基于遗传算法的二叉树画树算法*

黄竞伟 康立山 陈毓屏

(武汉大学软件工程国家重点实验室 武汉 430072)

E-mail: jwhuang@public.wh.hb.cn

摘要 用遗传算法设计了一种通用二叉树画树算法框架.在该框架下,可以根据应用的不同,通过设计反映美观标准的不同目标函数来得到不同的画树算法,而且容易添加或减少美观标准以适应不同用户的需要.与以前的算法相比,此算法具有算法统一、方法简单、容易实现和易于修改的优点,并且具有自适应、自学习和易于并行化的特点.

关键词 二叉树,画树算法,遗传算法.

中图法分类号 TP18

将一个图结构自动、美观地画到二维或三维空间的一个有限区域中是一个具有重要理论意义和实际应用价值的课题,它在诸如信息可视化、用户接口设计工具、VLSI 布局等实际问题中都有广泛的应用.一般可将图结构分为树、平面图、一般无向图、一般有向图.对于各类图结构已有不少画图算法,文献[1]是一篇有关画图算法的综述文章.画树算法,尤其是二叉树画树算法,是其中一类重要的算法.对于二叉树画树算法,许多人都作过深入的研究[2~4].画树算法的目的是将树画到平面某个有限区域中(比如显示屏上),而算法的主要任务是在一定的美观准则下,对树中的每一个结点 v 指定一对坐标 (x_v, y_v),一旦每个结点都被指定一对坐标后,则将树画出来就是一件很容易的事情了,只需在与边相连的结点之间画出一条直线段即可.根据应用和观点的不同,会有不同的美观准则,但对于二叉树而言,通常接受以下美观标准:

(1) 二叉树中同一层的结点应画在同一条水平线上;

(2) 层与层之间应相隔一个常数距离 $h_{\min}$;

(3) 父结点应画在其儿子结点的上方;

(4) 一个结点的左儿子结点应画在该结点的左边,右儿子结点应画在该结点的右边;

(5) 同层结点之间应保持一个最小间隔 $d_{\min}$;

(6) 一个父结点应画在其左、右儿子结点的中间.

虽然已有许多快速画出满足上述美观标准的二叉树的画树算法,然而这些算法不具有重用性,针对一种应用设计出来的画树算法,一般不再适用于其他应用,而且也难于在这些算法中添加新的美观准则,以适应用户的需要.在本文中,我们用遗传算法[5]设计了一种通用二叉树画树算法框架,在该框架下,我们可以根据应用的不同,通过设计不同的反映美观标准的目标函数来得到不同的画树算法,而且容易添加或减少美观标准以适应不同用户的需要.与以前的算法相比,我们的算法具有算法统一、方法简单、容易实现、易于修改等优点,并且具有自适应、自学习和易于并行化的特点.

* 本文研究得到国家自然科学基金(No. 69635030)和国家863高科技项目基金(No. 863-306-ZT06-06-3)资助.作者黄竞伟,1956年生,教授,主要研究领域为算法设计与分析,演化计算.康立山,1934年生,教授,博士生导师,主要研究领域为计算机科学理论,计算数学,并行计算,演化计算.陈毓屏,女,1942年生,副教授,主要研究领域为并行计算,演化计算,计算机软件.

本文通讯联系人:黄竞伟,武汉 430072,武汉大学软件工程国家重点实验室

本文1999-04-08收到原稿,1999-08-10收到修改稿

1 基于遗传算法的二叉树画树算法

在下面的讨论中,我们假定 n 表示二叉树中的结点个数.当我们以某种次序对二叉树的结点编号后,则用 (x_i,y_i) 表示编号为 i 的结点在平面区域上的坐标.

