

基于遗传算法的系统发生树构建方法

郭静¹, 王超¹, 陈峻^{2,3}

GUO Jing¹, WANG Chao¹, CHEN Ling^{2,3}

1.扬州工业职业技术学院 电子信息工程系, 江苏 扬州 225127

2.扬州大学 信息工程学院, 江苏 扬州 225009

3.南京大学 软件新技术国家重点实验室, 南京 210093

1.Department of Electronic and Information Engineering, Yangzhou Polytechnic Institute, Yangzhou, Jiangsu 225127, China

2.College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225009, China

3.National Key Lab of Novel Software Technology, Nanjing University, Nanjing 210093, China

E-mail: guojing@ypi.edu.cn

GUO Jing, WANG Chao, CHEN Ling. Phylogenetic tree constructing algorithm based on genetic algorithm. Computer Engineering and Applications, 2009, 45(16): 72-76.

Abstract: A method for Phylogenetic Tree Construction based on Genetic Algorithm(GA-PTC) is presented. GA-PTC first encodes the possible tree topologies into the solution space of the problem, and then searches for the optimal tree in the searching space. To coding the solutions, this paper proposes a suffix representation as the encoding strategy of genetic algorithm. To evaluate quality of the individual, this paper adopts a distance based fitness function to score solution, and selects parts of individuals according to a selection strategy called roulette wheel where the select probability is proportional to the fitness value. Experimental results show that GA-PTC can obtain higher accuracy results than UPGMA.

Key words: genetic algorithm; phylogenetic tree; suffix representation

摘要:提出了一种基于遗传算法的系统发生树构建方法。将遗传算法应用于系统发生树的构建, 首先, 用后缀表示法将树的拓扑结构表示成编码的形式。其次, 针对系统发生树的性质, 设计了交叉和变异操作方法, 确定了对个体的评价及选择策略, 从而通过遗传操作, 最终搜索到最优解。实验结果表明该算法可以得到与传统 UPGMA 算法拓扑结果一致的系统发生树, 并且除了最优拓扑结构的树之外, 该算法还可以输入多个具有相似质量的树。

关键词:遗传算法; 系统发生树; 后缀表示

DOI:10.3778/j.issn.1002-8331.2009.16.020 **文章编号:**1002-8331(2009)16-0072-05 **文献标识码:**A **中图分类号:**TP301

1 介绍

在生物信息学中, 根据现有生物基因或物种多样性来重建生物的进化史, 即系统发生分析问题, 是一个非常重要的问题。一个可靠的系统发生的推断, 将揭示出有关生物进化过程的顺序, 有助于人们了解生物进化的历史和进化机制。

进化历史可以表示成一棵树, 即系统发生树。由于能揭示不同物种和基因之间的关系, 系统发生树在分子遗传学、医学、及药物设计中有很重要的意义, 能为之提供洞察并促进其发展。一棵表示 n 个物种之间进化关系的系统发生树通常用二叉树来表示, 它可以是已标明最近共祖先分类单元(the most recent common ancestor of the taxa)所在位置的有根树, 也可以是最近共祖先分类单元所在位置未知的无根树。一棵系统发

生树由叶子、分枝点(内部结点)和分枝长度组成, 分别表示物种、祖先和两个分枝点之间或分枝点与叶子之间的距离。每个有子孙的分枝点表示其子孙的最近公共祖先, 分枝长度对应于时间评估。系统发生树中的叶子结点, 被称作分类单元, 通常表示一个对象(蛋白质、物种等)的集合, 是一项研究中被选定的基本单位。分枝点通常称为假定分类单元, 因为它们不能被直接观察到。

系统发生树构建问题是一个 NP^{完全}问题, 由任意 n 个物种的集合构建系统发生树可以得到很多不同拓扑结构的树。随着研究物种数目 n 的增加, 可能的系统发生树的数目呈指数级增长。公式(1)与(2)分别表示了计算有根树和无根树数目的方法, 记有根树和无根树的数目分别为 NT_r 和 NT_u 。例如:

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60673060, No.60773103); 江苏省自然科学基金(the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2008206); 江苏省教育厅自然科学基金(the Science Foundation of Jiangsu Educational Commission of China under Grant No.08kj520012)。

