

单木生长模型竞争指标的优化算法

张守攻 唐守正 惠刚盈

摘要 以数据区检索方式为标准, 竞争指标算法可分为4类。其中第二类为改进算法, 竞争木索引扫描和动态分区扫描, 是首次提出的两种算法。用独立的数据区建立索引, 改善了系统的灵活性, 优化算法显著提高了系统的运行速度, 解决了微机上开发应用单木模型系统部分关键问题。同时给出了不同算法结构的内存额外开销量和计算工作量的估计公式。

关键词 单木生长模型、竞争指数、算法设计、计算机模型

经营方案的优化选择和竞争指标的计算, 是单木生长模型时间占用量较大的两个部分。Yoshimoto 等^[1]改进了 PATH 算法^[2], 并将其引入单木生长模型, 压缩了系统运行时的内存和时间占用量。1990年 Yoshimoto 等^[3]又对 PATH 算法作了进一步改进, 改善了多次间伐问题的最优解。竞争指标可分为非完备型和完备型两大类群^[4], 除第一类中的对象木状态描述型指标以外, 计算工作量一般都很大。优化抉择过程要反复多次调用生长模拟过程, 竞争指标算法的优劣将显著影响系统的工作效率。这就是选优算法基本解决之后, 改进竞争指标算法的重要意义。

1 基本概念及符号

1.1 几个基本概念的定义

1.1.1 树木的影响圈半径(Fr) 即树木影响范围的最大半径^[5], 普遍认为这个半径大于树冠的扩张范围。

1.1.2 竞争木扫描半径(Sr) 以对象木为中心, 检索竞争木的最大半径。为了防止遗漏竞争木, 取 $Sr = 2 \times \text{Max}(Fr)$, 即 Fr 最大值的2倍。

1.1.3 边际效应带(bw) 处在样地边缘的树木, 其竞争区域是不完整的。因此, 计算出的竞争指数也存在一定的误差。误差值的大小受树木与样地边界的距离, 以及局部的空间分布等多种因素的影响。模型系统中常用设立边际效应带的办法消除误差, 带内的树木在生长模拟时不作为对象木看待^[6]。边际效应带宽度, 要根据树木的大小, 和竞争指标的定义要求确定, 此处仅作为一个概念引进, 不做更深入的探讨。

1.2 原始数据区的组织形式

为了便于讨论, 假设原始数据区名为 Plot Data, 每株树的信息用一个向量记载, 并将

1993-01-12收稿。

张守攻 副研究员, 惠刚盈(中国林业科学研究院林业研究所 北京 100091); 唐守正(中国林业科学研究院资源信息研究所)。

其中几个元素作如下定义。

t_i : 树木编号, 即外业调查时树木在样地中的实际编号。新型算法和改进算法, 在检索时都用数据的记录号, 以便消除检索模块对外业调查时树木编号方式的依赖。为了能输出与原始数据对应的结果, 在数据中仍保留树号。

x, y : 树木的原始定位坐标值, 统一用矩形样地左下角为零点定位。如用其他形状的样地, 或不同的定位坐标体系, 则应首先进行坐标转换。向量中的其他元素视具体需要而定。

2 竞争指标算法的基本类型和优化原理

由于编程习惯和模型系统数据组织形式的差异, 使算法的表现形式差别很大, 讨论难以展开。本文以反映算法的本质为原则, 把算法归为4类: ①全局扫描检索算法; ②坐标映射局部扫描检索算法; 以及本文提出的两种新算法, ③竞争木索引扫描检索算法; ④通用动态分区扫描检索算法。算法的名称, 是本文根据其特点命名的。在以后的讨论中, 将分别简称为算法1~4。

算法1把整个数据区作为一个整体单元对待。在模拟过程中, 反复运用线性检索^[1]方式扫描整个数据区, 以取得所需数据, 因而计算工作量大。算法2~4运用分区索引原理, 实现了数据区的局部扫描, 加快了系统的运行速度。建立索引的方式多种多样, 本文推荐独立的索引结构。其优点表现在以下3个方面: ①不改变原始数据区, 程序改动少, 开发和扩充方便; ②索引号采用整数, 占用内存少; ③避免了数据类型转换, 提高执行速度。

简而言之, 算法优化的基本原理可以归结为两点: 一是缩小每次检索扫描区域; 二是提高索引区的管理效率。由于专门研究竞争指数算法的文献很少, 所以本文采用由简到繁的顺序, 先介绍现有算法的原理, 以及对算法2的改进。最后, 在此基础上介绍算法3~4的原理及特点。

