(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112817841 A
(43)申请公布日 2021.05.18

(21)申请号 202110082992.7

(22)申请日 2021.01.21

(71)申请人 西安交通大学
地址 710049 陕西省西安市咸宁西路28号

(72)发明人 杜小智 贺红梅 刘晋兰 董鸿磊 张金金 段宇蓉

(74)专利代理机构 西安通大专利代理有限责任公司 61200
代理人 李红霖

(51)Int.Cl.
G06F 11/36(2006.01)
G06N 3/00(2006.01)

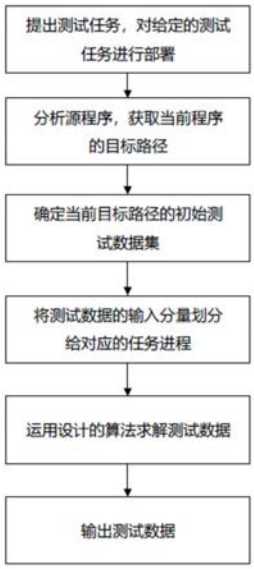
权利要求书3页 说明书9页 附图5页

(54)发明名称

基于路径覆盖的通信确定MPI并行程序测试数据生成方法

(57)摘要

本发明公开一种基于路径覆盖的通信确定MPI并行程序测试数据生成方法，包括如下步骤：S1，对待测试的MPI并行程序进行部署；S2，获取MPI并行程序的目标路径；S3，确定所述目标路径的初始测试数据集；S4，将每个初始测试数据的分量对应地划分给MPI并行程序的每一个子进程；S5，对所述子进程和所述MPI并行程序应用协同交叉人工蜂群搜索算法产生测试数据。本发明基于路径覆盖的通信确定MPI并行程序测试数据生成方法中，对子进程和所述MPI并行程序应用协同交叉人工蜂群搜索算法产生测试数据，协同交叉人工蜂群搜索算法将人工蜂群算法、单点交叉和协同机制很好的融合起来，从而解决了因并行程序自身特点而带来的测试数据生成问题。



CN 112817841 A

1.基于路径覆盖的通信确定MPI并行程序测试数据生成方法,其特征在于,包括如下步骤:

S1,对待测试的MPI并行程序进行部署;

S2,获取MPI并行程序的目标路径;

S3,确定所述目标路径的初始测试数据集;

S4,将每个初始测试数据的分量对应地划分给MPI并行程序的每一个子进程;

S5,对所述子进程和所述MPI并行程序应用协同交叉人工蜂群搜索算法产生测试数据。

2.根据权利要求1所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法,其特征在于,S5中,所述子进程与人工蜂群算法中的子种群一一对应,MPI并行程序与合作团体群对应;子进程和所述MPI并行程序通过协同进化机制进行协作。

3.根据权利要求2所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法,其特征在于,S5包括如下步骤:

S5.1,所有子进程并发产生测试数据,每个子进程均保存较优秀的测试数据并把它传递给MPI并行程序;

S5.2,MPI并行程序对接收到的子进程传递的较优秀的测试数据进行优劣比较,判断是否为期望的测试数据,若为期望的测试数据,则将期望的测试数据输出;若为非期望的测试数据,则将非期望的测试数据回传给子进程;

S5.3,重复S5.1-S5.2,直到达到预定的循环次数或者寻到完全覆盖目标路径的数据,结束循环。

4.根据权利要求2所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法,其特征在于,所述人工蜂群算法中,子进程的路径相似度$f\left(c(p_t^i),c(p_r^i)\right)$如下:

$$f\left(c(p_t^i),c(p_r^i)\right)=\frac{s\left(c(p_t^i),c(p_r^i)\right)}{\left|c(p_t^i)\right|}*f'\left(c(p_t^i),c(p_r^i)\right)$$

其中,$s(c(p_t^i),c(p_r^i))$表示在逐位对比的过程中,第一位开始连续相同编码的个数;$\frac{s\left(c(p_t^i),c(p_r^i)\right)}{\left|c(p_t^i)\right|}$表示两条路径连续相同的相同节点个数占目标路径节点个数的比例;$f'\left(c(p_t^i),c(p_r^i)\right)$为目标路径和穿越路径的相似度。

5.根据权利要求4所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法,其特征在于,目标路径和穿越路径的相似度$f'\left(c(p_t^i),c(p_r^i)\right)$如下:

$$f'\left(c(p_t^i),c(p_r^i)\right)=\sum_{k=1}^{\min}\frac{2*m_k}{(\min+1)*\min}*d_k\left(c(p_t^i),c(p_r^i)\right)$$

其中,min为$\left|c(p_t^i)\right|$和$\left|c(p_r^i)\right|$中的最小值函数,$\left|c(p_t^i)\right|$和$\left|c(p_r^i)\right|$为编码位数;p_t^i为子进程i的目标子路径,$c(p_t^i)$为p_t^i的编码;p_r^i为子进程i的穿越子路径,$c(p_r^i)$为p_r^i的编码;k表示$c(p_t^i)$和$c(p_r^i)$的第k位编码的异同性;m_k为当比较到第k位时相同位数量;$d_k\left(c(p_t^i),c(p_r^i)\right)$为

从左到右诸位对比$c(p_t^i)$和$c(p_r^i)$的编码结果。

6.根据权利要求2所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法，其特征在于，所述人工蜂群算法中，MPI并行程序的路径相似度$f(\vec{x})$如下：

$$f(\vec{x}) = \text{similarity(p)}$$

其中，similarity(p)为p_t和p_r的相似度；p_t和p_r分别为程序p的目标路径和穿越路径。

7.根据权利要求6所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法，其特征在于：

$$\text{similarity(p)} = \sum_{i=0}^{n-1}(w^{i'} * f)$$

其中，$w^{i'}$为子路径相似度权值w^i归一化之后的权值；$w^i = {}^i+1$，其中，1^i为子进程i直接涉及的程序输入个数；f为子进程的路径相似度。

8.根据权利要求6所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法，其特征在于，测试数据生成的数学模型如下：

$$\max\ f(\vec{x})\ (\vec{x} \in D)$$

其中，$\max\ f(\vec{x})$为MPI并行程序的路径相似度中的最大值，D为程序中参数的取值，$\vec{x}$为测试数据的输入。

9.根据权利要求3所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法，其特征在于，S5.1中，子进程产生测试数据的过程包括：

子种群优化与某一进程子进程路径有关的输入分量，在进化的过程中，以解码后的进化个体作为该进程的输入，执行该进程获得穿越路径，由个体穿越路径和目标子路径，计算得到个体的适应度值；当到达指定的进化周期时，子种群根据适应度，选择预设数量的优良个体组成代表个体集合发送给合作团体群，并等待接收合作团体群返回的优良个体，如果接收到的个体为空，表示已经找到了期望的测试数据，进化终止，否则继续进化；