作者简介:郭静(1981-), 女, 助教, 主要研究领域为并行计算, 算法优化; 王超(1982-), 男, 助教, 主要研究领域为机器学习, 车间调度; 陈峻(1951-), 男, 教授, 博士生导师, 主要研究领域为并行计算, 数据挖掘, 算法优化等。

收稿日期:2009-02-16 **修回日期:**2009-04-03

当用最大似然法构建15个物种的系统发生树时,可以得到213 458 046 676 875棵不同的树。由此可见,要对每种可能的进化树都进行穷举,显然是不可能的。

$$NT_u = \frac{(2n-3)!}{2^{n-2}(n-2)!} \quad (1)$$

$$NT_r = \frac{(2n-5)!}{2^{n-3}(n-3)!} \quad (2)$$

目前构建系统发生树的方法可以分为两大类:基于距离的方法,如UPGMA(非加权平均法)^[2]和邻接法^[3];基于特征的方法,如最大简约法和最大似然法。当用生物序列作为算法的输入时,两种方法都有其各自分析输入数据的方法。基于特征的方法通常比诸如多重序列比对的信息;而基于距离的方法在构建系统发生树之前利用相似性的原则构建距离矩阵。本文的方法可归为第一类,即基于距离的方法。

2 遗传算法

遗传算法(GA)是一种有效地解决最优化问题的方法。它最先由J.H.Holland于1975年提出^[4],目前已经成为进化计算研究的一个重要分支。遗传算法是模拟达尔文的遗传选择和自然淘汰的生物进化过程的计算模型。它的思想源于生物遗传学和适者生存的自然规律,是具有“生存+检测”的迭代过程的搜索算法。其中,选择、交叉和变异构成了遗传算法的遗传操作;参数编码、初始群体的设定、适应度函数的设计、遗传操作设计、控制参数的设定等五个要素组成了遗传算法的核心内容^[5]。作为一种新的全局优化搜索算法,遗传算法以其简单通用、鲁棒性强、适于并行处理以及高效、实用等显著特点,在组合优化、机器学习、信号处理、自适应控制和人工生命等领域得到了广泛应用,取得了良好效果,并逐渐成为重要的智能算法之一^[6]。

2.1 算法的基本思想

本节提出了一种基于遗传算法的系统发生树构建方法。算法以初始群体中的所有个体为对象,利用随机化技术指导对一个被编码的参数空间进行高效搜索,从而求得问题的最优解,即最满足条件的系统发生树。算法的基本过程描述如下:

- (1) 随机产生一组合法的系统发生树的后缀表示编码,构成初始种群;
- (2) 根据适应度函数 $fitness(T_k)$ 计算每一个体的适应度值,并进行排序;
- (3) 判断算法收敛性是否满足。若满足则输出搜索结果;否则执行下一步;
- (4) 根据赌轮盘选择策略从种群中选择出部分个体;
- (5) 按照交叉概率 p_c 执行交叉操作;
- (6) 按照变异概率 p_m 执行变异操作;
- (7) 返回步骤(2)。

其中,编码是算法解决问题的一个关键步骤。为将一组可能拓扑结构的系统发生树转化成遗传算法所能处理的搜索空间,提出后缀表示法,用一个字符序列表示一棵系统发生树。步骤(2)中,对个体进行评价时,采用基于距离设计的适应度函数对个体记分,并在步骤(4)中根据选择概率与适应度成正比的赌轮盘选择策略从父代中选择遗传操作的部分个体。算法对编码空间进行选择、交叉和变异等遗传操作来搜索最优解,根据系统发生树的性质设计交叉和变异算子。算法的核心内容是:编码,初始群体的设定,适应度函数的设计,选择策略,遗传算

子的设计等几个要素,本文将以此顺序来介绍本节中的内容。

2.2 后缀表示

为将系统发生树转化为遗传算法可以操作的编码,并便于遗传操作的实现,本文采用后缀表示方法表示系统发生树。科学研究证明,系统发生树一般是一棵二叉树。若一棵系统发生树中有 n 个叶子结点,在有根树中,内部结点数应为 $n-2$ 个,根结点数为1;若为无根树,则内部结点数为 $n-1$ 。综合两种情况,以下提到的内部结点包括根结点。在后缀表示方法中,系统发生树用一个字符序列来表示。字符序列中包含两种字符,分别表示树中的叶子结点和内部结点,其中, $n-1$ 个加号(+)表示内部结点,1到 n 个数字表示叶子结点,即 n 个待研究的物种序列。这样一棵系统发生树就可以用由1到 n 的数字和 $n-1$ 个加号构成的满足特定条件的序列来表示。

