

# 包装产品协同设计多版本增量基分布优化研究

刘峰, 纪钢

(重庆理工大学, 重庆 400054)

**摘要:** 为了提高包装产品协同设计过程中历史多版本的查询效率, 针对完整版本和增量结合的树形结构多版本演化模型, 设计并实现了一种以版本间距离程度为度量依据的中间完整基版本选择算法。实验表明, 通过该算法选择的增量还原基版本, 使多版本还原效率进一步提高; 同时, 该算法有使用简单、无复杂控制参数、版本树数据冗余易于控制等特点。

**关键词:** 包装产品协同设计; 版本树; 版本间距离; 基版本

**中图分类号:** TB482; TP302; TP391 **文献标识码:** A **文章编号:** 1001-3563(2011)01-0022-03

## Research on Base Distribution Optimization of Multi Versions Increment in Packaging Products Collaborative Design

LIU Feng, JI Gang

(Chongqing University of Technology, Chongqing 400054, China)

**Abstract:** In order to improve the efficiency of versions inquiring in packaging product collaborative design process, an algorithm for selecting base version by distances between related versions, was designed and implemented. This algorithm is of versions tree structure. Experiments showed that the version distribution model determined by the algorithm has better performance. The efficiency of regenerating all the versions is improved. The algorithm without complex control parameters is simple to use, and the data redundancy of versions is easy to control.

**Key words:** packaging product computer support collaborative design; versions tree; distance between versions; base version

计算机辅助协同设计应用于包装产品的结构和外观<sup>[1-2]</sup>设计, 可以让分布在网络上的多个设计者充分展现、交流各自的设计理念, 有利于设计出更好的包装产品。协同设计要求保存设计过程中重要的设计历史, 因为最终的设计方案通常是在对多个历史设计进行分析、比较、优化、综合后完成的, 多版本的形成是必然的。

多版本管理是协同设计的重要内容, 目前相关研究主要集中在多版本的产生与演化、版本复制及合并<sup>[3-4]</sup>、版本存储<sup>[5]</sup>与数据一致性<sup>[6]</sup>保证等方面。完整存储和增量存储是多版本存储的 2 种方式, 基于多版本结构考虑, 进行完整和增量存储方式的分布优化, 可以提高协同多版本的管理性能, 而与此相关的可见的研究文献较少。针对树形多版本结构, 文献

[7]提出了一种多版本分布优化策略, 但存在如下局限性: 未考虑增量版本之间的差异性, 认为版本间增量变化是均衡的, 而实际情况并非如此; 结点深度阈值作为该策略的一个评判参数, 其值的选择对结果影响较大, 文中用估值的方式确定其值, 需要经验知识, 导致算法使用复杂。

笔者拟对文献[7]的研究进行改进, 设计一种新的中间完整基版本选择确定算法, 并通过实验来验证算法计算结果的优越性与有效性。

## 1 中间完整基版本选择算法设计

### 1.1 问题描述

协同设计过程的多个版本之间有继承性, 相邻版

收稿日期: 2010-10-15

作者简介: 刘峰(1970-), 男, 重庆人, 硕士, 重庆理工大学教师, 主要研究方向为计算机辅助设计。

本间的变化差比起完整版本的信息量要小得多,如将设计过程中的每个版本都完整保存,必然存在大量的数据冗余。将除初始版本  $v_0$  外的其他版本都保存为增量版本,节省了存储空间,但要获取某版本,必须从  $v_0$  开始,遍历完所有的前向版本,逐级进行差值计算,效率低;如果其中一个版本发生破坏或丢失,其后序版本均不可恢复,数据可靠性差。版本树中,一些结点采用完整存储,其余结点采用增量存储,可兼顾存储空间和查询效率两方面性能,是合理的多版本存储模式。

为后续分析和表述方便,做以下定义:

定义 1。版本树  $T=(V, E)$ , 其中  $V$  是树结点集合,代表全部多版本。 $E$  为结点之间的连接边,代表版本之间有继承关系。每条边有对应的边权值  $\omega$ , 表示后版本可在前版本的基础上经过  $\omega$  步操作得到。

定义 2。基版本,即版本树中完整保存的版本,表示为  $v_i = \cup(O_j^i, \text{value}(O_j^i)), j \in (1, 2, \dots, \text{nobj})$ , 其中  $O_j^i$  表示版本  $v_i$  中的第  $j$  个图形对象,  $\text{value}(O_j^i)$  是对象  $O_j^i$  对应的状态值,  $\text{nobj}$  为  $v_i$  中图形对象的数量。根结点初始基版本外的其他基版本,称为中间基版本。