2.1 全局扫描检索算法(算法1)简介

2.1.1 原理及特点 此算法不增加模型对RAM的额外要求, 适合于各种研究对象, 也不依赖于竞争指标, 适应性广。此算法的检索效率低, 难以在微机上开发实用系统, 对于那些检索量大的竞争指标更是如此。不过该算法可以配合计算量小的竞争指标, 研制小型实验系统, 或用来比较不同算法的效率等。

2.1.2 计算工作量 各种算法都是2级嵌套结构, 所不同的是, 算法1的第2级是全局扫描, 而算法2~4的第2级是局部扫描。算法1的计算工作量(或称时间复杂性)为 n^2 , 算法2~4的计算工作量与其2级扫描记录数有关。

2.2 坐标映射算法(算法2)的原理及改进

Newnham^[2]采用固定株行数的方式(8倍植距), 对数据区进行局部扫描。袁小梅^[1]从3方面对这种算法作了改进, 即: ①采用可变的扫描范围代替固定株行数, ②引进树木实际坐标与逻辑坐标的映射关系, ③创建独立的索引数据区。但文中对算法的介绍和分析缺乏系统性, 原算法使用树木的实际编号进行定位, 数据区的管理效率低。另外, 文中给出的计算工作量估计值得商榷。本文改进了这种算法的结构设计, 并对其进行了系统的分析介绍。

1) 袁小梅. 关于人工林单木生长模型与林分模拟系统的研制. 北京林业大学, 硕士学位论文, 1990.

2.2.1 符号的含义 Row_i : 行间距, y 轴方向; Col_i : 列间距, x 轴方向; tx_i, ty_i : i 对象木的映射坐标。

$$tx_i = \text{Int}(x_i / Col + 0.5) + 1 \quad (1)$$

$$ty_i = \text{Int}(y_i / Row + 0.5) + 1 \quad (2)$$

索引登记区下标从 1 开始, x_i, y_i 为第 i 株对象木的原始坐标。

Δ_i, Δ_j : x 轴和 y 轴方向单侧扫描区间数。

$$\Delta_i = \text{Int}[(Sr/Col) + 0.999] \quad (3)$$

$$\Delta_j = \text{Int}[(Sr/Row) + 0.999] \quad (4)$$

2.2.2 基本原理 为了减少系统对样地中树木编号方式的依赖, 映射变换时使用数据的记录号。此处用一个简化的样地资料, 说明算法 2 的原理。设原始数据区中有 25 个记录, 12 号记录中树木的平面坐标为 (2, 3), 则映射坐标为 $tx_i = 3, ty_i = 3$, 即 $Tree\ Reg(3, 3) = 12$, 余类推。

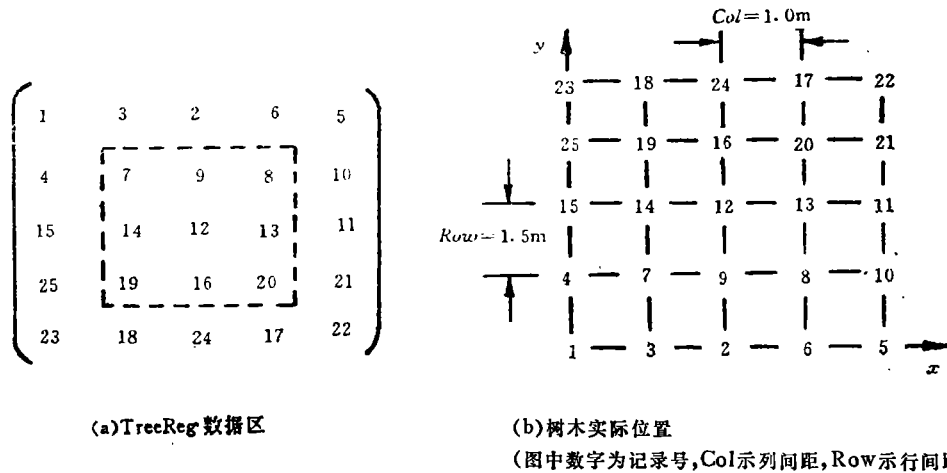


图 1 树木坐标映射变换示意

现假设要找出 12 号树的最近邻体, 采用算法 1, 要检索 25 个记录。而用算法 2, 仅需扫描图 1 (a) 虚线框中的几个记录。对本例而言, 其检索量仅为算法 1 的 1/3 左右。若检索扫描半径 Sr 已知, 索引登记区的扫描域为: $I_i \in [tx_i - \Delta_i, tx_i + \Delta_i]$, $J_j \in [ty_j - \Delta_j, ty_j + \Delta_j]$ 。