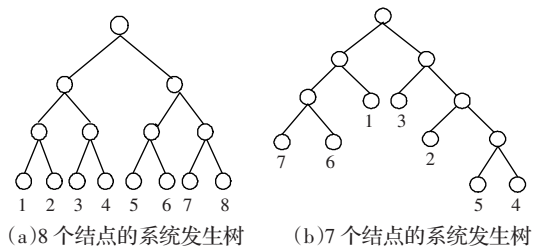


图1 两棵系统发生树

如图1(a)中的系统发生树,用数字和加号在编码中的顺序来表示叶子结点和内部结点的关系。如果叶子结点1和2有一个共同的祖先,则把表示祖先的内部结点放在1和2的后边,表示成如“1 2 +”的形式,同理,如果两棵子树“1 2 +”和“3 4 +”有一个公共祖先,则把子树看作一个整体,用同样的形式表示它们的关系,表示成如“1 2 + 3 4 +”的形式。依此类推,如图1(a)所示的整棵树用后缀表示可以编码为:“1 2 + 3 4 + + 5 6 + 7 8 + + +”。由此,给出一棵二叉树,可以很容易对其编码。例如,给定一棵如图1(b)所示的系统发生树,可以将其用后缀表示法编码为:7 6 + 1 + 3 2 5 4 + + + +;反之,如果已知叶子结点的标号为1,2,3, ..., n ,内部结点用加号来表示,满足以下约束条件的编码序列是一个合法编码序列:

- (1) 序列中必须包含1到 n 的所有数字;
- (2) 序列中必须包含 $n-1$ 个加号(+);
- (3) 按从左到右的顺序,序列中加号的数目最多是其前面数字数目减去1。

约束(3)中的合法性用栈结构可以很容易地判断,如图2。给一个编码设计一个计数器 $count$, 给数字和加号分别安排 $count$ 值为+1和-1,一个编码的 $count$ 值是它所有元素的 $count$ 值的总和。如果用后缀表示的一个编码是合法的,则它的 $count$ 值必须是1。此外,编码所对应的树的任意子树的 $count$ 值也必须为1。当在按顺序计算一个合法的编码的元素值时, $count$ 值应该永远都大于1。例如,图1(b)中所对应树的编码的 $count$ 值的计算过程可以用图2来表示。

count 值	1	2	1	2	1	2	3	4	5	4	3	2	1
	↑	↑	↓	↑	↓	↑	↑	↑	↑	↓	↓	↓	↓
编码	7	6	+	1	+	3	2	5	4	+	+	+	+

图2 编码技术的栈结构表示

此表示法可以明确反映出叶子结点与内部结点在树中的位置关系,并能使一个编码唯一对应一棵系统发生树,反之亦然。给定一个满足以上条件的编码序列,可以很容易地构造出此序列对应的系统发生树。例如:序列“1 2 + 3 4 5 + 6 7 + 8 + + +”对应的系统发生树如图3所示。

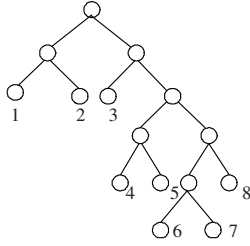


图3 编码技术的栈结构表示

根据以上的编码方式可以随机地产生初始种群,初始种群中的每一个个体都是由1到n的数字和n-1个加号组成的可以表示一棵系统发生树的序列,其产生过程为:

随机产生1到n的一种数字排列。

然后在n个数字中插入n-1个加号,使得:序列的count值在按从左到右的计算过程中始终大于1。

初始种群的设定对算法结果的影响很大,本文中是经验性选择初始种群的。

2.3 适应度函数及选择策略

算法中的适应度函数的作用是,对当前群体 $X(t)$ 中每个个体 x_i 计算其适应度 $f(x_i)$,适应度表示了该个体的性能好坏,根据适应度值选择遗传操作的个体。文中适应度函数用物种之间的距离来表示。当系统发生树的拓扑结构及树的分枝长度确定之后,便可以得到一个新的距离矩阵 D_k 。通过比较新的距离矩阵 D_k 和原来距离矩阵 D ,可以得到该树的适应度函数 $fitness(T_k)$,并用其来衡量该树的质量。系统发生树 T_k 的适应度函数可以用公式(3)计算,其中,n为物种的个数,C是一个常数, $D_k[i,j]$ 及 $D[i,j]$ 分别表示矩阵 D_k 及 D 的 (i,j) 元素。

$$fitness(T_k) = C \cdot \left[\frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(\frac{D_k[i,j] - D[i,j]}{D[i,j]} \right)^2}{\frac{n(n-1)}{2} - 1} \right]^{\frac{1}{2}} \quad (3)$$