用遗传算法设计画树算法的思想是将画树问题转化为函数优化问题,而这一问题的关键在于设计出能够反映美观标准的目标函数 $f(x_1,y_1,x_2,y_2,...,x_n,y_n)$,使得当满足美观标准的程度较高时,函数 $f(x_1,y_1,x_2,y_2,...,x_n,y_n)$ 的值就小一些,因而若我们希望将树画到一个有限矩形区域 $a\leqslant x\leqslant b,c\leqslant y\leqslant d$ 内,则画树问题就会转化为下列函数优化问题:

$$\mathrm{Min} f(x_1,y_1,x_2,y_2,...,x_n,y_n),\ a\leqslant x_i\leqslant b,c\leqslant y_i\leqslant d. \tag{1}$$

设计一个遗传算法的基本步骤是:(1) 编码;(2) 设计适应度函数;(3) 制定选择策略;(4) 选择控制参数;(5) 设计遗传算子;(6) 给出终止准则.下面,我们分别对上述 6 个步骤进行讨论.

(1) 编码

给定二叉树 T,设其前序遍历结点序列为 $v_1,v_2,...,v_n$,其坐标分别为 $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$,则对每个可能的解,我们用一个长度为 $2n$ 的实数串 $(x_1,y_1,x_2,y_2,...,x_n,y_n)$ 来表示.

(2) 设计适应度函数

首先,我们设计能够反映美观标准的目标函数 $f(x_1,y_1,x_2,y_2,...,x_n,y_n)$,根据上述美观准则(1)~(6),我们容易设计出下列目标函数:

$$\begin{aligned}
f=\ & k1\sum_{v\in V_1}\left(X(v)-\left(\frac{X(L(v))+X(R(v))}{2}\right)\right)^2+\\
& k2\sum_{\substack{v\in V_2\\ Y(v)>Y(L(v))-h_{\min}}}\left(|Y(v)-Y(L(v))|-h_{\min}\right)^2+\\
& k3\sum_{\substack{v\in V_3\\ Y(v)>Y(R(v))-h_{\min}}}\left(|Y(v)-Y(R(v))|-h_{\min}\right)^2+\\
& k4\sum_{v\in V_2}\left(\sqrt{(X(v)-X(L(v)))^2+(Y(v)-Y(L(v)))^2}-EdgeLength\right)^2+\\
& k5\sum_{v\in V_3}\left(\sqrt{(X(v)-X(R(v)))^2+(Y(v)-Y(R(v)))^2}-EdgeLength\right)^2+\\
& k6\sum_{\substack{v\in V_4\\ X(L(v))>X(v)-\frac{d_{\min}}{2}}}\left(X(L(v))-X(v)+\frac{d_{\min}}{2}\right)^2+\\
& k7\sum_{\substack{v\in V_5\\ X(v)>X(R(v))-\frac{d_{\min}}{2}}}\left(X(v)-X(R(v))+\frac{d_{\min}}{2}\right)^2+\\
& k8\sum_{v\in V}\sum_{\substack{v_1\in LRcontour(v)\\ v_2\in RLcontour(v)\\ Level(v_1)=Level(v_2)\\ X(v_1)>X(v_2)-d_{\min}}}\left(X(v_1)-X(v_2)+d_{\min}\right)^2.
\end{aligned}$$

其中 $L(v)$ 表示结点 v 的左儿子,$R(v)$ 表示结点 v 的右儿子,$X(v)$ 表示结点 v 的 x 坐标,$Y(v)$ 表示结点 v 的 y 坐标,V_1 为树中具有左儿子和右儿子结点的集合,V_2 为树中具有左儿子结点的集合,V_3 为树中具有右儿子结点的集合,V_4 为树中具有左儿子但无右儿子结点的集合,V_5 为树中具有右儿子但无左儿子结点的集合,V 为树中所有结点的集合.$LRcontour(v)$ 和 $RLcontour(v)$ 表示以结点 v 为根结点的左、右子树的右、左边界,其定义我们在下面给出.$ki(1\leqslant i\leqslant 8)$ 为常数,可根据美观标准的不同进行选择.第 4 和第 5 个和式中的 $EdgeLength$ 表示树中边的长度,虽然树中边的长度通常并不相同,但增加第 4 和第 5 两个和式,并通过 $ki(1\leqslant i\leqslant 8)$ 的选择,可使二叉

树画出来更美观一些.

