发明的效果可以通过以下仿真实验进一步说明:

[0055] (1) 仿真实验所采用的标准测试数据集

[0056] 采用以下 2 组柔性作业车间调度问题的标准实例集来检验本发明的有效性:

[0057] a) 10 个由 Brandimarte 提出的实例(Bdata)。10 个问题由工件数从 10 到 20,机器数从 4 到 15,每组问题的工序数从 5 到 15;

[0058] b) 21 个由 Barnes 和 Chambers 提出的实例(BCdata)。21 个实例问题主要对经典作业车间调度问题中最困难的三个问题(mt10, 1a24, 1a40)通过一定的原则构造而来。工件数从 10 到 15,机器数从 11 到 18。

[0059] (2) 仿真实验参数设置

[0060] 算法循环迭代的次数阈值 $\max Gen = 200$,机器选择子问题的种群规模 P_1 ,锦标赛选择因子 $k_1 = 2$,交叉概率为 $P_{c1} = 0.9$,变异概率为 $P_{m1} = 0.1$,锦标赛选择因子 $k_2 = 2$,工序排序子问题的种群规模为 P_2 ,交叉概率 $P_{c2} = 0.9$,变异概率 $P_{m2} = 0.1$;由于问题规模、复杂程度的不等,在种群规模上也有所不同,经过对每个问题的多次运算而得到每个问题的较佳的种群规模,一般从 100 到 500 不等, P_1 和 P_2 在仿真结果表中给出。

[0061] (3) 仿真实验环境

[0062] 本发明采用 C++ 语言编程实现,程序运行环境为:P4CPU,主频 2.4G,内存为 1GB。

[0063] (4) 仿真内容:本发明与其他算法的性能比较

[0064] 该实验将本发明的多种群协同进化方法(Multi-Population Cooperative Evolutionary Algorithm, MPCEA)与其他算法进行性能比较验证本发明的有效性。为了尽量消除由于算法的随机性而带来的计算误差,使得计算结果更具有有效性和一般性,对每个问题实例连续运行 10 次。其中, $n \times m$ 表示问题的工件数和机器数, LB 和 UB 分别表示到目前为止已知的最优下界和最优上界, C_{max} 表示算法取得的最优值,其表示最大完工时间的最优解, Aver. 表示 10 次运行的平均结果,本发明的计算结果已标出。

[0065] 表 1MPCEA 针对 Bdata 的运行结果

| | 问题 | $n \times m$ | (LB, UB) | LEGA | | MPCEA | | |
|--------|------|--------------|------------|------------|---------|-------------|------------|---------|
| | | | | C_{\max} | $Aver.$ | $P_1 = P_2$ | C_{\max} | $Aver.$ |
| [0066] | mk01 | 10×6 | (36, 42) | 40 | 41.5 | 100 | 40 | 40 |
| | mk02 | 10×6 | (24, 32) | 29 | 29.1 | 100 | 26 | 26.2 |
| | mk03 | 15×8 | (204, 211) | N/A | N/A | 100 | 204 | 204 |
| | mk04 | 15×8 | (48, 81) | 67 | 67.34 | 100 | 60 | 60 |
| | mk05 | 15×4 | (168, 186) | 176 | 178.1 | 200 | 173 | 175 |
| | mk06 | 10×15 | (33, 86) | 67 | 68.82 | 200 | 58 | 58 |
| | mk07 | 20×5 | (133, 157) | 147 | 152.9 | 200 | 141 | 143 |
| [0067] | mk08 | 20×10 | (523) | 523 | 523.34 | 200 | 523 | 523 |
| | mk09 | 20×10 | (299, 369) | 320 | 327.74 | 200 | 307 | 307 |
| | mk10 | 20×15 | (165, 296) | 229 | 235.72 | 200 | 200 | 204 |

[0068] 注 :N/A 表示该方法没有提供该数据

[0069] 表 1 比较了 MPCEA 与 LEGA 的调度性能。LEGA 是 Ho 等将机器学习机制和遗传算法进行混合提出的方法。从表 1 中可以看出本发明在所有 10 个问题上都取得了比较好的结果。和 LEGA 的最优值相比, MPCEA 在其中在 8 个问题上优于 LEGA 得到的最优值, 2 个问题是一样的, 其中本发明的 $Aver.$ 值也比 LEGA 的小, 即表示本发明具有良好的稳定性。