S5.2中，MPI并行程序产生测试数据的过程包括：

合作团体群收到各子种群发送来的代表个体后，根据每个子种群进化的输入分量集合，对代表个体进行组合，形成本种群的初始进化个体；得到本种群的初始进化个体后，在给定周期的每一代进化中，解码进化个体作为整个并行程序的输入，通过执行整个程序得到穿越路径，由个体穿越路径和目标路径，得到个体的适应度；若有个体的适应度为1，则该个体为期望的测试数据，输出个体，终止进化，同时给子种群发送空信息，终止所有种群的进化；否则继续进化生成子代种群；当到达进化周期时，根据每个子种群进化的输入分量，将优势个体进行划分，并发送到相应的子种群中。

10.根据权利要求9所述的基于路径覆盖的通信确定MPI并行程序测试数据生成方法，其特征在于，所述对代表个体进行组合的过程包括：

首先，求输入变量与子种群进化的输入分量集合的差集；然后，从其他子种群的代表个体中，提取与差集分量相关的值，扩展被差输入分量集合对应子种群的代表个体，产生合作团体群的进化个体，用来进化产生覆盖目标路径的期望测试数据；求两子种群进化的输入分量集合的交集，并从其他子种群的代表个体中，提取与交集分量相关的值，替换已产生的

进化个体的相应值，产生新的进化个体；最后，若产生的进化个体不同于已存在的进化个体，且在进化个体中，至少包含一个完整的代表个体，那么将该进化个体加入到合作团体群中。

基于路径覆盖的通信确定MPI并行程序测试数据生成方法

技术领域

[0001] 本发明涉及软件工程领域，涉及测试数据生成方法，尤其涉及一种基于路径覆盖的通信确定MPI并行程序测试数据生成方法。

背景技术

[0002] 并行程序具有求解问题高效的特点，这要归功于其运行过程中多个进程并行执行。但同时该类程序运行过程中涉及到通信、同步，以及不确定性等，这常常会导致数据竞争、死锁，以及资源冲突等问题。而截止到目前，已有的工作大多偏向于检测上述由并行执行引起的死锁、资源冲突等问题。这些方法虽然确保了并行程序的成功运行，却并不能保证程序能够正确运行并得到期望的结果。在并行程序能够正常运行的前提下，即运行过程中不会发生死锁和资源冲突等问题，如何通过测试数据生成来进一步提高程序的可靠性，是需要研究的问题。

[0003] 现有工作关注于软件测试中测试数据的生成问题，研究启发式搜索算法来自动地产生测试数据。由于资源的竞争、网络延时等会影响并行程序的执行，因此现有的生成串行程序测试数据的方法不能完全应用于并行程序，同时，对应启发式搜索算法的适应度函数，没有多方面考虑路径相似度的权重问题，导致测试数据生成时间长迭代次数多效率低。

发明内容

[0004] 针对目前方法测试数据生成时间长、迭代次数多的问题，本发明提供一种基于路径覆盖的通信确定MPI并行程序测试数据生成方法，有效缩短测试数据的生成时间和迭代次数，提高测试数据的生成效率。

[0005] 本发明的目的是通过以下技术方案来实现：

[0006] 基于路径覆盖的通信确定MPI并行程序测试数据生成方法，包括如下步骤：

[0007] S1，对待测试的MPI并行程序进行部署；

[0008] S2，获取MPI并行程序的目标路径；

[0009] S3，确定所述目标路径的初始测试数据集；

[0010] S4，将每个初始测试数据的分量对应地划分给MPI并行程序的每一个子进程，MPI并行程序与子进程为总分关系，也可表述为总进程与子进程；

[0011] S5，对所述子进程和所述MPI并行程序应用协同交叉人工蜂群搜索算法产生测试数据。

[0012] 优选的，S5中，子进程与人工蜂群算法中的子种群一一对应，MPI并行程序与合作团体群对应；子进程和所述MPI并行程序通过协同进化机制进行协作。

[0013] 优选的，S5包括如下步骤：

[0014] S5.1，所有子进程并发产生测试数据，每个子进程均保存较优秀的测试数据并把它传递给MPI并行程序；

[0015] S5.2，MPI并行程序对接收到的子进程传递的较优秀的测试数据进行优劣比较，判

断是否为期望的测试数据，若为期望的测试数据，则将期望的测试数据输出；若为非期望的测试数据，则将非期望的测试数据回传给子进程；

[0016] S5.3，重复S5.1-S5.2，直到达到预定的循环次数或者寻到完全覆盖目标路径的数据，结束循环。

[0017] 优选的，所述人工蜂群算法中，子进程的路径相似度$f\left(c(p_t^i),c(p_r^i)\right)$如下：

[0018] $$f\left(c(p_t^i),c(p_r^i)\right)=\frac{s\left(c(p_t^i),c(p_r^i)\right)}{\left|c(p_t^i)\right|}*f'\left(c(p_t^i),c(p_r^i)\right)$$

[0019] 其中，$s(c(p_t^i),c(p_r^i))$表示在逐位对比的过程中，第一位开始连续相同编码的个数；$\frac{s\left(c(p_t^i),c(p_r^i)\right)}{\left|c(p_t^i)\right|}$表示两条路径连续相同的相同节点个数占目标路径节点个数的比例；$f'\left(c(p_t^i),c(p_r^i)\right)$为目标路径和穿越路径的相似度。

[0020] 优选的，目标路径和穿越路径的相似度$f'\left(c(p_t^i),c(p_r^i)\right)$如下：

[0021] $$f'\left(c(p_t^i),c(p_r^i)\right)=\sum_{k=1}^{\min}\frac{2*m_k}{(\min+1)*\min}*d_k\left(c(p_t^i),c(p_r^i)\right)$$

[0022] 其中，min为$\left|c(p_t^i)\right|$和$\left|c(p_r^i)\right|$中的最小值函数即$\min\left(\left|c(p_t^i)\right|,\left|c(p_r^i)\right|\right)$，$\left|c(p_t^i)\right|$和$\left|c(p_r^i)\right|$为编码位数；$p_t^i$为子进程i的目标子路径，$c(p_t^i)$为$p_t^i$的编码；$p_r^i$为子进程i的穿越子路径，$c(p_r^i)$为$p_r^i$的编码；k表示$c(p_t^i)$和$c(p_r^i)$的第k位编码的异同性；$m_k$为当比较到第k位时相同位数量；$d_k\left(c(p_t^i),c(p_r^i)\right)$为从左到右诸位对比$c(p_t^i)$和$c(p_r^i)$的编码结果。

[0023] 优选的，所述人工蜂群算法中，MPI并行程序的路径相似度$f(\vec{x})$如下：

[0024] $$f(\vec{x})=\text{similarity(p)}$$

[0025] 其中，similarity(p)为p_t和p_r的相似度；p_t和p_r分别为程序p的目标路径和穿越路径。

[0026] 优选的：

[0027] $$\text{similarity(p)}=\sum_{i=0}^{n-1}\left(w^{i'}*f\right)$$

[0028] 其中，$w^{i'}$为子路径相似度权值w^i归一化之后的权值；$w^i=l^i+1$，其中，l^i为子进程i直接涉及的程序输入个数；f为子进程的路径相似度。

