



(12) 发明专利申请

(10) 申请公布号 CN 115702403 A

(43) 申请公布日 2023. 02. 14

(21) 申请号 202180044956.X

(22) 申请日 2021.06.23

(30) 优先权数据

20181952.1 2020.06.24 EP

(85) PCT国际申请进入国家阶段日

2022.12.23

(86) PCT国际申请的申请数据

PCT/EP2021/067235 2021.06.23

(87) PCT国际申请的公布数据

W02021/260059 EN 2021.12.30

(71) 申请人 西门子股份公司

地址 德国慕尼黑

(72) 发明人 M·巴斯马齐 M·O·奥兹坎

N·耶斯卡亚

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

专利代理师 徐红燕 吕传奇

(51) Int.Cl.

G05B 19/418 (2006.01)

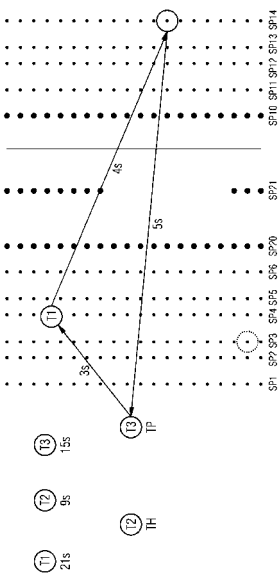
权利要求书2页 说明书9页 附图2页

(54) 发明名称

用于制造工件的最优工具配置

(57) 摘要

一种通过计算机控制的制造机器用预定义的序列的机床制造工件的方法,其中利用遗传算法确定最优工具配置。



1. 一种通过具有最优工具配置的计算机控制的制造机器用预定义的序列的机床 (T1, T2, T3) 制造工件的方法,

这些工具位于由工具配置设置的相应位置 (P1, P2, P3), 并且其被分配给相应的预先计算的操作时间,

并且该机器具有工具转移点 (TP) 和用于将选择的工具从相应位置移动到转移点的工具处理器,

并且执行以下步骤:

a) 分配工具的初始位置, 并且为所有工具计算

a1) 第一转移时间函数, 返回在相应位置和转移点之间的时间,

a2) 第二转移时间函数, 返回工具处理器移动而不移动工具的时间,

b) 生成可能的工具位置序列作为工具位置序列的随机集合,

c) 使用第一成本函数为可能的工具位置序列计算相应的第一成本因子, 该第一成本函数使用第一转移时间函数和第二转移时间函数,

d) 根据相应的第一成本因子对步骤c) 中获得的工具位置序列进行排序, 并且使用具有最低第一成本因子的工具位置序列的子集合作为工具位置序列的选择集合,

并且基于工具位置序列的选择集合探索新的工具位置序列, 并且将新的工具位置序列添加到工具位置序列的选择集合,

继续步骤c) 以计算成本函数并且执行所述排序和选择以及探索, 直到达到迭代次数的预定义参数,

将具有最低成本函数的工具位置序列的选择集合定义为最优工具配置,

e) 根据最优工具配置将工具从其初始位置转移到最优位置,

f) 由制造机器利用最优工具配置制造工件。

2. 根据前述权利要求所述的方法, 其中, 根据以下相关性定义第一成本函数:

$$C = \sum_{i=2}^N (o_i + \max(0, MT_i - o_i))$$

其中

$$MT_i = G(t_{i-1}, v(t_{i-1})) + R(v(t_{i-1}), v(t_{i+1})) + G(t_{i+1}, v(t_{i+1}))$$

C... 第一成本函数,

G... 第一转移时间函数,

R... 第二转移时间函数,

O ... 操作时间函数,

MT... 工具更换设备的移动时间。

3. 根据前述权利要求中任一项所述的方法, 其中, 使用第二成本函数, 进一步考虑重新分配工具所需的时间, 使用第一成本因子, 以及进一步关于包括重新分配时间在内的单个制造步骤对优化的影响的重新分配因子, 或者可替换地关于单个制造步骤的发生次数对整个制造的影响的计数因子, 为选择的集合计算相应的第二成本因子。