[0070] 表 2MPCEA 针对 BCdata 的运行结果

| 问题 | $n \times m$ | (LB, UB) | M&G 算法 | | MPCEA | | |
|---------|--------------|------------|------------|--------|-------------|------------|-------|
| | | | C_{\max} | Aver. | $P_1 = P_2$ | C_{\max} | Aver. |
| mt10c1 | 10×11 | (655, 927) | 928 | 928 | 200 | 927 | 927 |
| mt10cc | 10×12 | (655, 914) | 910 | 910 | 200 | 910 | 910 |
| mt10x | 10×11 | (655, 929) | 918 | 918 | 200 | 918 | 918 |
| mt10xx | 10×12 | (655, 929) | 918 | 918 | 200 | 918 | 918 |
| mt10xxx | 10×13 | (655, 936) | 918 | 918 | 500 | 918 | 918 |
| mt10xy | 10×12 | (655, 913) | 906 | 906 | 500 | 905 | 905 |
| mt10xyz | 10×13 | (655, 849) | 847 | 850 | 500 | 847 | 849 |
| setb4c9 | 15×11 | (857, 924) | 919 | 919.2 | 500 | 914 | 914 |
| setb4cc | 15×12 | (857, 909) | 909 | 911.6 | 500 | 909 | 910 |
| [0071] | setb4x | 15×11 | 925 | 925 | 500 | 925 | 925 |
| | setb4xx | 15×12 | 925 | 926.4 | 500 | 925 | 925 |
| | setb4xxx | 15×13 | 925 | 925 | 500 | 925 | 925 |
| | setb4xy | 15×12 | 916 | 916 | 500 | 910 | 912 |
| | setb4xyz | 15×13 | 905 | 908.2 | 500 | 905 | 909 |
| | seti5c12 | 15×16 | 1174 | 1174.2 | 500 | 1174 | 1174 |
| | seti5cc | 15×17 | 1136 | 1136.4 | 500 | 1136 | 1138 |
| | seti5x | 15×16 | 1201 | 1203.6 | 500 | 1201 | 1204 |
| | seti5xx | 15×17 | 1199 | 1200.6 | 500 | 1204 | 1204 |
| | seti5xxx | 15×18 | 1197 | 1198.4 | 500 | 1200 | 1204 |
| | seti5xy | 15×17 | 1136 | 1136.4 | 500 | 1136 | 1136 |
| | seti5xyz | 15×18 | 1125 | 1126.6 | 500 | 1125 | 1127 |

[0072] 表 2 比较了 MPCEA 与 M&G 算法的调度性能。M&G 算法是 Mastrolilli 和 Gambardella 利用禁忌搜索求解柔性作业车间调度的方法。从表 2 中可以看出本发明在 21 个问题上有 19 个问题取得了比较好的结果。和 M&G 算法的最优值相比, MPCEA 在其中的 4 个问题上优于 M&G 算法得到的最优值, 15 个问题是一样的, 2 个问题较 M&G 算法差。其中本发明的大部分 Aver. 值也比 M&G 算法的小, 即表示本发明具有较好的稳定性。

[0073] 通过以上的 2 组柔性作业车间调度问题的标准实例集的测试结果来看, 本发明在绝大部分问题的求解结果上都优于其他方法, 这说明本发明在求解柔性作业车间调度问题上具有优越的性能, 同时也具有较好的稳定性。