[0029] 优选的，测试数据生成的数学模型如下：

[0030] $$\max\ f(\vec{x})\ \ (\vec{x}\in D)$$

[0031] 其中，$\max\ f(\vec{x})$为MPI并行程序的路径相似度中的最大值，D为程序中参数的取值，$\vec{x}$为测试数据的输入。

[0032] 优选的，S5.1中，子进程产生测试数据的过程包括：

[0033] 子种群优化与某一进程子进程路径有关的输入分量，在进化的过程中，以解码后的进化个体作为该进程的输入，执行该进程获得穿越路径，由个体穿越路径和目标子路径，计算得到个体的适应度值；当到达指定的进化周期时，子种群根据适应度，选择预设数量的优良个体组成代表个体集合发送给合作团体群，并等待接收合作团体群返回的优良个体，如果接收到的个体为空，表示已经找到了期望的测试数据，进化终止，否则继续进化；

[0034] S5.2中，MPI并行程序产生测试数据的过程包括：

[0035] 合作团体群收到各子种群发送来的代表个体后，根据每个子种群进化的输入分量集合，对代表个体进行组合，形成本种群的初始进化个体；得到本种群的初始进化个体后，在给定周期的每一代进化中，解码进化个体作为整个并行程序的输入，通过执行整个程序得到穿越路径，由个体穿越路径和目标路径，得到个体的适应度；若有个体的适应度为1，则该个体为期望的测试数据，输出个体，终止进化，同时给子种群发送空信息，终止所有种群的进化；否则继续进化生成子代种群；当到达进化周期时，根据每个子种群进化的输入分量，将优势个体进行划分，并发送到相应的子种群中。

[0036] 优选的，所述对代表个体进行组合的过程包括：

[0037] 首先，求输入变量与子种群进化的输入分量集合的差集；然后，从其他子种群的代表个体中，提取与差集分量相关的值，扩展被差输入分量集合对应子种群的代表个体，产生合作团体群的进化个体，用来进化产生覆盖目标路径的期望测试数据；求两子种群进化的输入分量集合的交集，并从其他子种群的代表个体中，提取与交集分量相关的值，替换已产生的进化个体的相应值，产生新的进化个体；最后，若产生的进化个体不同于已存在的进化个体，且在进化个体中，至少包含一个完整的代表个体，那么将该进化个体加入到合作团体群中。

[0038] 本发明具有以下有益的技术效果：

[0039] 本发明基于路径覆盖的通信确定MPI并行程序测试数据生成方法中，对子进程和所述MPI并行程序应用协同交叉人工蜂群搜索算法产生测试数据，协同交叉人工蜂群搜索算法将人工蜂群算法、单点交叉和协同机制很好的融合起来，从而解决了因并行程序自身特点而带来的测试数据生成问题。本发明在通信确定的MPI并行程序的测试数据生成中，数据迭代次数和测试数据生成时间明显减少，数据效果更好，具有非常优秀的性能。

附图说明

[0040] 图1为本发明实施例提供的基于路径覆盖的通信确定MPI并行程序测试数据生成方法粗粒度级流程图；

[0041] 图2为本发明实施例提供的基于路径覆盖的通信确定MPI并行程序测试数据生成方法细粒度级流程图，对图1的每一个步骤进行了细粒度级研究；

[0042] 图3为本发明实施例提供的MPI通信确定程序消息接收语句；

[0043] 图4为本发明实施例提供的人工蜂群算法步骤流程图；

[0044] 图5为本发明实施例提供的单点交叉案例示意图；

[0045] 图6为发明实施例提供的子任务与总任务协同迭代生成测试数据示例图。

具体实施方式

[0046] 下面结合附图和实施例对本发明做进一步的详细说明,所述是对本发明的解释而不是限定。

[0047] 参照图1和图2,本发明基于路径覆盖的通信确定MPI并行程序测试数据生成方法包括如下步骤:

[0048] S1:针对待测试任务即MPI并行程序进行部署;

[0049] S2:分析被测程序(即MPI并行程序),获取该程序的目标路径;

[0050] S3:明确目标路径后,应用边界值和等价类相结合的方法确定当前目标路径的初始测试数据集;

[0051] S4:将每个初始测试数据的分量对应地划分给每一个子进程;

[0052] S5:对子任务(即子进程)和总任务(即总进程)应用协同交叉人工蜂群搜索算法产生测试数据。

[0053] 其中,S2中获取被测程序的目标路径的过程具体如下:

[0054] 首先绘制被测程序的控制流图,多条流线的交汇点都是用圆圈标识,该圆圈被称作节点。在节点与节点间的连线称为边,它是具有方向的直线或者弧线。其次计算圈复杂度,它指出了程序逻辑结构的复杂度,明确了独立路径的个数。圈复杂度计算公式为V(G)=e-n+2p,e是强连通图的边数,n是节点数,p是连通区域数。最后从被测程序中选择目标路径时,它们须在被测程序的控制流图中均匀分布且覆盖程序中较多语句。

[0055] S3中获取初始测试数据集的过程具体如下:

[0056] 等价类划分就是对被测程序的输入变量集合,按不同规则进行划分,主要划分成有效等价类(符合程序输入规则)和无效等价类(不符合程序输入规则)。边界值分析作为等价类划分的一个补充方法,将等价类划分的边界作为用例设计,在有效和无效等价类的边界取值中,会存在值重复的情况,可按有效无效进行取舍。

[0057] S4中将每个初始测试数据的分量对应地划分给每一个子进程的具体过程如下:

[0058] 假设被测MPI并行程序的测试数据为(x_1,x_2,x_3,...,x_k),记为解空间$\vec{x} = x_1, x_2, x_3, \ldots, x_k$,每一个$x_i$对应被测程序中的一个变量,$x_k$是最后一个变量,通过k的值可以确定被测程序中变量的个数;假设该MPI并行程序的初始进程集划分为:

[0059] $process^0(x_1,x_2)$,$process^1(x_2,x_3)$,.....,$process^m(x_{k-1},x_6)$,其中每一个process称为一个子进程。该步骤采用的划分是把解空间的输入变量与子进程相对应,即每个子进程所需的输入变量都划分给对应的进程,即x_1,x_2→$process^0$,x_2,x_3→$process^1$,...,x_{k-1},x6→$process^m$。当生成与解空间对应的测试数据后,也将测试数据对应的输入变量划分给对应的子进程,这样使划分不再具有盲目性。

[0060] S5中应用协同交叉人工蜂群搜索算法产生测试数据的过程具体如下:

[0061] 协同机制的主要原理是拆分,它是将一个大的问题分解成多个小的子问题,每个子问题单独进行求解,子问题并发执行并且它们之间互不干涉。交叉机制选择单点交叉,单点交叉是指依据概率选取两个个体后,从个体中选择一个点(两个个体选择同样的点),该点称为交叉点,对交叉点后面或前面的位串进行交换则完成了单点交叉。人工蜂群算法内有三样不同的蜜蜂,雇佣蜂、观察蜂和搜索蜂,雇佣蜂和观察蜂在蜂巢附近依据积累的经验来选择蜜源并调整蜜源所在的方向,搜索蜂随机选择一个位置采摘花蜜。若蜜蜂从当前位

置采摘的花蜜多，他们会舍弃花蜜少的蜜源，把花蜜多的位置信息保存在记忆中。为解决并行程序特点所带来的问题，将上述方法进行结合，设计得到协同交叉人工蜂群搜索算法，应用该方法来生成通信确定的并行程序的测试数据。

[0062] 在生成测试数据的过程中，多个子任务并发产生测试数据，然后保存较优秀的测试数据并把它传递给总任务，总任务对当前的数据进行优劣比较，再将满足需要的数据回传给子任务。如此交替下去直到达到预定的循环次数或者寻到完全覆盖目标路径的数据。子进程与总进程相互传递信息、相互协作，共同完成寻找测试数据的过程。

[0063] 实施例

[0064] 参阅图1和图2，本实施例基于路径覆盖的通信确定MPI并行程序测试数据生成方法，包括以下步骤：

[0065] 步骤1、提出测试任务，对给定的测试任务进行部署。

[0066] 该步骤的具体内容阐述如下：

[0067] 该步骤中的测试任务指的就是通信确定的MPI并行程序，例如将求解3个数x、y、z的最大公约数的任务分成4个子任务，将这些子任务分别部署在4台计算机上，就完成了对给定测试任务的部署。编写一个MPI程序，需要六个基本的MPI函数，即MPI_Recv(void*buf，int count，MPI_Datatypedatatype，intsource，inttag，MPI_Comm_WORLD，&status)是消息接收语句。其中，source是发送信息的进程编号，tag对传送的信息进行编号。当source与tag都有具体的值时，说明当前的MPI程序采用确定的通讯方式。如图3所示，第一条消息接收语句接收到进程0编号1的消息，第二条接收语句接收到进程0编号2的消息。

[0068] 步骤2、分析源程序(即MPI并行程序)，获取当前程序的目标路径。

[0069] 该步骤的具体内容阐述如下：

[0070] S21：分析程序的数据流向，绘制程序控制流图；

[0071] S22：根据控制流图，计算圈复杂度，得到独立路径的个数；

[0072] 圈复杂度指出了程序逻辑结构的复杂度，明确了独立路径的个数。圈复杂度计算公式为V(G)＝e-n+2p，e是强连通图的边数，n是节点数，p是连通区域数。

[0073] S23：结合被测程序和圈复杂度，得到被测程序的独立路径；

[0074] 有了线性独立路径数量，即圈复杂度之后，进一步需要确定被测程序全部独立路径。主要步骤为：(1)选择一条独立路径，选择包含尽可能多的判定节点的路径；(2)回溯该路径，依次“反转”每个判定节点，即判定节点取其他值；(3)重复步骤(2)，直到所有的判定节点都取了不同的值为止。由此，即可得到被测程序的全部独立路径。

[0075] S24：在独立路径中，选择在被测程序的控制流图中均匀分布且覆盖程序中较多语句的路径作为目标路径。

[0076] 一般来说，一个程序包含多条独立路径，然而测试程序的全部路径是不现实的，也是没有必要的。所以，在实验过程中，仅选择被测程序的一部分路径作为目标路径，但是目标路径的选取对测试数据的生成代价有很大的影响，所以，为了减小这个威胁，在所有的目标路径中，选取在被测程序的控制流图中均匀分布且覆盖程序中较多语句的路径作为目标路径。

[0077] 步骤3、应用边界值分析和等价类划分方法分析被测程序，确定当前目标路径的初始测试数据集。

[0078] 步骤4、根据MPI并行程序的特性将每个测试数据的分量对应地划分给每一个子任务。

[0079] 步骤5、采用协同交叉人工蜂群搜索算法生成测试数据。

[0080] 该步骤的具体内容阐述如下：

[0081] S51：人工蜂群算法；

[0082] 人工蜂群算法流程图如图4所示，该算法中的每一只蜜蜂为一个子种群，设子种群为$\{bee^0, bee^1, \ldots, bee^n\}$，

[0083] 初始的子任务集为$process^0(x_1, x_2)$，$process^1(x_2, x_3)$，...，$process^m(x_{k-1}, x_6)$。考虑将人工蜂群算法中的子种群个数划分为n，且与子任务数m相同，并且子任务与子种群一一对应。则有$bee^i \Leftrightarrow process^j$，(i∈n，j∈m，n＝m)，并且i与j的取值可以相同也可以不相同。

[0084] 人工蜂群算法含有雇佣蜂、观察蜂、搜索蜂三种蜜蜂，下面对每一类蜜蜂的功能进行详细地阐述。

[0085] (1)雇佣蜂：雇佣蜂在当前的数据不能覆盖目标路径前提下，探索当前数据周围的数据，在当前数据的基础上加上一个随机的值来生成新的数据，数据的好坏根据适应值函数的值进行判断。若找到适应值恒为1的数据，则存储该数据退出寻优阶段。

[0086] (2)观察蜂：观察蜂依据概率p_j^i来选择数据，然后在探索所选择数据周围的数据，生成新数据的方法是在当前数据上加上一个随机数值，然后计算新数据的适应值。当某一数据的适应值为1时，说明找到所需的测试数据，存储数据并结束寻优。

[0087] (3)搜索蜂：搜索蜂把进行多次迭代后仍没有改善的两个数据进行交叉操作，运用单点交叉方式产生新的数据。具体的交叉方式如图5所示，假设为基因A序列和基因B序列设置交叉点和交叉结束点，单点交叉即对这两点之间的片段进行交换，基因A序列交叉前后的数据分别是“651471036”和“651823545”。交叉机制的引入可以保证再次探索的数据是以较优的数据为前提进行的，使探索具有导向性。同时，新的数据是一种异于所存在数据的，充分地保证了数据的多样性。然后计算新的数据的适应值，若适应值为1，则找到所需的数据，存储该数据并退出寻找数据的过程；若适应值不为1，则保存适应值高的数据，进入下一次循环继续寻找覆盖目标路径的数据。

[0088] S52：协同进化机制；

[0089] 在本发明中协同进化思想主要体现在子任务与总任务的协作上，如图6所示，子任务将最优数据传给总任务，总任务将较优数据传给子任务，相互迭代共同协作，产生最终数据。协同进化机制中的决策机构在本算法中的体现为当前测试数据对目标路径的覆盖率，即决策机构对应着传统人工蜂群算法中的适应度函数，适应度函数是通过真正走过的节点数占目标路径节点数的百分比来定义的，计算公式会依据实际问题进行设计。