求和公式扩展到n个物种之间所有 $n(n-1)/2$ 个两两距离。若物种在系统发生树中的距离等于原始距离,即距离矩阵 D 等于 D_k ,则 $fitness$ 值将达到最小值0,若两者相差很大,则 $fitness$ 值变得较大。

在算法运行的过程中,竞争是通过父辈的选择来体现的。根据公式(3)计算种群的适应度值之后,在适应度值基础上按照某种选择策略进行父代个体的选择,并确定交叉或变异个体。本文采用的选择机制是选中概率和评价值成正比的赌轮盘(Roulette wheel)机制。在每一代的进化过程中,首先按照每个个体的适应度值的大小分割轮盘的位置:令 $\sum f_i$ 表示群体的适应度值的总和, f_i 表示种群中第i个个体的适应度值,它在轮盘上所占的比例为 $f_i / \sum f_i$ 。个体的适应度值越大,则在轮盘上占有的面积越大,被选中的概率越大。然后随机选取轮盘上的一

点,其所对应的个体即被选入到交配池中。

2.4 交叉和变异算子的设计

在算法的第(5)步中,执行交叉操作,交叉的本质是从种群中选择父代以生成新的种群。当以概率 p_c 选择好了两个要交叉的个体之后,用以下方式对两个个体进行交叉操作。

- (1)随机选取某一范围内的字符串互换,窗口大小设为s;
- (2)检查窗口内加号个数的合法性:若一个编码中的加号个数少于另外一个编码中的加号个数,则后移(或前移)窗口,调整要互换的编码;
- (3)互换两个编码的窗口内容,生成两条新序列;
- (4)新序列中加号合法性检查:若互换后编码中的加号位置不合法,则将加号后移;
- (5)新序列中数字合法性检查:若编码中有重复的数字,则与对方编码中的重复数字互换。

在执行交叉操作后,生成的新的系统发生树必须仍然由原来的物种集合组成,并且是一棵二叉树。物种数目一定的情况下,二叉树的内部结点数目是一定的。因此,步骤(2)中要求要交换的两个窗口中加号数目相等,以使得新的系统发生树中的内部结点数目不变,即还是一棵二叉树。步骤(4)检查交换后新序列中的加号合法性,使得生成的后缀表示序列是合法的。步骤(5)保证系统发生树的叶子结点包含所有物种集合中的序列。

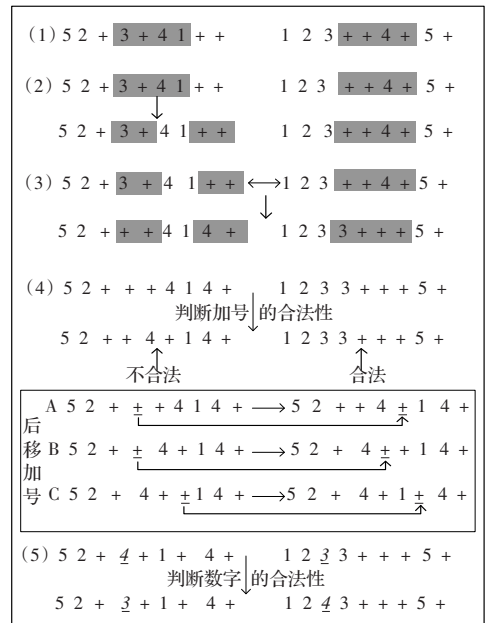


图4 交叉操作的过程

例如,图4中,两个个体“5 2 + 3 + 4 1 + +”和“1 2 3 + + 4 + 5 +”按照上述步骤执行交叉操作,设窗口大小为4。具体方法为:

- (1)在两个个体中随机选取一个交换窗口,设为“3 + 4 1”和“+ + 4 +”。
- (2)检查窗口内加号个数的合法性,发现第一个窗口中加号个数为1,而第二个中加号个数为3,加号个数不相等。于是后移第一个窗口,直到其加号个数等于3为止。如图所示第一个窗口内容变为“3 + + +”。
- (3)交换两个窗口的内容,两个个体分别变为“5 2 + + + 4 1 4 +”和“1 2 3 3 + + + 5 +”。

(4)判断个体中加号的合法性。根据“按从左到右的顺序,序列中加号的数目最多是其前面数字数目减去1”的原则,发现第一个个体中加号是不合法的,第二个个体中加号是合法的。于是将第一个个体中不合法的加号依次后移,具体方法为:找到序列中第一个不合法的加号(图中以下划线表示不合法),将其移至其后第一个数字“4”之后,序列变为“5 2 + + 4 + 1 4 +”(图4中A部分所示)。此时,序列中仍有不合法的加号,再次找到第一个不合法的加号,同样移至其后第一个数字之后(图中B部分所示)。依次类推,将最后一个不合法的字符移至其后第一个数字之后(图中C部分所示),至此序列中加号的位置已经全部合法。

(5)检查序列中数字的合法性。从图中可以看出,两条序列中各有一个重复字符“4”和“3”,将各序列中第一个重复字符互换。至此,两条合法的序列已经生成,即产生了两个新个体:“5 2 + 3 + 1 + 4 +”和“1 2 4 3 + + + 5 +”。

在算法的第(6)步中,执行变异操作。变异的目的是防止种群中的解陷入局部最优,通过改变个体中的某一基因以改变搜索方向和扩大搜索空间。算法采用以下方式进行变异操作(如图5所示):

- (1)随机选择编码中的任意两个位置;
- (2)交换该位置上的字符;
- (3)检查加号位置的合法性,若不合法,则加号后移,直至编码变为合法序列为止。

在步骤(3)中进行加号合法性检查,以使生成的新序列能合法地表示一棵系统发生树。例如,图5中,对个体“1 2 + 3 4 + 5 +”进行变异操作:如图所示,选中序列中阴影部分的字符交换,生成一个新的序列。然后检查加号的合法性,若加号合法,则新个体生成;否则,用交叉操作第(4)步中的方式后移加号,直到序列变得合法为止。

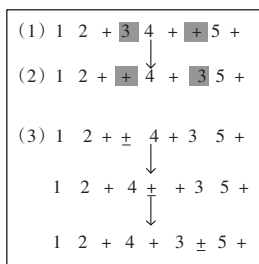


图5 变异操作的过程

2.5 参数设置

算法中有两个基本参数:交叉概率 p_c 和变异概率 p_m 。交叉过程的目的是,由拓扑结构中较好的部分组合成新的系统发生树,从而使得新的系统发生树与旧个体相比,得到的距离更接近物种之间的原始距离矩阵。变异的目的是为了防止算法陷入局部最优,在旧基因的基础上改变部分基因以得到更多拓扑结构,从而增大解的搜索空间,但变异概率不能太大,否则算法就会变成随机搜索。采用试探法确定交叉概率和变异概率。此外,种群数目也是影响算法优化性能和效率的因素之一。群体规模越大就越容易找到最优解。但是群体规模与计算机的内存容量及运行时间有关,如果取得过大,算法的复杂度会增加,一般来说,应在计算条件允许的范围内尽可能取大一些。算法的参数设置详见表1。

表1 遗传算法的参数设置

初始种群大小=50	变异概率=0.01
代数=500	交叉概率=0.8
选择方法=Roulette wheel	交叉窗口大小=6

3 实验结果及分析

从PTC(Phylogenetic Tree Construction package)所提供的三类物种中选择了两类数据用本章的算法GA-PTC(GA Phylogenetic Tree Construction)和UPGMA^[7]方法进行了实验,PTC软件^[8]的下载地址为:<http://www.securebio.umb.edu/cgi-bin/TreeConstructor.pl>,它提供了Animal、Plant、Other三类生物序列,并对对选中的序列进行比对和构建系统发生树。选择一组包含20条序列的数据进行实验。