由以上分析可知, 算法 2 仅适用于个体空间分布呈规则格局的研究对象。例如统一一株行距且呈矩形配置的人工林, 而且要求其植苗定位的误差小于 1/2 株行距。

2.2.3 内存额外开销 映射坐标与树木的定位坐标是一一对应的, 因此, 索引区占用的最小内存量, 为记录项数(样本数)乘以整数的字长。

$$Extra\ Ram = N \cdot 2 \text{ (bytes)} \quad (\text{以 16 位机为例, 下同}) \quad (5)$$

索引数据区 $Tree\ Reg$ 是大小为 $(Num\ Rows, Num\ Cols)$ 的 2 维整数数组, 其中 $Num\ Rows$ 和 $Num\ Cols$ 分别表示样地中树木的行数和列数。

2.2.4 计算工作量 此算法的计算工作量, 可以用检索记录数的变化来表示。又由于算法 2 仅适用于规则格局的特殊性, 还可以用样地面积和检索区域的关系表示。在规则格局条件下, 样地面积与检索面积之比, 等于总株数与检索株数之比, 即 $N/Sn = L \cdot W / 4Sr^2$ 。计算工作量

$$Imp\ Time = n^2 \cdot 4Sr^2 / (L \cdot W) \quad (6)$$

式中： N 为总株数， S_n 检索株数， L 样地长， W 样地宽， S_r 搜索半径。

2.3 竞争木索引扫描检索算法(算法3)的结构设计及特点

如果仅把符合条件的竞争木记入索引区，计算时就无需进行再检索，从而最大限度地提高索引区的利用率，达到减小计算工作量的目的，这就是算法3的设计思想。

2.3.1 基本原理 算法3的原始数据区与索引数据区保持行相容。在登记过程中，把与对象木*i*发生竞争树木的记录号，写入第*i*索引向量内。若根据图1(b)的空间位置，进行最近4株竞争木索引，*Tree Reg*数据区内7号记录对象木的索引向量为(9, 14, 4, 3)，余类推。在生长动态模拟时，只要按*i*索引向量的指示，调出竞争木的数据，便可快速求出*i*对象木的竞争指数值。

由此算法的数据组织特点可看出，它最适用于竞争木株数固定不变的竞争指标的计算，而对林分的实际空间格局无依赖。

2.3.2 内存额外开销 索引区的记录数与数据区的记录数相同，其记录长度等于每株对象木的竞争木数量，那么，本算法的额外内存开销量

$$Extra\ Ram = N \cdot C_n \cdot 2 \text{ (bytes)} \tag{7}$$

C_n 表示与任一对象木发生竞争的竞争木数。索引数据区*Tree Reg*的维数由(N, C_n)确定。

2.3.3 计算工作量 索引区的记录内没有冗余项，检索效率高，计算工作量的估计也非常简单，等于总株数乘以竞争木株数

$$Imp\ Time = n^2 \cdot C_n / n = n \cdot C_n \tag{8}$$

2.4 通用动态分区扫描检索算法(算法4)的结构设计及特点分析

通过以上讨论不难理解，算法1~3在应用中受到很多限制。算法1的速度太慢，算法2对研究对象的空间格局类型，样本点的空间配置方式有特殊的要求，算法3受竞争指标的影响大。下面将要介绍的算法4，突破了以上几方面的限制，具有广泛的适用性。