定义 3。增量版本,即以相邻 2 个版本之间的操作变化差异的形式记录的版本,可表示为  $v_j^* = \cup(\text{Sid}, \text{Oid}, \text{Otype}, \text{Attr}, \text{Valold}, \text{Valnew})_j, j \in (1, 2, \dots, \omega)$ 。其中 Sid 为协同站点标识, Oid 是被操作图形对象标识, Otype 是操作类型, Attr 是图形对象的被操作属性, Valold 是 Attr 属性修改前的值, Valnew 是 Attr 属性修改后的值,  $\omega$  是相邻版本间差异操作步数。

定义 4。多版本分布模式  $f(T)$ 。版本树的全部版本结点集合,可划分为完整版本集合  $B$  和增量版本集合  $I, B+I=T$ ; 版本树中,哪些版本属于  $B$ , 哪些版本属于  $I$ , 这种划分方式即为多版本的分布模式。

完整版本和增量版本的分布影响着版本树的性能。完整版本设置越多,数据冗余变大,但可以提高增量版本的还原效率。研究目的就是要确定一个较优的版本分布模式  $f(T)$ , 节省版本存储空间,提高版本查询效率。

## 1.2 中间基版本选择算法思想

版本树中,版本结点的子结点数,边权值  $\omega$  的大小是影响中间基版本选择的重要因素。如某版本  $v$  存在  $p$  个子版本,  $v$  是在前基版本上经过  $q$  步操作

得到,如将  $v$  设为基版本,可节省恢复  $v$  的所有子版本需进行的  $p \cdot q$  步操作计算量。那些前向边权值  $\omega$  越大,子结点数越多的版本结点,越是应该设置成为基版本。

用前向边权值  $\omega$  和后序子结点个数的积作为指标,来计算版本间距离。版本树中,那些与其他版本距离越远的版本,越是应该被选择为基础版本,因为与其他版本的距离越远,把它作为基础版本,增量还原计算步数减少得就越多,多版本还原就越快。距离计算方式为:

版本树中,记版本  $v_i$  沿树根方向,追溯到根的全部权值(增量)之和为  $\omega t$ , 版本  $v_i$  的所有后序派生子结点个数记为  $k$ 。

$$\text{指标 } x = \omega t \cdot (k + 1) \quad (1)$$

式(1)表示如以  $v_i$  为基版本,可节省的增量恢复操作步数。

设论域  $v = \{v_1, \dots, v_m\}$  为分析的  $m$  个版本对象,对其中的每一个版本  $v_i$ , 用(1)式计算的属性指标记为  $x_i$ , 为了判定版本间的距离程度,先计算各版本对象两两之间的距离  $d_{ij}$ , 并构成版本树结点集合  $v$  的距离矩阵  $D$ 。

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & & d_{2m} \\ \vdots & & & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mm} \end{bmatrix} \quad (2)$$

$$\text{其中 } d_{ij} = \sqrt{\frac{1}{m}(x_i - x_j)^2} \quad (3)$$

$$\text{令 } Ld_i = \sum_{j=1}^m d_{ij} \quad (4)$$

$Ld_i$  是距离矩阵  $D$  中第  $i$  行之和,其值大小反映了第  $i$  个版本  $v_i$  与版本树中其他版本对象的距离。

## 1.3 中间基版本选择算法描述

以版本间距离为评判依据,协同设计多版本增量基选择算法流程为:以初始基版本  $v_0$  为基础,计算用(1)式表示的每个版本的指标,再按(2)–(4)式计算每个版本和其他版本的距离,结果中最大距离对应的版本为所求的第 1 个基版本。再以已求得的新的基版本集合为基础,重新计算指标和距离,再判断选出另一基版本,重复以上过程,直到找到指定个数的基版本为止。

## 2 验证实验及结论

### 2.1 验证实验

为了比较文献[7]方法和本文算法进行版本树优化的效果,设计了如下实验。实验中,采用基于 AutoCAD 环境二次开发实现的协同图形编辑系统,进行包装盒的模拟协同设计,增量版本的还原使用 AutoLisp 语言编程实现。通过在本地操作产生后随机延时几秒再发送给其他协作用户,来增加并发操作机率,并按意愿保证策略导致版本复制,截取了其中的 16 个版本,形成的实例版本树<sup>[8]</sup>见图 1。