[0090] 子任务先利用引用交叉机制的人工蜂群算法求解测试数据，当找到覆盖子路径的数据或达到最大迭代周期时，将数据传递给总任务，总任务对各个数据分量进行组合形成总任务的初始数据，总任务再应用引用交叉机制的人工蜂群算法寻找较优的测试数据，并回传给子任务，子任务以该数据为前提继续寻优。子任务与总任务交替执行，协作产生MPI并行程序的测试数据。

[0091] S53:适应度函数设计;

[0092] 适应度函数是通过真正走过的节点数占目标路径节点数的百分比来定义的,计算公式会依据实际问题进行设计。假设,对于有n个进程的并行程序,给定目标路径$p_t = p_t^0 \cup p_t^1 \cup \ldots \cup p_t^{n-1}$,当以$\vec{x}$为输入数据运行被测程序时,它穿越的路径为$p_r(\vec{x}) = p_r^0 \cup p_r^1 \cup \ldots \cup p_r^{n-1}$,因此只要先计算出子任务的路径相似度,进一步即可求出总任务路径相似度。

[0093] (1)计算子任务的路径相似度

[0094] 路径相似度计算的核心是需要计算两条路径连续相同节点所占的比例,而在比较两条路径节点的方法中,哈夫曼编码作为一种最优前缀编码,可以将数据传输的数量减少到最小。所以对被测程序进行哈夫曼编码,而且实际上,当被测程序表示成二叉树时,通常不必考虑权重问题。在本发明中,分支语句的假分支用0表示,其真分支用1表示,因此,可以从根节点遍历到叶子结点的路径编码,并且该路径编码被称为前缀编码。

[0095] 基于上述思想,本发明给出的方法如下。记并行程序有n个进程,目标路径记为p_t,对于进程i记目标子路径为p_t^i,则其编码为$c(p_t^i)$,编码位数为$|c(p_t^i)|$。以$\vec{x}$为测试数据的输入,穿越的路径记为pr,对于进程i记穿越子路径为p_r^i,则其编码为$c(p_r^i)$,编码位数为$|c(p_r^i)|$。从左到右诸位对比$c(p_t^i)$和$c(p_r^i)$的编码,结果记为:

[0096] $d_k(c(p_t^i), c(p_r^i))$ (1)

[0097] 其中k表示$c(p_t^i)$和$c(p_r^i)$的第k位编码的异同性,若相同,则$d_k(c(p_t^i), c(p_r^i)) = 1$,否则$d_k(c(p_t^i), c(p_r^i)) = 0$。这样比较次数最多为$\min(|c(p_t^i)|, |c(p_r^i)|)$次。

[0098] 对于穿越路径的每一个编码位,考虑编码位的重要性,当比较的相同位数越多,个体就越接近目标路径的测试数据。为了便于记录相同位的个数,在进行编码对比时,用计数器m来进行标记,即$m < min(|c(p_t^i)|, |c(p_r^i)|)$。为更清楚地区分不同编码位相应的相似度,针对编码位不同,设置了相应的权重,即两条路径在比较第k位编码时,总共的相同节点个数占目标路径节点个数的比例,但权重之和为1,故设置权重公式为$\frac{2*m_k}{(\min+1)*\min}$,其中,min为$|c(p_t^i)|$和$|c(p_r^i)|$中的最小值函数即$\min(|c(p_t^i)|, |c(p_r^i)|)$。当比较到第k位时相同位数量记为mk,目标路径和穿越路径的相似度记为$f'(c(p_t^i), c(p_r^i))$,可表示为:

[0099] $f'(c(p_t^i), c(p_r^i)) = \sum_{k=1}^{\min} \frac{2*m_k}{(\min+1)*\min} * d_k(c(p_t^i), c(p_r^i))$ (2)

[0100] 对比$c(p_t^i)$和$c(p_r^i)$,相同编码位数越多,$f'(c(p_t^i), c(p_r^i))$就越大。

[0101] 对于进程的穿越路径,考虑路径本身的重要性,从前到后连续相同的编码位,位数越多个体越接近于目标路径的测试数据。因此对公式进行修改,如下所示:

[0102] $f(c(p_t^i), c(p_r^i)) = \frac{s(c(p_t^i), c(p_r^i))}{|c(p_t^i)|} * f'(c(p_t^i), c(p_r^i))$ (3)

[0103] 调整后的相似度为$f(c(p_t^i), c(p_r^i))$,其中$s(c(p_t^i), c(p_r^i))$表示在逐位对比的过程中,

第一位开始连续相同编码的个数，$\frac{s\left(c(p_t^i),c(p_r^i)\right)}{|c(p_t^i)|}$表示两条路径连续相同的相同节点个数占目标路径节点个数的比例。与目标路径连续相同的编码位数越多，说明该路径与目标路径的相似程度越高。

[0104] 通过以上的分析，得到在进程i中目标路径和穿越路径的相似度计算公式。

[0105] (2)计算总任务的路径相似度

[0106] 对于一个消息传递并行程序而言，应该考虑该程序的每一进程直接涉及的程序输入。一般而言，不同进程直接涉及的程序输入也是不同的；某进程直接涉及的程序输入越多，那么，该进程子路径的相似度对整个路径相似度的贡献越大。为了体现子路径相似度的贡献，在计算路径相似度时，根据某进程直接涉及程序输入的个数，设定该进程子路径相似度的权值。

[0107] 基于上述思想，本发明给出的方法如下。记进程i为p^i，子路径相似度权值为w^i，该进程直接涉及的程序输入个数为1^i，因为权值不能小于或等于0，那么，w^i的值为：

[0108] $w^i=1^i+1$ (4)

[0109] 这样一来，就可以得到$w^0,w^1,\ldots,w^{n-1}$的值。对上述权值归一化操作，并记w^i归一化之后的权值为$w^{i'}$可以表示为：

[0110] $$w^{i'}=\frac{l^i+1}{\sum_{j=0}^{n-1}l^i+1} \quad (5)$$

[0111] 对应并行程序p的路径p_t和p_r，其进程p^i的子路径分别为p_t^i和p_r^i，采用公式(3)得到的子路径相似度为$f\left(c(p_t^i),c(p_r^i)\right)$简记为f，那么再结合本部分的方法，$p_t$和$p_r$的相似度为：