在设计出目标函数 $f(x_1,y_1,x_2,y_2,\ldots,x_n,y_n)$ 之后，由于该问题是求最小值点，故我们可以选择一个常数 $C_{\max}\geqslant f(x_1,y_1,x_2,y_2,\ldots,x_n,y_n)$，直接从 $f(x_1,y_1,x_2,y_2,\ldots,x_n,y_n)$ 设计适应度函数 $F(x_1,y_1,x_2,y_2,\ldots,x_n,y_n)=C_{\max}-f(x_1,y_1,x_2,y_2,\ldots,x_n,y_n)$.

(3) 选择策略

为了防止发生早熟现象，我们首先用Sigma 比例变换技术对个体的适应值进行变换，即对个体 i 的适应值 $f(i)$，首先用公式

$$\mathrm{ExpVal}(i)=\begin{cases}1+(f(i)-f(t))/2\sigma(t), & \text{若 } \sigma(t)>0\\ 1, & \text{若 } \sigma(t)=0\end{cases}$$

将 $f(i)$ 变换到 ExpVal(i)，其中 $f(t)$ 为第 t 代群体平均适应值，而 $\sigma(t)$ 表示第 t 代群体的标准方差. 然后，对 ExpVal(i) 用基于适应值比例的选择策略，但保留适应值最高的染色体.

(4) 控制参数

在我们的算法中，取 $N=20, P_c=0.25, P_m=0.55$，它们分别表示群体的大小、杂交概率和变异概率.

(5) 遗传算子设计

杂交算子首先随机地在 1～n 中选择一个整数 r，然后将两个父体在先序遍历序列中的第 r 个结点为根的子树上进行交换. 我们采用如下的非一致变异算子：设父体为 $A=(a_1,a_2,\ldots,a_k,\ldots,a_{2n})$，第 k 个分量被选出来进行变异. 假设第 k 个分量为 y 坐标，其取值范围为 $[c,d]$，则变异后所得到的新个体为 $A'=(a_1,a_2,\ldots,a'_k,\ldots,a_{2n})$，其中

$$a'_k=\begin{cases}a_k+\Delta(t,d-a_k), & \text{如果 Random}(2)=0\\ a_k-\Delta(t,a_k-c), & \text{如果 Random}(2)=1\end{cases}.$$

这里，Random(2) 产生 0 或 1 的一个随机整数，而 $\Delta(t,y)=y\cdot(1-r^{(1-t/T)^5})$，其中 r 为 $[0,1]$ 之间的一个随机数，t 为当前演化代数，T 为最大演化代数.

(6) 终止准则

采用算法运行若干代后，终止算法.

基于遗传算法的通用画树算法框架描述如下：

```
Procedure Draw_Tree_By_GA;
Begin
  t:=0;initialize(P(t));Evaluate(P(t));
  While t<Max_Generation Do
  Begin
    P1:=Select(P(t));P2:=Crossover(P1);P(t+1):=Mutate(P2);
    Evaluate(P(t+1));t:=t+1;
  End;
  Draw_Tree(T);
End.
```

其中过程 Draw_Tree(T) 画出具有最大适应值的二叉树 T.

2 算法的实现

下面讨论若干算法的实现细节，主要讨论如何计算目标函数.

为了计算目标函数值，我们只需对二叉树进行一次前序遍历即可，在遍历过程中计算目标函数值. 例如，当访问结点 v 时，检查 v 的左儿子和右儿子结点是否为空，若都不为空，则可计算目标函数中的项：

$$k1\left(X(v)-\frac{X(L(v))+X(R(v))}{2}\right)^2,$$

$$k2(|Y(v)-Y(L(v))|-h_{\min})^2,$$

$$k3(|Y(v)-Y(R(v))|-h_{\min})^2,$$

$$k4\left(\sqrt{(X(v)-X(L(v)))^2+(Y(v)-Y(L(v)))^2}-EdgeLength\right)^2,$$

$$k5\left(\sqrt{(X(v)-X(R(v)))^2+(Y(v)-Y(R(v)))^2}-EdgeLength\right)^2.$$

为了满足美观准则(5),即同层结点之间应保持一个最小间隔 $d_{\min}$,我们需要在对给定的二叉树进行前序遍历时,对树中的每个结点 v,检查它的左子树的右边界与右子树的左边界上深度相同的结点是否具有间隔 $d_{\min}$.设 v_1是结点 v 的左子树右边界上的一个结点,v_2为结点 v 的右子树的左边界上的一个与 v_1有同一深度的结点,当$X(v_1)>X(v_2)-d_{\min}$时,则计算 $k8(X(v_1)-X(v_2)+d_{\min})^2$.