4. 根据权利要求3所述的方法, 其中第二成本函数是根据以下相关性定义的:

$$CP_1 = C + (\alpha \times RT)$$

其中

$$RT = \sum_{j=1}^X R(v(tr_j), vt(tr_j))$$

其中

CP_1 ...第二成本函数类型1,

R ...重新分配时间,

RT ...总重新分配时间。

5.根据权利要求3所述的方法,其中第二成本函数是根据以下相关性定义的:

$$CP_2 = (C \times \rho) + RT$$

其中

$$RT = \sum_{j=1}^X R(v(tr_j), vt(tr_j))$$

其中

CP_2 ...第二成本函数类型2,

R ...重新分配时间,

RT ...总重新分配时间。

6.根据前述权利要求中任一项所述的方法,其中,新工具位置序列的探索包括应用遗传交叉函数,其中,来自所选择的集合的两个工具位置序列被组合成新工具位置序列。

7.根据前述权利要求中任一项所述的方法,其中,新工具位置序列的探索包括变异函数的应用,其中,来自所选择的集合的工具位置序列内的工具位置被交换,优选随机交换。

8.具有机床(T1,T2,T3)的计算机控制的制造机器,包括具有存储器的控制设备,用于确定用于制造工件的工具的最优序列,

其中工具位于相应的位置(P1,P2, P3)并且其被分配给相应的预先计算的操作时间,

并且该机器具有工具转移点(TP)和用于将选择的工具从相应位置移动到转移点的工具处理器,

其特征在于,控制设备被配置为执行根据前述权利要求之一的方法。

用于制造工件的最优工具配置

技术领域

[0001] 本发明涉及一种通过具有最优工具配置的计算机控制的制造机器用预定义的序列的机床制造工件的方法。

背景技术

[0002] 计算机数字控制 (CNC) 机器的所谓“工具索引问题”(TIP, tool indexing problem) 出现在制造业中, 这是由于高度复杂的切割操作和对工具在刀库槽 (magazine slot) 中的最优放置的需要。

[0003] 机床的序列限定了工件的制造过程。

[0004] TIP 的解决方案产生了一个新的问题, 即工具如何能够从其初始位置 (即定位) 转移到最优位置, 使得所有转移时间的总和最小化, 这可以称为“工具替换问题” (TRP)。

[0005] 这是一项非常复杂和耗时的任务, 尤其是对于包含许多工具的大型刀库。为了在整个制造过程中有一个完整的优化, 这两个问题应该一起考虑。事实上, 在工具更换花费的时间比通过工具位置索引优化获得的增益多的情况下, 不能被认为是改进。

发明内容

[0006] 本发明的目的是以这样一种方式解决如前所述的“工具重新放置问题”, 即它可以在边缘设备, 即具有有限计算能力的计算设备上执行。

[0007] 该问题通过如前所述的方法来解决, 其中工具位于由工具配置设置的相应位置, 并且其被分配给相应的预先计算的操作时间, 并且该机器具有工具转移点和用于将选择的工具从相应位置移动到转移点的工具处理器, 并且执行以下步骤:

a) 分配工具的初始位置, 以及为所有工具计算

a1) 第一转移时间函数, 返回在相应位置和转移点之间的时间,

a2) 第二转移时间函数, 返回工具处理器移动而不移动工具的时间,

b) 生成可能的工具位置序列作为工具位置序列的随机集合,

c) 使用第一成本函数为可能的工具位置序列计算相应的第一成本因子, 该第一成本函数使用第一转移时间函数和第二转移时间函数,

d) 根据相应的第一成本因子对步骤 c) 中获得的工具位置序列进行排序, 并且使用具有最低第一成本因子的工具位置序列的子集作为工具位置序列的选择集合,

并且基于工具位置序列的选择集合探索新的工具位置序列, 并且将新的工具位置序列添加到工具位置序列的选择集合,

继续步骤 c) 以计算成本函数并且执行所述排序和选择以及探索, 直到达到迭代次数的预定义参数,

将具有最低成本函数的工具位置序列的选择集合定义为最优工具配置,

e) 根据最优工具配置将工具从其初始位置转移到最优位置,

f) 由制造机器利用最优工具配置制造工件。

[0008] 因此,实现了最优工具位置序列考虑了初始工具布置以及工具处理器的工具移动,以及因此,应用于工件制造的整体最优工具位置序列或工具位置序列。

[0009] 在本发明的上下文中,所述机床涉及在单个计算机控制的制造机器上使用的工具。除了工具索引问题之外,将工具从其初始位置重新放置到最优位置是另一个问题,并且花费大量时间。联合解决这两个问题将比单独解决节省更多的时间。具有最优工具位置将通过减少主轴中的等待时间来减少NC程序的运行时间-实际上是工件的制造时间。此外,这些工具将以最优方式转移到它们的最优位置。