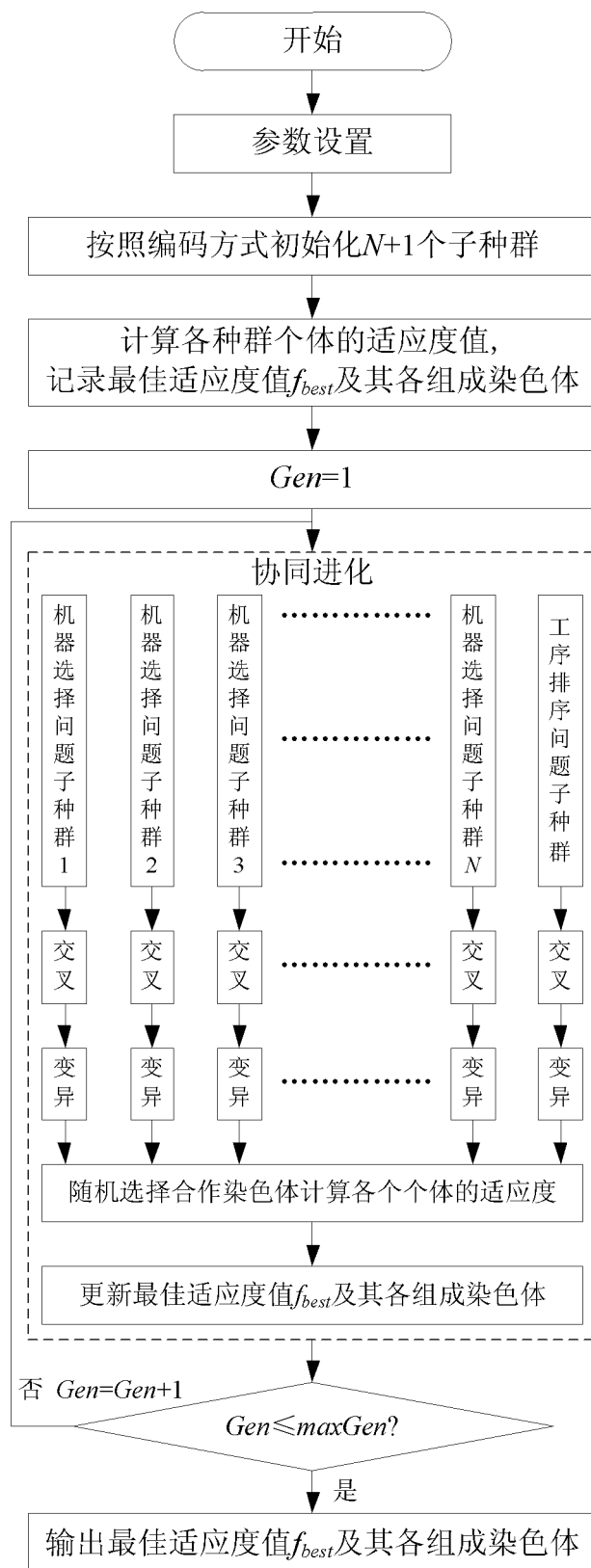


图 1

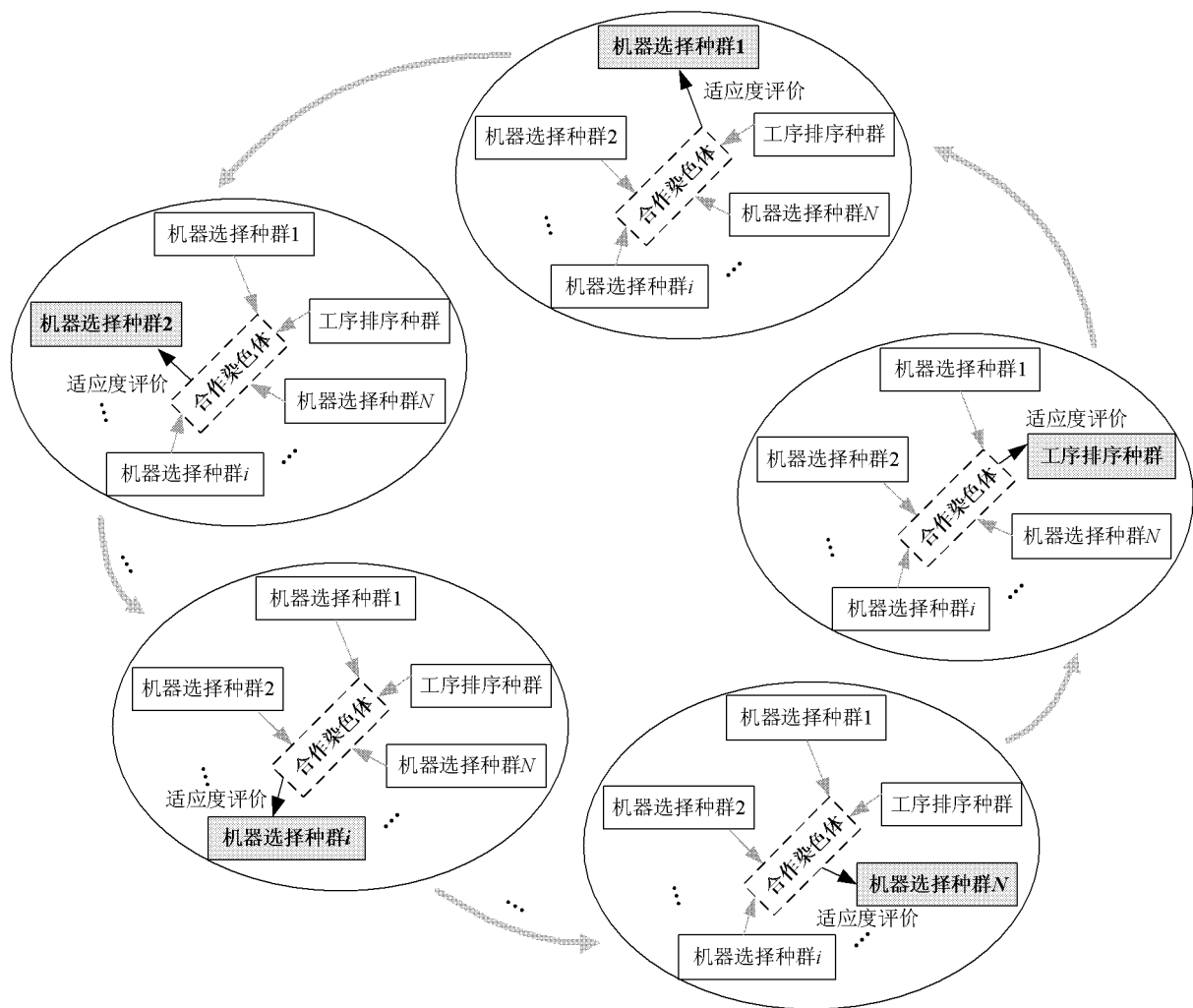


图 2