为了有效地计算目标函数中的第8个和式,对于二叉树中的每个结点 v,我们需要计算它的左、右边界.

一棵深度为 h 的二叉树 T 的左(右)边界是一个结点的序列 $v_0,v_1,\ldots,v_h$,其中 v_i 是 T 的深度为 i 的最左(右)边的结点.

下面给出二叉树边界递归构造方法.

一棵空的二叉树的左(右)边界为空.若 T 是一棵非空的二叉树,其左子树 L 和右子树 R 的边界已经构造好,则可按如下方法构造 T 的边界:

(1) 若 L 和 R 的深度相同,则 T 的左(右)边界由 T 的根结点与 L,R 的左、右边界构成;

(2) 若 L 的深度比 R 的深度小,则 T 的右边界由 T 的根结点与 R 的右边界构成.令 h'为 L 的深度,w 为 R 的左边界上深度为 $h'+1$的结点.T 的左边界由 T 的根结点、L 的左边界和 R 的左边界至 w 以下的部分构成;

(3) 若 L 的深度比 R 的深度大,T 的边界的构造与(2)类似.

有了上述子树边界的构造方法,我们就可以容易地计算出目标函数的第8个和式了.

3 算法的扩充与实验结果

用遗传算法设计画图算法的一个很大的优点在于,这样设计出来的算法具有可重用性,适应性强,可以根据不同的应用和不同的美观标准,在不改变算法基本框架的前提下,仅通过重新设计反映新的美观标准的目标函数,便可得到以满足实际需求的新的画树算法.下面,我们举例说明这一点.

若对二叉树的任意两条边(u,v)和(s,t)而言,表示(u,v)和(s,t)的直线段互不相交,且每条边都被表示为从父节点到左儿子结点的垂直线段或到右儿子结点的水平线段,二叉树的这样一种画法称为 h-v 画法.例如,图1所示的二叉树的一种 h-v 画法如图2所示.

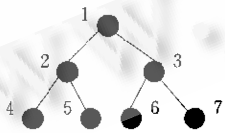

Fig. 1 Binary tree
图1 二叉树

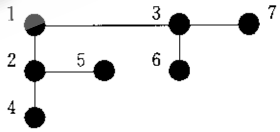

Fig. 2 h-v drawing of binary tree
图2 二叉树的h-v画法

若我们想得到二叉树的一种 h-v 画法,只需在上述画树算法中将目标函数改为下列函数:

$$g=k1\sum_{\substack{v\in V_1\\X(v)\neq X(L(v))}}(X(v)-X(L(v)))^2+$$

$$k2\sum_{\substack{v\in V_2\\Y(v)\neq Y(R(v))}}(Y(v)-Y(R(v)))^2+$$

$$k3\sum_{v\in V_1}(|X(v)-X(L(v))|-EdgeLength)^2+$$

$$k4\sum_{v\in V_2}(|X(v)-X(R(v))|-EdgeLength)^2+$$

$$k5\sum_{\substack{v\in V_1\\ Y(v)\geqslant Y(L(v))-EdgeLength}}(Y(v)-Y(L(v))+EdgeLength)^2+$$

$$k6\sum_{\substack{v\in V_1\\ X(v)\geqslant X(R(v))-EdgeLength}}(X(v)-X(R(v))+EdgeLength)^2+$$

$$k7\sum_{v\in V_1}\sum_{\substack{v_1\in LRMost(v)\\ v_2\in RLMost(v)\\ Level(v_1)=Level(v_2)\\ X(v_1)>X(v_2)-d_{\min}}}(X(v_1)-X(v_2)-EdgeLength)^2$$

即可. 其中 V_1 为树中具有左儿子结点的集合,V_2 为树中具有右儿子结点的集合,V 为树中所有结点的集合,$LRMost(v)$ 为结点 v 的左子树的 h-v 画法中各层结点最右边结点的集合,$RLMost(v)$ 为结点 v 的右子树的 h-v 画法中各层结点最左边结点的集合,$Level(v)$ 为二叉树 h-v 画法中结点 v 所在的层次. 例如,对图2中的 h-v 画法,我们有 $LRMost(1)=\{4,5\}$,$RLMost(1)=\{3,6\}$,$Level(1)=Level(3)=Level(7)=1$,$Level(2)=Level(5)=Level(6)=2$.