[0010] 很明显,当所用工具的数量增加时,最优序列在本发明的上下文中变得更加相关。特别地,如果所使用的工具的数量至少为三个或五个,优选至少十个或二十个,更优选至少五十个或一百个或更多,则本发明可以发挥其优点。

[0011] 本发明在非常短的时间内解决了这些问题,并且与强力方法相比需要更少的存储器,因此对于包含数百个工具的巨大刀库更方便。

[0012] 与文献中没有联合地考虑TRP和TIP的强力方法或研究相比,本发明可以节省时间并提高生产率。

[0013] 此外,没有必要支付给第三方库。此外,该解决方案不需要OSS清算。

[0014] 一眼看去,本发明适用于具有有限计算能力的软件平台,如边缘设备。

[0015] 本发明从任意工具位置序列的集合开始,并开发改进的新工具位置序列,以便找到具有相对可选的成本因子的工具位置序列。

[0016] 结果不是绝对最优的解决方案,但是允许计算非常好的解决方案,但是成本显著降低。

[0017] 探索新的工具位置序列是有利的,因为可以基于工具位置序列执行工具位置序列的新评估,该工具位置序列已经被识别为具有低适应度函数值的良好工具位置序列。因此,可以预见,两个选择的工具序列以这样一种方式组合,即利用来自选择的序列的部件获得一个新的序列。这种组合也提供具有低适应度函数值的好的新序列的可能性很高。

[0018] 在本发明的进一步发展中,可以预见,第一成本函数根据以下相关性来定义:

$$C = \sum_{i=2}^N (o_i + \max(0, MT_i - o_i))$$

其中

$$MT_i = G(t_{i-1}, v(t_{i-1})) + R(v(t_{i-1}), v(t_{i+1})) + G(t_{i+1}, v(t_{i+1}))$$

C...第一成本函数,

G...第一转移时间函数,

R...第二转移时间函数,

O ...操作时间函数,

MT...工具更换设备的移动时间。

[0019] 第一成本函数允许适应度函数的简单计算,该适应度函数可用于评估专用工具位置序列的性能。然而,其他适应度函数适用于评估访问工具的成本。

[0020] 在本发明的进一步发展中,可以预见的是,使用第二成本函数,进一步考虑重新分

配工具所需的时间,使用第一成本因子,以及进一步关于包括重新分配时间在内的单个制造步骤对优化的影响的重新分配因子,或者可替换地关于单个制造步骤的发生次数对整个制造的影响的计数因子,为选择的集合计算相应的第二成本因子。

[0021] 此方面允许进一步改进与特定实施例相关的所应用的成本函数的计算,例如并入与改变起来昂贵的特定工具相关的重新定位因子,或与生产的工件数量相关的计数因子。

[0022] 在本发明的进一步发展中,预见到第二成本函数根据以下相关性来定义:

$$CP_1 = C + (\alpha \times RT)$$

其中

$$RT = \sum_{j=1}^X R(v(tr_j), vt(tr_j))$$

其中 CP_1 ...第二成本函数类型1(重新定位因子),

α ...重新分配因子

R...重新分配时间,

RT...总重新分配时间。

[0023] 这一方面允许通过并入与改变起来昂贵的特定工具相关的重新定位因子来改进成本函数,以及因此其例如在优化过程中根本不应该被考虑,或者至少具有低优先级,即权重。