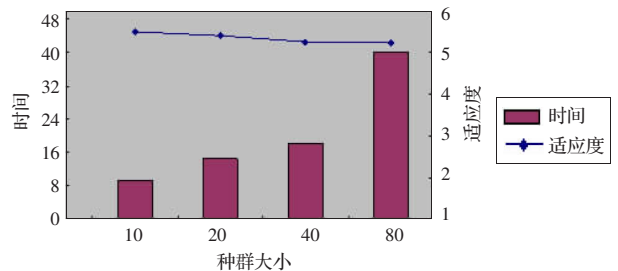


图6 种群大小对算法的影响

图6说明了种群大小对算法性能的影响,在其他设置固定的情况下,种群大小设为10到80之间的不同数值。从图中可以看出,当种群大小达到80时,算法消耗的时间急剧增长,比种群大小为10时多消耗近80%的时间,但是适应度值的变化曲线却比较平缓。实验中种群大小设为50。

用两种方法构建系统发生树的结果如图7所示。从图中可以看出用GA-PTC和UPGMA方法构建的系统发生树很相似,两种方法得到的树的拓扑结构基本一致,这说明用GA-PTC可以得到输入物种的正确分类。

表2是用两种方法构建的系统发生树的适应度和时间比较结果。其中,对于GA-PTC给出了20次单独运行得到的平均值,而UPGMA运行多次都得到同样的结果。在用真实数据对两种算法进行实验比较时,由表2中适应度函数值的比较可以看出,GA-PTC得到的树比UPGMA得到的树的适应度值要略高一些,另外,GA-PTC所用的时间也比UPGMA少。

表2 用GA-PTC和UPGMA得到的系统发生树对应的适应度值的比较

方法	适应度值	计算时间/s
UPGMA	5.79	0.70
GA-PTC	5.31	0.61

实验中给出的结果是根据遗传算法找到的一棵最优树,在算法执行时,除了最优树之外,还可以得到一组相似质量的树。当给定的构建系统发生树的数据集不确定或不完整时,可能需要多种可能的拓扑结构,所以一组相似质量的树是很有必要的,而UPGMA以及其他方法都只产生一棵树的拓扑结构。但是,遗传算法的性能受参数设置的影响很大,文中虽然给出了算法的基本参数设置,但并不是最佳参数。例如,当要分析的物种数目变大时,需要对参数进行调整,特别是交叉窗口应该调大。若实验中所用到的物种序列的长度改变时,GA-PTC的参数设置也需要随之调整。

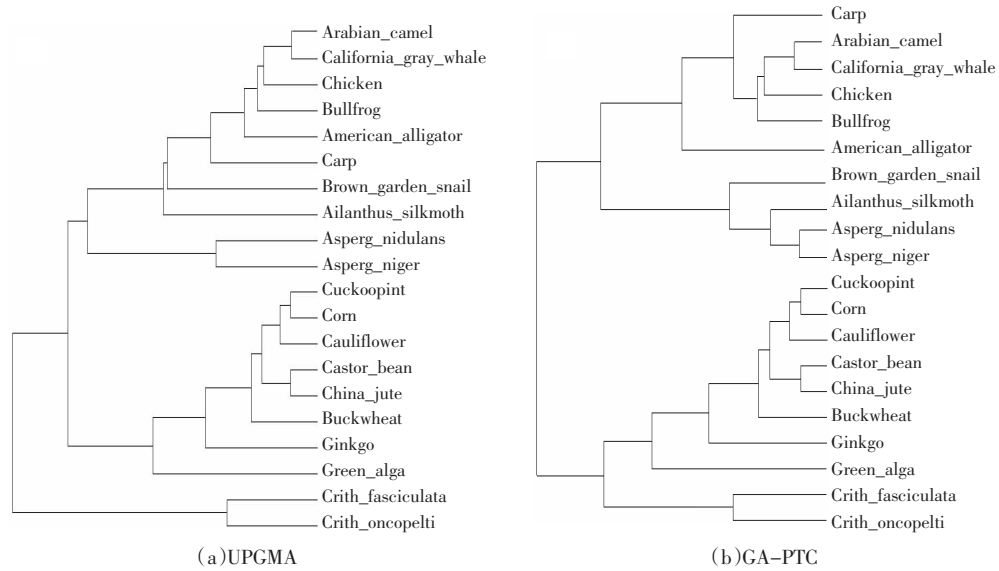


图7 用两种方法构建出来的系统发生树的拓扑结构

4 结论

本文提出的基于遗传算法的构建系统发生树的方法 GA-PTC 中,采用后缀表示法对系统发生树进行编码,用基于距离的适应度函数对个体进行评价,从初始群体出发,经过一系列的选择、交叉、变异等遗传操作,搜索最优解。从算法与传统系统发生树构建算法 UPGMA 的实验结果比较中可以看出,本文的算法可以得到与 UPGMA 一致的拓扑结构的系统发生树,并且从适应度函数值的比较中得出,本文算法更优。此外,除了最优解之外,本文算法还可以输出具有相似质量的拓扑结构,而 UPGMA 以及其他方法都只产生一棵树的拓扑结构,这一点在利用不确定的数据集构建系统发生树时显得尤为重要。