当用 f 作为目标函数时,算法所画出的若干图例如图3所示.

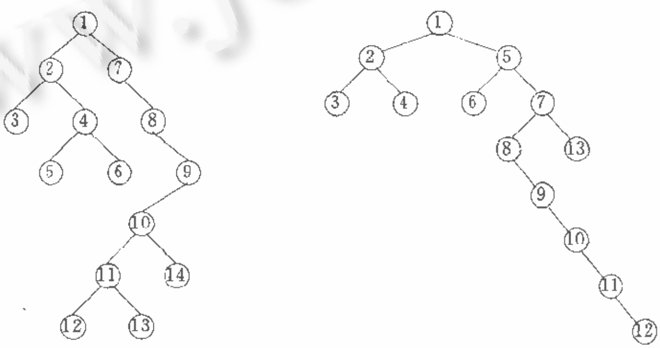

Fig. 3　Experimental results 1

图3　实验结果1

当用 g 作为目标函数时,算法所画出的若干图例如图4所示.

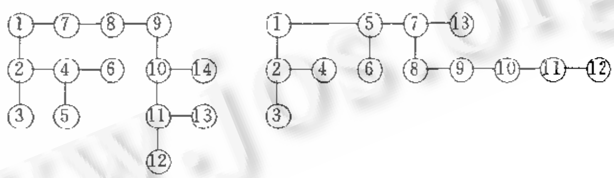

Fig. 4　Experimental results 2

图4　实验结果2

4　结束语

遗传算法是一种简单、鲁棒性强、很有用的自动画图算法设计工具,用遗传算法设计的画树算法具有算法框架统一、容易修改和扩充、适应性强的优点,可以通过设计不同的目标函数来得到不同的画树算法,从而可以提高算法的重用性. 同时还具有自适应、自学习的特点. 对于所要画的树结构,我们不必精确地指定树中结点的坐标,而只需说明怎样画,即指定美观准则并设计出反映美观准则的目标函数即可. 尤其是在有些应用中,当确定性算法不易得到时,用遗传算法设计画图算法则显得更为重要,而且用遗传算法设计的画树算法易于扩展到动态画树算法上. 目前,我们正在从事这方面的研究.

参考文献

1 Battista G D, Eades P, Tamassia R *et al.* Algorithms for drawing graphs: an annotated bibliography. Computational Geometry: Theory and Applications, 1994,4(5):235~282
2 Reingold E M, Tilfold J S. Tidier drawings of trees. IEEE Transactions on Software Engineering, 1981,7(2):223~228
3 Gard A, Goodrich M T, Tamassia R. Area-Optimal upward tree drawings. International Journal of Computational Geometry Applications, 1996,6(3):333~356
4 Crescenzi P, Penna P. Minimum-Area h-v drawings of complete binary trees. In: DiBattista G ed. Proceedings of the Graph Drawing'97. Berlin: Springer-Verlag, 1998. 371~382
5 Michalewicz Z. Genetic Algorithms+Data Structure=Evolution Program. Berlin: Springer-Verlag, 1996

Binary Tree Drawing Algorithm Based on Genetic Algorithms

HUANG Jing-wei KANG Li-shan CHEN Yu-ping

(*State Key Laboratory of Software Engineering Wuhan University Wuhan* 430072)

Abstract In this paper, a new general binary tree drawing algorithm frame is designed by using genetic algorithms. Under the frame, according to different applications, different binary drawing algorithms can be obtained by designing different objective functions reflecting aesthetic criteria. Furthermore, it is easy to add or reduce some of the aesthetic criteria in order to satisfy different users. Compared to previous algorithms, this algorithm is of the following advantages: the frames of the algorithms are unified, the method is simple, implementation and revision are easy. It has the following characters: self-adaptive, self-study and easily parellelized.

Key words Binary tree, tree drawing algorithm, genetic algorithm.