[0024] 在本发明的进一步发展中,预见到第二成本函数根据以下相关性来定义:

$$CP_2 = (C \times \rho) + RT$$

$$\text{其中 } RT = \sum_{j=1}^X R(v(tr_j), vt(tr_j))$$

其中

CP_2 ...第二成本函数类型2(计数因子),

ρ ...计数因子

R...重新分配时间

RT...总重新分配时间。

[0025] 这一方面允许通过并入与生产的工件数量相关的计数因子来改进成本函数,当频繁重复时,该计数因子对优化过程有很大影响。

[0026] 在本发明的进一步发展中,预见到新工具位置序列的探索包括遗传交叉函数的应用,其中来自所选择集合的两个工具位置序列被组合成新的工具位置序列。遗传交叉将在下面进一步描述。

[0027] 因此,通过将已经识别的好的工具位置序列与低成本因子值相结合,可以以容易的方式执行新工具位置序列的探索。

[0028] 在本发明的进一步发展中,预见到新工具位置序列的探索包括突变函数的应用,其中来自所选择集合的工具位置序列内的工具位置被交换,优选地随机交换。

[0029] 因此,新工具位置序列的探索可以通过突变以简单的方式执行,这将在下面进一步描述。

[0030] 交叉函数在本领域中是已知的,但是应用有序的交叉函数可以在本发明的上下文中导致对最优工具位置序列的候选的改进选择。

[0031] 在遗传算法(GA)和进化计算中,交叉(也称为重组)是一种遗传算子,用于结合两个父母的遗传信息来生成新的后代。这是从现有种群中随机产生新解决方案的一种方法,类似于生物学中有性繁殖过程中发生的交叉。

[0032] 在一些遗传算法中,并非所有可能的染色体都代表有效解决方案。在某些情况下,可以使用专门设计的交叉和突变算子来避免违反问题的约束。

[0033] 该问题通过具有机床的计算机控制的制造机器来解决,该机器包括具有存储器的控制设备,用于确定用于制造工件的工具的最优序列,其中工具位于相应的位置,并且其被分配给相应的预先计算的操作时间,并且该机器具有工具转移点和用于将选择的工具从相应的位置移动到转移点的工具处理器,其中控制设备被配置为执行根据本发明的方法。

附图说明

[0034] 下面参照附图中所示的实施例更详细地解释本发明。附图中示出:

图1是在加工期间在CNC机器上的工具处理草图的示例,

图2是根据本发明的方法的实施例。

具体实施方式

[0035] 在CNC机器中用于加工一个工件的总生产时间考虑两部分:

i) 当工具加工材料、线切割或钻孔时的加工时间,以及

ii) 当机械臂在刀库上移动以取回后续工具或工具被刀库中的夹具切换时的非加工时间。

[0036] 一个工件的加工是用特定序列的工具来完成的,并且任何工具都可以在这个序列中多次使用。

[0037] 在加工工件时,前面的工具应回到相应的位置。然后,机器人臂应该移动到后续工具的位置,并最终将其带到工具转移点,从该点,所选择的工具被其中操作工具的机器的工具架(例如主轴)拾取。

[0038] 因此,这是为下一次操作取回后续工具的三步骤过程。

[0039] 在此过程期间,如果当前工具操作在下一个工具被取回至转移点之前完成,则主轴会出现一段空闲时间。

[0040] 它被称为等待时间,在此时段期间不能进行任何加工。

[0041] 为了提高生产率,应该尽可能最小化等待时间。

[0042] 这个问题被称为TIP,并且解决方案需要一起考虑如前面提到的工具重新分配问题TRP。

[0043] 这些问题是一种组合优化问题,并且可以公式化为广义的“旅行推销员问题”(TSP),其中TSP中的城市对应于TIP中的工具,以及TSP中的距离对应于TIP中的工具的转移时间。

[0044] 虽然TSP的成本函数最小化了销售员走过的总距离,但是TIP最小化了总生产时间。

[0045] 主要区别在于,尽管城市位置是固定的以及在TSP中寻找最优解决方案路径,而工具位置序列是已知的,在TIP中寻找最优工具位置索引。

[0046] 尽管它们不同,但两个问题的主要思想是相同的,它们都通过改变城市路径或通过改变工具位置来寻求最优穿越时间的解决方案。

[0047] 在下面的章节中,对这些问题的约束及其公式进行了解释,并提出了一种遗传算法(GA)的实现方法,给出了详细的步骤。

[0048] 约束

使用CNC机器和刀库运行带来了一些约束,这些约束在其本质上是优化问题的条件,并且它们也应该在解决方案和公式中得到满足。在本情况中,定义了以下约束:

- 每个工具应位于不同的位置。在结果集合中不应有任何重复的位置。

[0049] ·每个工具都有一个“向左尺寸”和“向右尺寸”参数,这两个参数在半个位置处定义。通常,这些尺寸是一,这是最小值。对于无法放入一个工具位置的较大工具,这些值将大于1。较大的工具占据相邻位置的一半位置,并且没有工具可以位于重叠的位置上。

[0050] ·机器操作员可以启用或禁用刀库位置。在禁用位置无法定位任何工具。

[0051] ·有两种类型的工具可以存在于刀库中,即“重型”和“轻型”工具。正如它们的名字暗指一样,这种区别是基于工具的重量。工具重量影响工具在工具库上移动的时间。将工具从转移点返回到一个位置的时间和将工具从一个位置取出到转移点的时间根据工具重量而变化。

[0052] 问题公式化

该问题定义如下:

- 工具列表 $T = t_1, t_2, \dots, t_N$, 它们根据制造过程的发生而排序,

- 刀库位置列表 $P = p_1, p_2, \dots, p_M$, 其中 $M > N$,

- 位置函数 $v(t_i)$, 其返回工具 t_i 的刀库位置,

- 每个工具的操作时间 $O = o_1, o_2, \dots, o_N$, 它们是运行一次NC程序时预先计算的,

- 转移时间函数 $G(t_i, p_m)$, 它返回工具 t_i 从一个刀库位置 p_m 到转移点(机械臂)的转移时间,反之亦然。

[0053] ·转移时间函数 $R(p_n, p_m)$, 其返回空机械臂从一个刀库位置 p_m 到另一个刀库位置 p_n 的转移时间,

目的是为工具分配合适的位置,使得NC程序的总执行时间最小。

[0054] 这个问题可以公式化如下:

找出使成本函数最小化的工具的位置V:

$$C = \sum_{i=2}^N (o_i + \max(0, MT_i - o_i))$$

其中

$$MT_i = G(t_{i-1}, v(t_{i-1})) + R(v(t_{i-1}), v(t_{i+1})) + G(t_{i+1}, v(t_{i+1}))$$

表示机械臂的移动时间,其包括:

·完成其切割的前一个工具从转移点到其原来位置的转移时间。

[0055] ·将空机械臂从上一个工具的位置转移到下一个将被取回的的工具的位置的转移时间。

[0056] ·下一个工具从其位置到转移点的转移时间。

[0057] 并且部分 $\max(0, MT_i - o_i)$ 表示等待时间 t_i , 在取回下一个工具 t_{i+1} 比当前工具 t_i 的操作时间更长的情况下。

[0058] 工具位置序列由制造过程通过所述工具列表 $T = t_1, t_2, \dots, t_N$ 预定义。

[0059] 工具布置在由所述位置函数 $v(t_i)$ 表示的位置, 这是本发明的目的。

[0060] 图1示出了CNC机器上工具移动的示例。

[0061] 可以看到带有预定义位置的主轴刀库SP1-SP6、SP10-SP14和SP20-SP21。

[0062] 在这种情况下存在三种工具, 它们是:

i) 工具T1, 其是先前使用的工具, 并且它应该被返回到其先前的位置,

ii) 工具T2, 其在工具架TH中, 即主轴中, 并且当前正在切割工件, 以及

iii) 工具T3, 其是将被使用的下一个工具, 并且应该被取回到转移点TP。

[0063] 在该图中, 工具T1、T2、T3下面的数字表示以秒为单位的相应操作时间 O_{T1}, O_{T2}, O_{T3} 。

[0064] 对于这个示例,

·工具T2的操作时间 O_{T2} 是9s,

·前一工具的返回时间 $G(t_{T1}, v(t_{T1}))$, 所谓的第一转移时间函数, 是3s,

·从前一工具位置到后继工具位置的空移动时间 $R(v(t_{T1}), v(t_{T3}))$, 所谓的第二转移时间函数是4s, 并且

·后续工具到转移点的拾取时间 $G(t_{T3}, v(t_{T3}))$ 为5s。

[0065] 这个示例的等待时间是通过下式为3秒

$$\max\left(0, \left[G(t_{T1}, v(t_{T1})) + R(v(t_{T1}), v(t_{T3})) + G(t_{T3}, v(t_{T3}))\right] - o_{T2}\right)$$