[0112] $$\text{similarity(p)}=\sum_{i=0}^{n-1}\left(w^{i'}*f\right) \quad (6)$$

[0113] 其中n为进程的个数，MPI并行程序的相似度只与输入数据有关，因此可以把并行程序的相似度记为：

[0114] $$f(\vec{x})=\text{similarity(p)} \quad (7)$$

[0115] 当相似度的定义明确后，了解到我们所需要的是能够覆盖目标路径的数据，基于此，给出测试数据生成的数学模型：

[0116] $$\max\ f(\vec{x})\ (\vec{x}\in D) \quad (8)$$

[0117] 该数学模型明确了目标函数即以一组数据运行被测程序时，程序穿越的路径尽可能地为目标路径。约束条件为输入数据一定在给定的输入空间内。

[0118] S54：子任务和总任务协同交叉生成测试数据。

[0119] 在该步骤提出的协同交叉人工蜂群测试数据生成算法中包含两类种群，除了若干个子种群外，还有合作团体群即总种群，且两类种群进化方式不同。

[0120] (1)子任务测试数据生成

[0121] 子种群只优化与某一进程子任务路径有关的输入分量。在进化的过程中，以解码后的进化个体作为该进程的输入，执行该进程获得穿越路径，由个体穿越路径和目标子路径，计算得到个体的适应度值。当到达指定的进化周期时，子种群根据适应度，选择一定数量的优良个体组成代表个体集合发送给合作团体群，并等待接收合作团体群返回的优良个

体，如果接收到的个体为空，表示已经找到了期望的测试数据，进化终止。否则继续进化。

[0122] (2)总任务测试数据生成

[0123] 合作团体群收到各子种群发送来的代表个体后，为了保证合作团体群具有性能良好的初始种群，根据每个子种群进化的输入分量集合，对代表个体进行合理组合，形成本种群的初始进化个体。组合方式如下：

[0124] 首先，求输入变量与子种群进化的输入分量集合的差集；然后，从其他子种群的代表个体中，提取与差集分量相关的值，扩展被差输入分量集合对应子种群的代表个体，从而产生合作团体群的进化个体，用来进化产生覆盖目标路径的期望测试数据；此外，为了产生新的进化个体，求两子种群进化的输入分量集合的交集，并从其他子种群的代表个体中，提取与交集分量相关的值，替换已产生的进化个体的相应值，进而，产生新的进化个体；最后，若产生的进化个体不同于已存在的进化个体，且在进化个体中，至少包含一个完整的代表个体，那么，将该进化个体加入到合作团体群中。

[0125] 得到本种群的初始进化个体后，在给定周期的每一代进化中，解码进化个体作为整个并行程序的输入，通过执行整个程序得到穿越路径，由个体穿越路径和目标路径，得到个体的适应度。若有个体的适应度为1，也就是说，该个体为期望的测试数据，输出个体，终止进化，同时给子种群发送空信息，终止所有种群的进化。否则继续进化生成子代种群。当到达进化周期时，根据每个子种群进化的输入分量，将优势个体进行划分，并发送到相应的子种群中。

[0126] 本发将人工蜂群算法、单点交叉和协同机制很好的融合起来，从而解决了因并行程序自身特点而带来的测试数据生成问题。同时适应度函数作为人工蜂群算法的数据评价标准，决定了算法的优劣性，所以综合考虑路径相似度中的权重问题，对适应度函数进行了针对性的改进。采用哈夫曼编码，在子路径的计算中，考虑了编码位和路径本身的重要性，在总路径的计算中考虑每一子路径对总路径的贡献度，从而设计出更适合MPI并行程序的适应度函数。本发明在通信确定的MPI并行程序的测试数据生成中，数据迭代次数和测试数据生成时间明显减少，数据效果更好，具有非常优秀的性能。

[0127] 以上内容仅为说明本发明的技术思想，不能以此限定本发明的保护范围，凡是按照本发明提出的技术思想，在技术方案基础上所做的任何改动，均落入本发明权利要求书的保护范围之内。

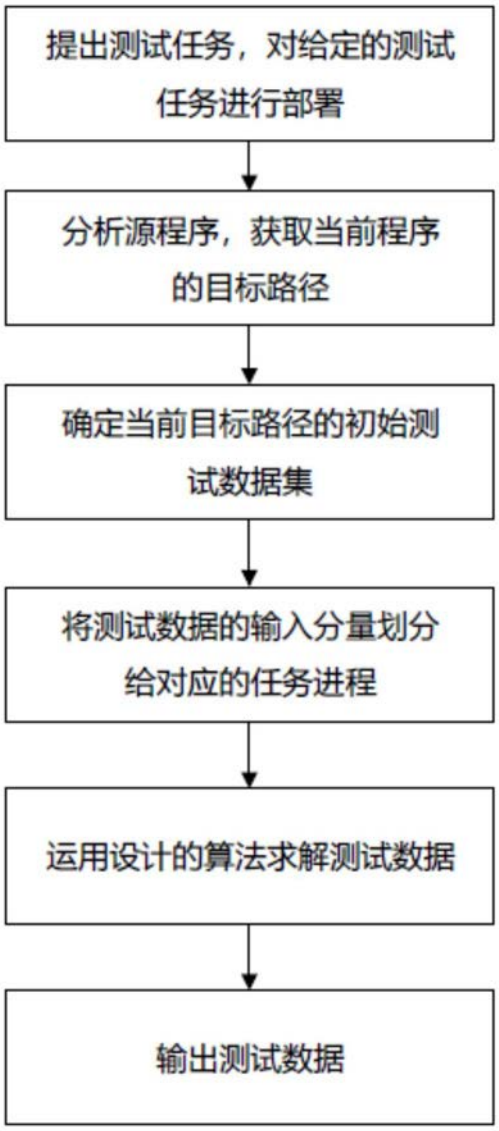

提出测试任务，对给定的测试任务进行部署
分析源程序，获取当前程序的目标路径
确定当前目标路径的初始测试数据集
将测试数据的输入分量划分给对应的任务进程
运用设计的算法求解测试数据
输出测试数据