2.4.1 基本原理 m ：样地边长的分段数(两条边相同)；*Sect Num*：小区的编号，按行编排。 L_m, W_m 分别表示小区有效区域的横边和纵边长

$$L_m = (L - 2bw) / m \tag{9}$$

$$W_m = (W - 2bw) / m \tag{10}$$

Sect Rows, Sect Cols 分别代表小区的行和列。

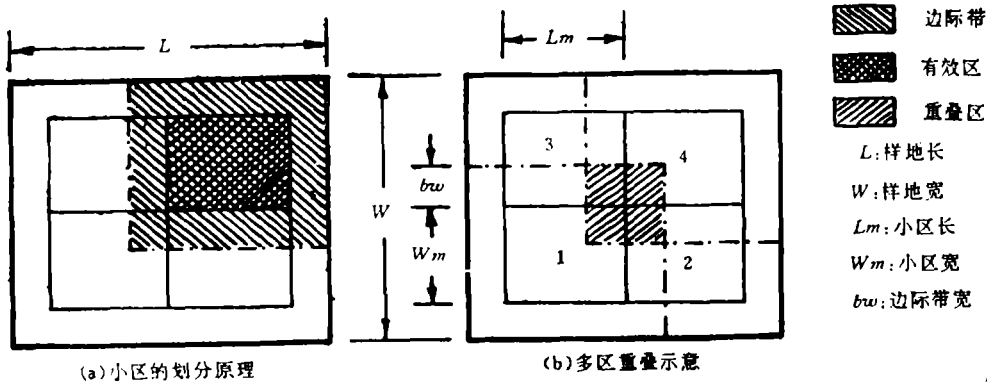


图2 动态分区索引原理示意

本算法是把原始数据划分成少数几个区, 按区进行索引登记。每小区内树木数据的记录号用一个索引向量记载。小区面积等于有效面积与边际效应带面积之和, 如图 2(a) 所示。边际效应带内的树木, 在本小区不作对象木看待。由于小区边际效应带的加入, 索引区有部分重叠。如图 2(b) 阴影区内的树木, 在 4 个区内都要作索引登记。因此, 在索引登记时要逐区检验, 而检索时则可一次定到区。索引登记阶段把 (Sx_1, Sy_1, Sx_2, Sy_2) 看成活动框标, 用每个区号为它们计算赋值。

当 $Sx_1 \leq x_i \leq Sx_2$, 且 $Sy_1 \leq y_i \leq Sy_2$ 时, i 号记录写入本区索引。

Sx_1, Sy_1, Sx_2, Sy_2 的赋值分为两种情况: 当 $SectNum \bmod m = 0$ 时, $SectCols = m$, $SectRows = Int(SectNum/m)$; 否则, $SectCols = SectNum \bmod m$, $SectRows = Int(SectNum/m) + 1$ 。

$$Sx_1 = (SectCols - 1) \cdot Lm \quad (11)$$

$$Sx_2 = SectCols \cdot Lm + 2bw \quad (12)$$

$$Sy_1 = (SectRows - 1) \cdot Wm \quad (13)$$

$$Sy_2 = SectRows \cdot Wm + 2bw \quad (14)$$

检索时区号的计算

$$SectNum = Int[(y_i - bw)/Wm] \cdot m + Int[(x_i - bw)/Lm] + 1 \quad (15)$$

2.4.2 内存额外开销 样地数据不分区, 或分成若干不带边际效应带的区, 其索引项之和都为 N , 内存的额外开销量为 $N \cdot 2$ 。为了提高模拟的精度, 分解后的小区都应加上各自的边际效应带。由图 2 可知, bw 越宽, 重叠部分越多, 内存额外开销越大。

分区后, 每小区的平均个体数为 $[n(Lm + 2bw) \cdot (Wm + 2bw)] / (L \cdot W)$ 。总共有 $m \cdot m$ 个小区, 分区后的记录总数与原始数据项之比, 即为系统资源开销系数

$$O(m) = [L - 2bw(1 - m)][W - 2bw(1 - m)] / (L \cdot W) \quad (16)$$

不分区时, $O(1) = 1$ 。一般情况下, 总有 $bw > 0$, $m > 1$ 成立, 由式(16)知, bw 越宽, $O(m)$ 的值越大, m 的影响与之相同。特别, 如取 $bw = 0$, 则无论分区数为多少, $O(m)$ 总为 1。额外内存开销

$$ExtraRam = O(m) \cdot N \cdot 2 \text{ (bytes)} \quad (17)$$

2.4.3 计算工作量 模拟样地的大小、样地内的株数、分区数等, 都不同程度地影响计算工作量。为简单起见, 假定树木在样地内的分布是均匀的。没有分区前, 每检索一株竞争木要扫描 n 个记录, 分成 $m \cdot m$ 个区后, 扫描的记录数为 $n(Lm + 2bw)(Wm + 2bw) / (L \cdot W) + m$ 。其中 $(Lm + 2bw) \cdot (Wm + 2bw)$ 为一个小区的面积; m 是样地边长的分段数, 此处正好表示找到待检树木所在区所需的平均判别次数。则计算工作量

$$ImpTime(m) = n^2 \cdot (Lm + 2bw)(Wm + 2bw) / (L \cdot W) + m \cdot n \quad (18)$$

对于特定的模拟对象, (18) 式中只有 m 是可变的。因此, 工作量函数是 m 的一元函数, 由 $ImpTime(m)$ 可以求得最佳分区数。经变量代换(18)式可写成

$$ImpTime(m) = n(am^3 + bm^2 + cm + d) / (a \cdot m^2) \quad (19)$$

其中: $a = L \cdot W / n$, $b = (2bw)^2$, $c = 2bw[(L - 2bw) + (W - 2bw)]$,
 $d = L \cdot W - 2bw(L + W - 2bw)$ 。

$$\text{令: } dImpTime/dm = n(am^3 - cm - 2d) / (a \cdot m^2) = 0 \quad (20)$$

即, $am^3 - cm - 2d = 0$ (21)