TRP联合TIP

第二个问题是TRP, 并且它与TIP联合地被解决。

[0066] 在某些情况下,将工具从其初始位置转移到最优索引位置需要大量时间,特别是如果结果包含大量的工具实际位置或包含较长的工具行进距离。

[0067] 因此,在这种情况下,就生产率而言,不是具有需要许多转移时间的最优解决方案,而是具有较少工具转移时间的稍微次优的解决方案可能更好。

[0068] 这是整个优化的重要部分。

[0069] 该概念预见到将工具重新分配所需的时间添加到GA的成本函数中,并调整算法。

[0070] 其次,从盈利角度考虑TRP时,制造件数也很重要。

[0071] 以前,据说主要目标是通过扩展的GA解决方案减少主轴中的等待时间并提高生产率。

[0072] 最终的目标分配可能会导致以下情况:一些工具必须首先从其原始刀库位置运走,然后才能将任何其他工具放置在该位置。

[0073] 产生的问题是如何将初始工具分配转移到目标分配,使得所有转移时间的总和最小化。

[0074] 因此,除了最小化等待时间之外,该问题还从两个角度考虑了工具的重新分配时间:

I. 第一个是重新分配因子 α 对优化结果的影响。为了研究重新分配因子 α 的影响,将上面定义为C的成本函数扩展如下:

·要放置在不同于初始位置的目标位置的工具体列表 $Tr = tr_1, tr_2, \dots, tr_X$

·目标位置函数 $vt(tr_j)$,其返回工具 tr_j 的目标位置。

[0075] ·重新分配因子 α 在0和1之间的范围内,允许重构成本函数:

$$CP_1 = C + (\alpha \times RT)$$

其中

$$RT = \sum_{j=1}^X R(v(tr_j), vt(tr_j))$$

表示对于一个解决方案的总重新分配时间RT。

[0076] 换句话说,重新定位因子是工具从第一位置重新定位到第二位置是否昂贵的表示,例如时间消耗。因此,应该避免这样的工具重新定位,这通过在优化过程中重新定位因子的相应高权重来表示。重新定位因子是介于0和1之间的权重值。

[0077] II. 第二个是生产率视角。NC程序执行的估计次数作为GA解决方案的一个参数来获取,以及根据这个运行次数,提供了最优解决方案作为结果。

[0078] 为了考虑生产率上的重新分配时间,上面定义的成本函数被扩展,其中 ρ 作为NC程序执行的估计计数因子,如下所示:

$$CP_2 = (C \times \rho) + RT$$

换句话说,计数因子是一种工具是否经常在预定义的制造序列中使用的表示。因此,如果经常使用,专用制造步骤的权重很高,即优化过程中的权重很重要,并且相应的计

数因子值也很高。

[0079] 计数因子是介于0和1之间的权重值。

[0080] GA步骤

在这一节中,解释GA的实现细节。GA是一种元启发式算法,并且已被证明在优化问题中非常有效。

[0081] GA实现包括五个主要步骤:

1. 随机初始群体创建:GA的第一步是从随机生成的染色体开始。初始群体的数量由“初始群体”超参数定义。创建的解决方案由工具位置序列组成,其称为“基因”。解决方案的长度由用于切割的NC程序中使用的工具数量来定义。

[0082] 2. 对解决方案进行排序:排序是GA方法的第二步。这是基于“适应度”函数完成的,其细节将在下面提供。总切割时间较少的解决方案排序在其他解决方案之前。

[0083] 3. 选择:GA的第三步是为下一代和为变异选择解决方案。在这个步骤中,淘汰具有低适应度函数分数的解决方案。精英染色体/解决方案总是在没有任何修改或变异的情况下被带入到下一代。精英染色体的尺寸由另一个称为“精英尺寸”的超参数确定。这里使用的选择方法是“健康比例选择”,也称为“轮盘赌选择”。这种选择方法旨在为下一代选择潜在有用的解决方案。在精英被直接带入下一代后,其他解决方案通过给予与其适应度分数成比例的机会来选择,并在交叉和变异后带入下一代。因此,具有较低成本的解决方案(即,较好的解决方案)更有机会被这种选择方法选择。