图1

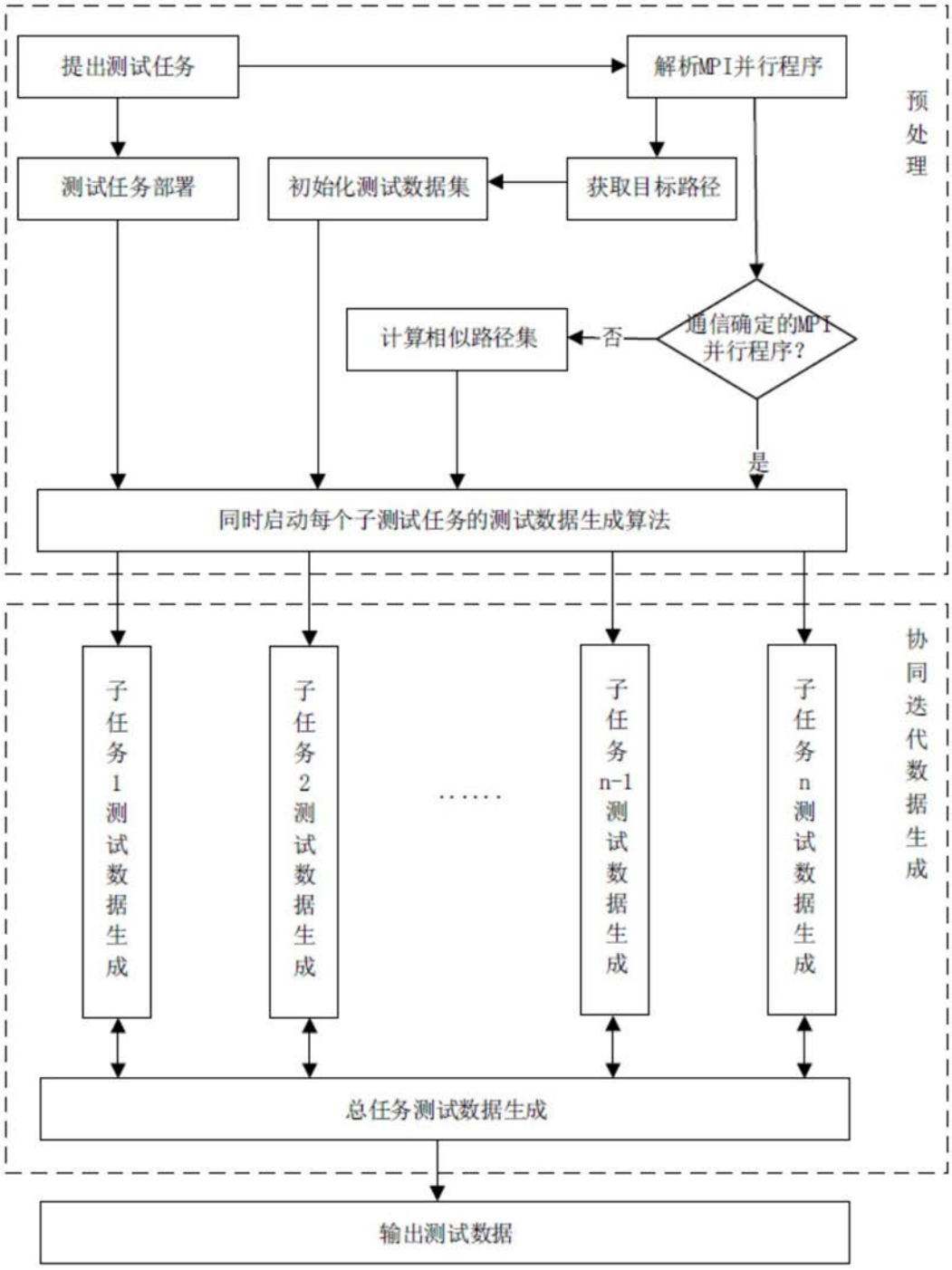

提出测试任务
解析MPI并行程序
预处理
测试任务部署
初始化测试数据集
获取目标路径
通信确定的MPI并行程序?
否
计算相似路径集
是
同时启动每个子测试任务的测试数据生成算法
协同迭代数据生成
子任务1测试数据生成
子任务2测试数据生成
.....
子任务n-1测试数据生成
子任务n测试数据生成
总任务测试数据生成
输出测试数据

图2

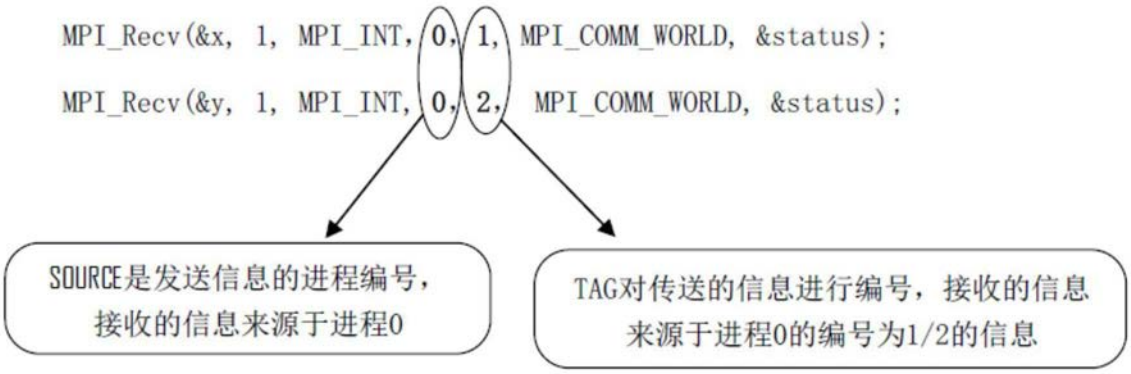

MPI_Recv(&x, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, &status);
MPI_Recv(&y, 1, MPI_INT, 0, 2, MPI_COMM_WORLD, &status);
SOURCE是发送信息的进程编号，
接收的信息来源于进程0
TAG对传送的信息进行编号，接收的信息
来源于进程0的编号为1/2的信息