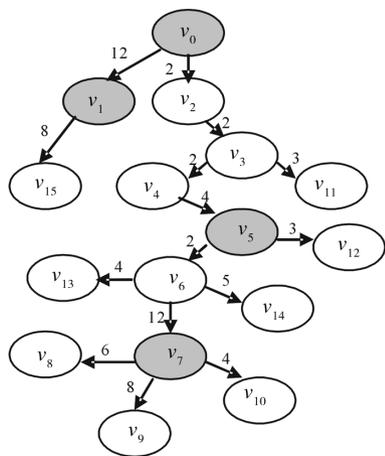


图 1 版本树实例

Fig. 1 Example of versions tree

使用文献[7]方法,深度阈值取 3,计算得到 3 个中间基版本。按本文算法,也以计算出 3 个中间基版本为目标,首先以  $v_0$  为基版本,计算指标  $x_i$ ,进行第 1 次迭代,结果中  $Ld_7 = 2.4609$ ,在结果中最大,故取  $v_7$  为第 1 个基版本。以  $v_0, v_7$  为基版本,重新计算指标  $x_i$ ,进行第 2 次迭代,结果中  $Ld_5 = 2.5650$ ,其值最大,故取  $v_5$  为第 2 个基版本。重复以上过程,进行第 3 次迭代后,  $Ld_1 = 2.7812$  最大,取  $v_1$  为第 3 个基版本。

针对实例版本树,2 种方案的优化结果比较见表 1。

表 1 2 方案优化结果比较

Tab. 1 Comparison of optimization result using two strategies

方案	文献[7]方法	本文算法
结果基版本	$v_3, v_5, v_7$	$v_7, v_5, v_1$
增量还原总步数	75	63

在相同数据冗余(有 3 个中间基版本按完整方式保存)的情况下,采用本文算法,计算得到的基版本和文献[7]相比,要恢复出版本树中的所有版本,可减少 12 步增量操作,节省近 1/6 的耗时。

### 2.2 算法有效性验证

为了验证算法的有效性,采用和图 1 算例相同的实验方法,随机生成了多个版本树。分别使用文献[7]方法和本文的算法来对随机生成的版本树,进行中间完整基版本的分布优化。

以生成相同个数的中间基版本为前提(保证相同的数据冗余),具体方法是先使用文献[7]的方法计算,以文献[7]方法得到的结果基版本个数作为本文算法的输入(待求基版本个数),再使用本文算法来选择确定每个版本树的系列中间基版本。

对每个版本树的 2 种方案的优化结果,进行多版本还原,并记录了实测的版本还原时间  $t$ ,绘制的对比图见图 2。

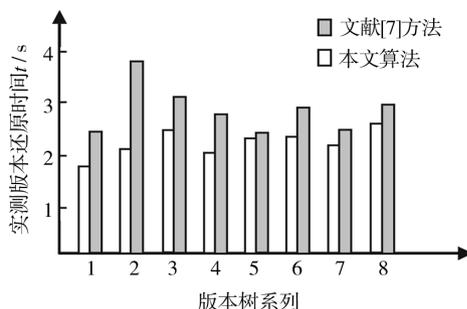


图 2 多版本树还原时间两方案对比测试结果

Fig. 2 Test results of times for regenerating versions

从图 2 可以看出,在基版本个数相同的情况下,对每一个版本树,使用本文算法得到的多版本存储分布模式,和使用文献[7]方法所得结果相比,实测的版本还原时间都有缩短。

## 3 结语

基于以上算法改进及实验结果和分析,以包装产品计算机辅助协同设计中的多版本性能优化为研究目标,能解决实际问题。按算法优化结果进行多版本的存储管理,在尽可能节省版本存储空间的情况下,提高了多版本的查询效率,增强了协同设计过程的实时性。

相对结晶度可以定量反映被测物质结晶程度的大小。通过结晶度的计算可知<sup>[8]</sup>,淀粉相对结晶度为 40.12%,改性淀粉的相对结晶度为 39.82%,说明氧化和交联改性主要发生在淀粉的无定形区,对微晶区的结构几乎没有影响<sup>[9]</sup>。