[0084] 4. 在选择之后,下一步骤是应用交叉,其中用于这一点的不同的选项是适用的。有序交叉(OX)优于本领域已知的其他交叉技术,因为它选择了比亲代更均匀分布的基因。

[0085] 5. 变异:最后一步骤是对找到的解决方案进行变异。这里使用“变异率”超参数来决定是否对当前解决方案进行变异。

[0086] 6. 作为预定义的“世代计数”超参数,重复上述过程。

[0087] 在遗传算法和进化计算中,交叉也称为重组,是一种遗传算子,用于结合两个父母的遗传信息来产生新的后代。这是从现有群体中随机产生新解决方案的一种方式,类似于生物学中有性繁殖过程中发生的交叉。

[0088] 解决方案也可以通过克隆现有的解决方案来产生,这类似于无性繁殖。新生成的解决方案在被添加到群体中之前通常被变异。

[0089] 在本发明上下文中的变异表示在优化过程中使用的工具位置序列中工具位置的变化。

[0090] 图2示出了根据本发明的方法的实施例。

[0091] GA方法开始于在步骤A“initialPopulation()”中生成可能解决方案的随机化初始集合1。

[0092] 每个解决方案都用一个“基因”来表示。

[0093] 接下来,在步骤B“rankPossibleSolutions()”中,根据排序后的“适应度”函数对解决方案进行排序,即排序后的群体数据2。

[0094] 在这个阶段,在步骤C“selection()”中,使用专门的选择方法,低适应度的解决方案被淘汰,并且被选择的个体3被进一步处理。

[0095] 下一代解决方案是通过步骤D“treedPopulation()”中的过程开发的,其模拟

自然选择中的变异和交配,并产生具有孩子和精英4的新群体。

[0096] 根据在步骤E执行的下一个变异群体5重复该循环,直到达到预定义的终止标准。

[0097] 参考数字列表:

1-5 数据输出

A-E 方法步骤,函数调用

SP1-SP6、SP10-SP14、

SP20-SP21 主轴刀库,预定义位置

T1,T2,T2 工具

TH 工具架,主轴

TP 工具转移点

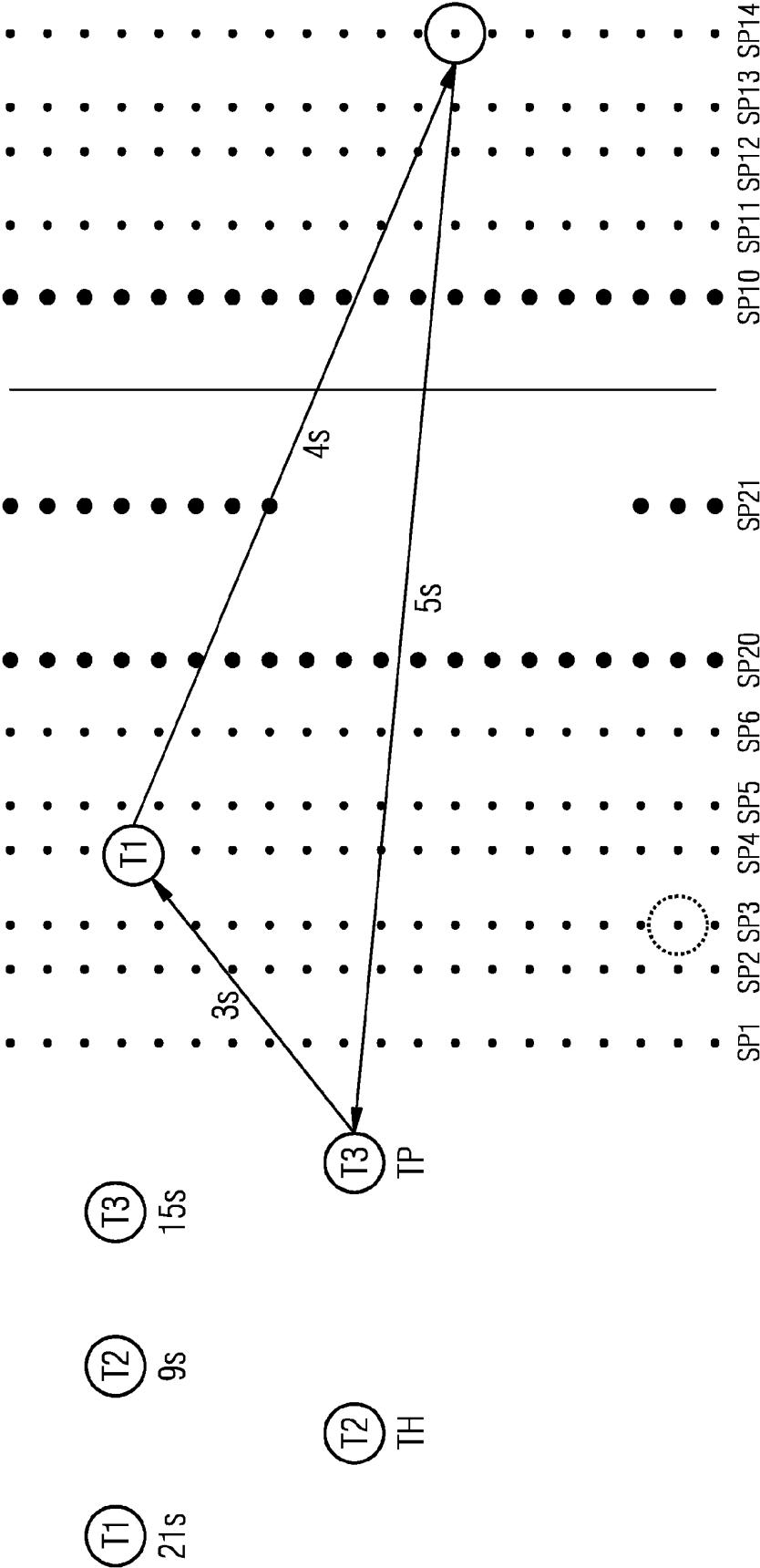


图 1

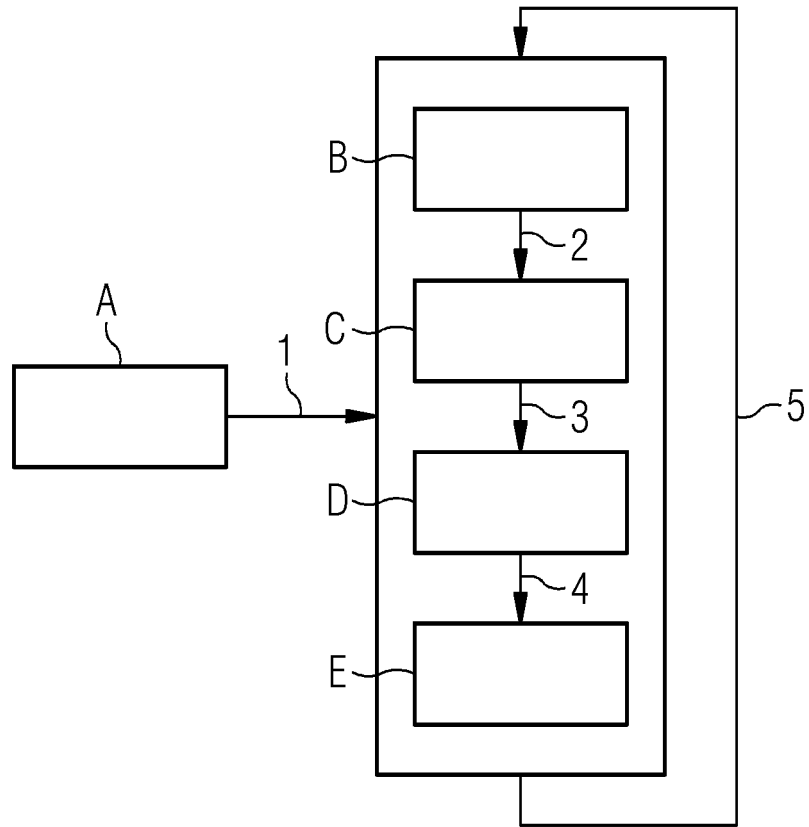


图 2