图3

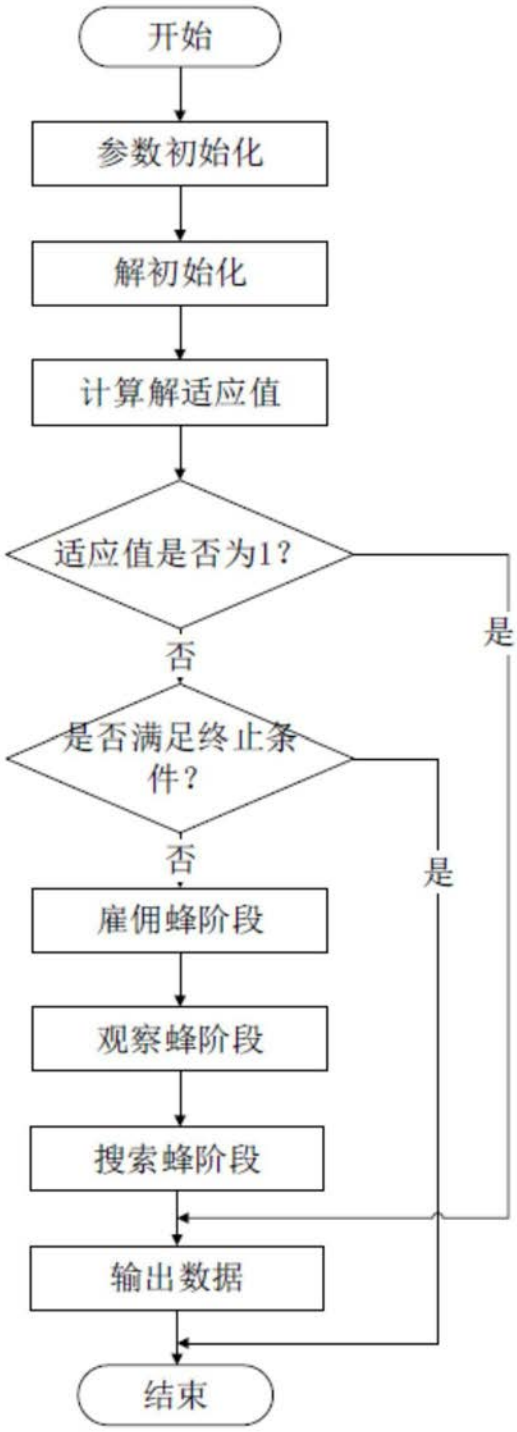

开始
参数初始化
解初始化
计算解适应值
适应值是否为1?
是
否
是否满足终止条件?
是
否
雇佣蜂阶段
观察蜂阶段
搜索蜂阶段
输出数据
结束

图4

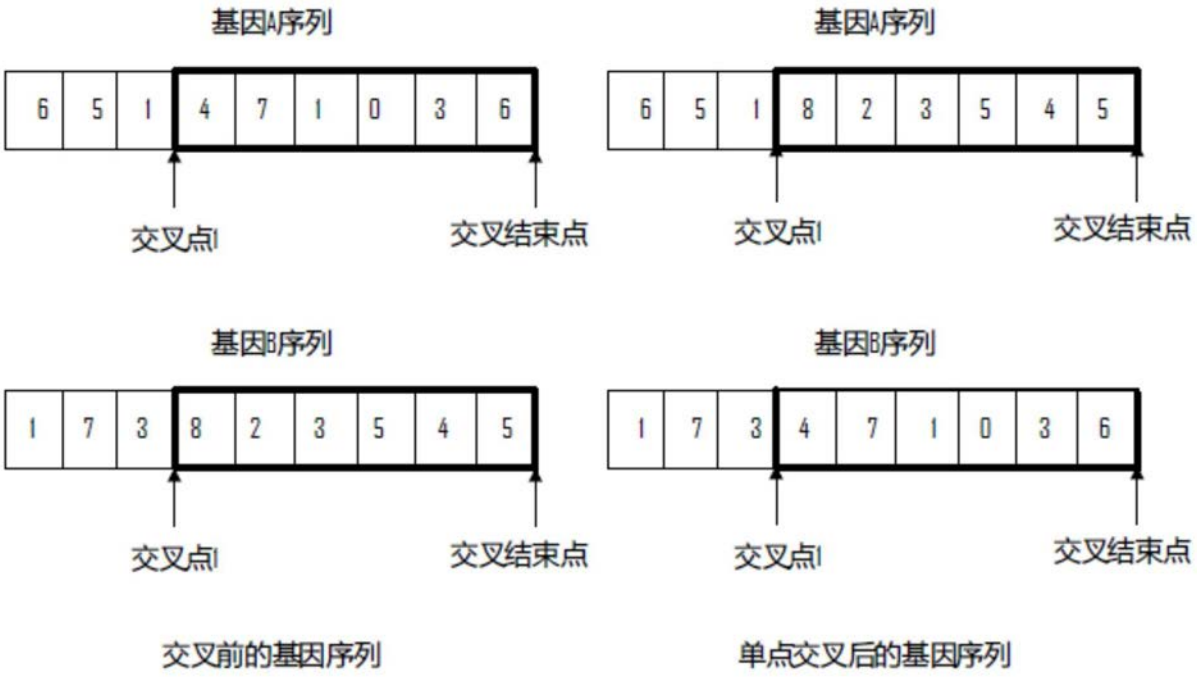

基因A序列
6 5 1 4 7 1 0 3 6
交叉点1
交叉结束点
基因A序列
6 5 1 8 2 3 5 4 5
交叉点1
交叉结束点
基因B序列
1 7 3 8 2 3 5 4 5
交叉点1
交叉结束点
基因B序列
1 7 3 4 7 1 0 3 6
交叉点1
交叉结束点
交叉前的基因序列
单点交叉后的基因序列

图5

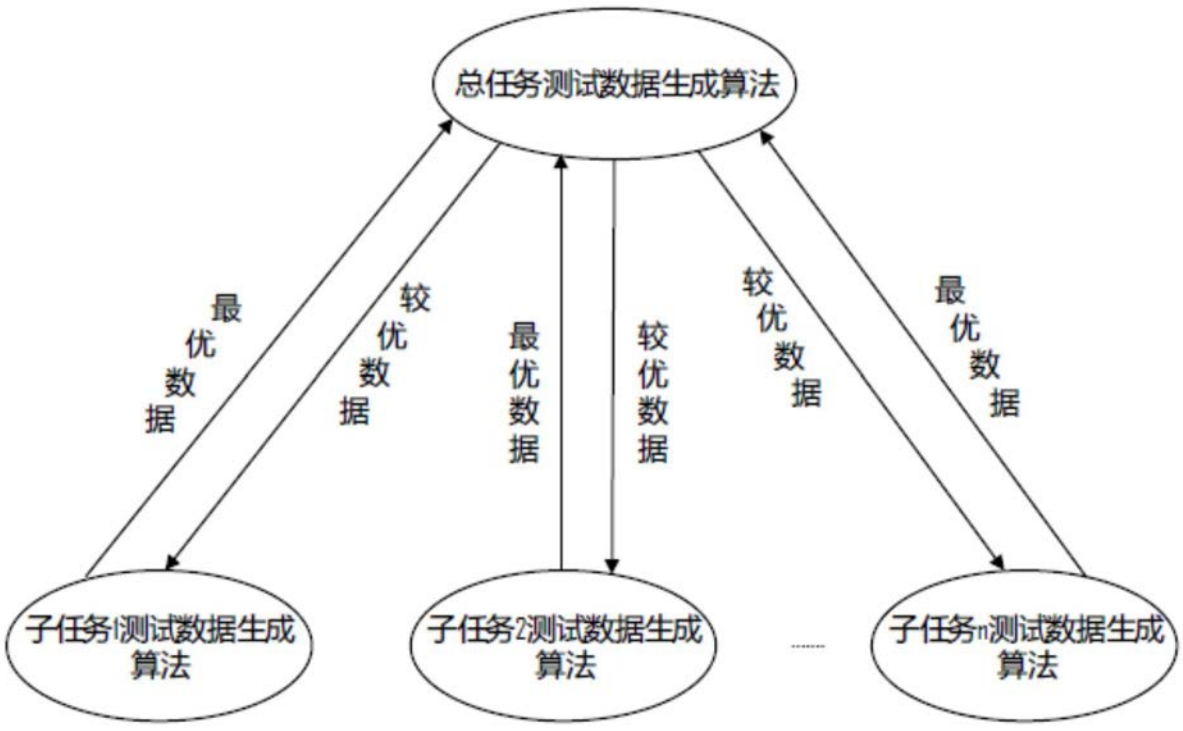

总任务测试数据生成算法
最优数据
较优数据
最优数据
较优数据
较优数据
最优数据
子任务1测试数据生成算法
子任务2测试数据生成算法
子任务n测试数据生成算法

图6