参考文献:

- [1] Bodlaender H L, Fellows M R, Warnow T J. Two strikes against perfect phylogeny [C]//Proc of 19th International Colloquium on Au-

- tomata Languages and Programming, 1992, 8(3):56-67.
 [2] Sokal R R, Michener C D. A statistical method for evaluating systematic relationships [J]. University of Kansas Scientific Bulletin, 1958, 28: 1409-1438
 [3] Saitou N, Nei M. The neighbor-joining method: A new method for reconstructing phylogenetic trees [J]. Molecular Biology and Evolution, 1987, 4: 406-425.
 [4] Holland J H. Adaptation in natural and artificial systems [M]. Ann Arbor: The University of Michigan Press, 1975.
 [5] 华灵群. 基于遗传算法的旅行商问题仿真计算 [D]. 西北工业大学, 2006.
 [6] 李长河. 人工智能及其应用 [M]. 北京: 机械工业出版社, 2006.
 [7] Rohlf F J. Classification of Aedes by numerical taxonomic methods (Diptera: Culicidae) [J]. Ann Entomol Soc Am, 1963, 56: 798-804.
 [8] Chen Yang, Sami Khuri. PTC: An interactive tool for phylogenetic tree construction [C]//Proceedings of the IEEE Computer Society Bioinformatics Conference CSB'03, 2003.

(上接 68 页)

参考文献:

- [1] Pawlak Z. Rough sets: Theoretical aspects of reasoning about data [M]. Boston: Kluwer Academic Publishers, 1991.
 [2] Pawlak Z. Rough sets [J]. Communication of the ACM, 1995, 38(1): 89-95.
 [3] 王国胤. Rough 集理论与知识获取 [M]. 西安: 西安交通大学出版社, 2001.
 [4] 苗夺谦, 王珏. 基于粗糙集的多变量决策树构造方法 [J]. 软件学报, 1997, 8(6): 425-431.
 [5] 米据生, 吴伟志, 张文修. 不协调目标信息系统知识约简的比较研究 [J]. 模糊系统与数学, 2003, 17(3): 54-60.
 [6] 王珏, 苗夺谦, 周育健. 关于 Rough Set 理论与应用的综述 [J]. 模式识别与人工智能, 1996, 9(4): 337-344.
 [7] 张文修, 梁怡, 吴伟志. 信息系统与知识发现 [M]. 北京: 科学出版社, 2003.
 [8] 张文修, 米据生, 吴伟志. 不协调目标信息系统的知识约简 [J]. 计算机学报, 2003, 26(1): 12-18.
 [9] Kryszkiewicz M. Comparative studies of alternative of knowledge reduction in inconsistent systems [J]. Intelligent Systems, 2001, 16(1): 105-120.
 [10] Grecos M B, Slowinski R. Rough approximation of preference relation by dominance relations [J]. European Journal of Operational Research, 1999, 117: 63-68.
 [11] Shao M W, Zhang W X. Dominance relation and rules in an incomplete ordered information system [J]. International Journal of Intelligent Systems, 2005, 20: 13-27
 [12] Xu W H, Zhang W X. Methods for knowledge reduction in inconsistent ordered information systems [J]. Journal of Applied Mathematics & Computing, 2008, 26(1/2): 313-323.
 [13] 徐伟华, 张文修. 基于优势关系下协调近似空间 [J]. 计算机科学, 2005, 32(9): 164-165.
 [14] 徐伟华, 张文修. 基于优势关系下不协调目标信息系统的知识约简 [J]. 计算机科学, 2006, 33(2): 182-184.
 [15] 徐伟华, 张文修. 基于优势关系下不协调目标信息系统的分布约简 [J]. 模糊系统与数学, 2007, 21(4): 124-131.