因为分区数不能为小数, 所以解(21)式时, m 取最接近于根的整数。将 m 代入(18)式, 便可求得分区索引条件下的计算工作量。

3 结 论

由以上分析可知, 各种算法在其适应性、内存占有量和计算工作量三者之间都存在一定的矛盾。因此, 只能在了解每种算法特点的基础上, 根据研究对象的具体情况, 选择最佳的计算方法。表1是几种算法的特点汇总, 表2是不同条件下各种算法的比较结果。

可见, 在微机上开发单木生长模型不宜选用算法1。就检索速度来说, 算法3最快, 并且适用于不同的研究对象。不过要根据竞争指标修改索引模块, 除非研究对象和竞争指数均已固定, 否则不提倡选择算法3。从总体效果考虑, 以人工林为研究对象的场合应选算法2; 以天然林、非规则格局的人工林为研究对象, 或进行通用软件系统开发, 应选择算法4。

表1 各种算法特点及适用范围

计算方法	RAM 额外开销	计算工作量	适用范围
算法 1	0	n^2	适用于各种场合, 但其速度慢, 除一些计算量小的竞争指标外, 微型机上一般不宜采用此算法
算法 2	$n \cdot 2$	$n^2 \cdot 4Sr^2 / (L \cdot W)$	仅适用于呈矩形配置的人工林, 对竞争指标无限制, 运算速度快
算法 3	$n \cdot Cn \cdot 2$	$n \cdot Cn$	适用于各种场合, 速度最快, 其主要缺点是, 数据组织模块依赖于竞争指标, 程序改动大, 如有间伐操作, 必须重新进行索引
算法 4	$n \cdot 2 \cdot O(m)$	(18)式	适用于各种场合, 速度较快

表2 各种算法的选择指南

对象	可选算法	速度优先	适应性	RAM开销	综合效果
规则格局	1, 2, 3, 4	3, 2, 4, 1	(1, 2, 4), 3	1, 2, 3, 4	2
任意格局	1, 3, 4	3, 4, 1	(1, 4), 3	1, 3, 4	4

注: 括号内数据的优先级相同。

参 考 文 献

- 1 Yoshimoto A, Paredes V, Brodie JD. Efficient optimization of an individual tree growth model. California, USA. USDA For. Serv. Gen. Tech. Rep., 1988, RM-161, 154~162.
- 2 Gonzalo L, Paredes V, Brodie JD. Efficient specification and solution of the evenaged rotation and thinning problem. For. Sci., 1987, 33(1), 14~29.
- 3 Yoshimoto A, Haight RG, Brodie JD. A comparison of pattern search algorithm and the modified PATH algorithm for optimizing an individual tree model. For. Sci., 1990, 36(2), 394~412.

- 4 关玉秀, 张守攻. 竞争指标的分类及评价. 北京林业大学学报, 1992, 14(4):18.
- 5 Krajicek JE, Brinkman KA, Gingrich SF. Crown competition — a measure of density. For. Sci., 1961, 7(1):35~42.
- 6 Pukkala T. Effect of spatial distribution of trees on the volume increment of a young Scots pine stand. Silva Fennica, 1989, 22(1):1~17.
- 7 Thesen A. (薛华成译). 运筹学的计算机方法. 北京: 清华大学出版社, 1986.189.
- 8 Newnham RM. The development of a stand model for Douglas-fir. Ph. D. Thesis. Fac. For., Univ. B. C., Vancouver, Canada, 1964. 201.

An Improvement to Competition Index Calculating Algorithm for an Individual Tree Model

Zhang Shougong Tang Shouzheng Hui Gangying

Abstract The algorithms of estimating competition index was divided into two groups according to its competition tree searching regimen, and one of them was improved. Then, two new searching algorithms were created, which operate well for stand with various spatial patterns. Introducing an independent index system makes a computer model restructured easily. With new and improved algorithm, a system can get a stand growth simulation done in a dramatically shorter time than it used to. One of the staggering problems, unendurable of implementing time, which had been an obstacle to the application of individual tree model, was overcome. The equations were provided to access extra RAM demand and calculate the implementing time.

Key words individual tree model, competition index, algorithm design, computer model

Zhang Shougong, Associate Professor, Hui Gangying (The Research Institute of Forestry, CAF Beijing 100091), Tang Shouzheng (The Research Institute of Forest Resources Information Technique, CAF).