## 2.4 扫描电镜分析

通过扫描电镜分析,可以了解微观结构与宏观性能之间的联系<sup>[10]</sup>。淀粉改性前后的扫描电镜照片见图 3。

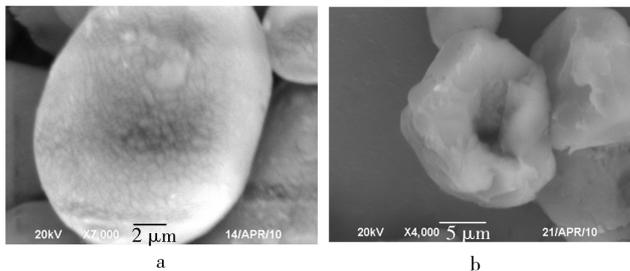


图 3 淀粉改性前后的 SEM 图

Fig. 3 SEM images of starch before and after modification

从图 3 可知,淀粉改性前呈规则的椭圆状颗粒,表面光滑,颗粒与颗粒之间相互独立;氧化交联改性后,淀粉颗粒的表面发生了扭曲变形,表面有“絮状”漂浮,但仍保持了整体形貌不变,颗粒与颗粒之间部分通过“絮状”的穿插相互连接。通过对图 2 的分析可知,MF 的存在,可使淀粉的内部的微结晶区向表面渗透,形成“絮状”结构。MF 交联改性的是淀粉的表面,通过改性微结晶区,形成网状结构,提高耐水性能。

## 3 结论

1) MF 改性淀粉胶粘剂、过硫酸铵氧化与酸解淀

粉,可降低胶粘剂粘度,提高稳定性;三聚氰胺甲醛树脂交联淀粉,可形成交联网状结构,提高耐水性。

2) MF 改性的淀粉胶粘剂,可作为瓦楞纸粘合剂使用,符合瓦楞纸粘合剂使用国家标准。

3) MF 改性淀粉,提高了淀粉的结晶度,使淀粉塑化,增加了淀粉的粘结强度。

## 参考文献:

- [1] 张玉龙,王化银. 淀粉胶黏剂[M]. 第 2 版. 北京:北京化学工业出版社,2008.
- [2] 骆光林,王茜. 淀粉粘合剂性能改善研究[J]. 上海包装, 2006(12):38.
- [3] 孙伟圣. 氨基树脂改性淀粉胶粘剂合成,结构与性能的研究[D]. 广州:华南农业大学,2006.
- [4] GB/T 6543-2008,运输包装用单瓦楞纸箱和双瓦楞纸箱[S].
- [5] LOIS E, SCHEYER M P. Use of Chemically Modified Wheat Gluten to Reduce Formaldehyde Emissions During Curing of Pigment Print Pastes on Fabrics [J]. Starch/Stärke, 2000, 52: 420-422.
- [6] IMAM S H, MAO L, CHEN L. Environmentally Friendly Wood Adhesive from a Renewable Plant Polymer: Characteristics and Optimization[J]. Polymer Degradation and Stability, 2001, 73: 529-533.
- [7] TOR S N, GREGORY R Z. Structural Features of Non-granular Spherulitic Maize Starch[J]. Carbohydrate Research, 2002, 337: 1467-1475.
- [8] 张本山,张友全,杨连生,等. 淀粉多晶体系结晶度测定方法研究[J]. 华南理工大学学报(自然科学版), 2001, 29(5): 55-58.
- [9] 马骁飞,于九皋. 尿素和甲酰胺塑化热塑性淀粉[J]. 高分子学报, 2004, 4(8): 483-489.
- [10] 张美珍,柳百坚,谷小昱. 聚合物研究方法[M]. 北京:中国轻工业出版社,2007.

(上接第 24 页)

## 参考文献:

- [1] 杨德龙,崔杜武. 分布式同步协同包装纸盒 CAD/CAM 系统的设计与实现[J]. 计算机辅助设计与图形学学报, 2005, 17(9): 2138-2142.
- [2] 康晓兵,魏生民,卫延斌. 基于 SVG 和 X3D 的网络化协同产品包装设计模式[J]. 包装工程, 2006, 27(2): 128-132.
- [3] 张雪平,殷国富,王伟,等. 基于网络技术的实时协同设计并发控制策略研究[J]. 四川大学学报(工程科学版), 2005, 37(5): 144-147.

- [4] 侯守明,潘亚锋,沈志广. 基于网络的实时协同设计技术研究及实现[J]. 机械设计与制造, 2009(8): 246-248.
- [5] 邵伟峰,杨洋. 协同编辑系统中多版本 XML 文档的存储与查询[J]. 计算机工程, 2006, 32(24): 75-77.
- [6] 窦万峰,朱鸣,陈卉. 实时协同编辑系统中可靠性和容错研究[J]. 微电子学与计算机, 2007, 24(9): 95-97.
- [7] 付喜梅. 基于 STEP 的协同设计版本存储控制策略[J]. 计算机工程, 2008, 34(24): 61-63.
- [8] 迟建,李晓娟. 包装纸盒设计系统——Visual Basic. NET 二次开发 AutoCAD[J]. 包装工程, 2005, 26(6): 96-98.