

# 基于多层次通用件的改进遍历及计算方法

方霞<sup>1</sup>, 潘梅森<sup>1</sup>, 王喜富<sup>2</sup>

(1. 湖南文理学院 计算机学院, 湖南 常德 415000;

2. 北京交通大学 交通运输学院, 北京 100044)

**摘要:**探讨了产品物料清单(BOM)的多层结构、层次码结构和复合结构存储形式,及常用BOM递归遍历算法和层次遍历算法。针对实际应用中多层次通用件计算的重复和遗漏问题,基于低层码LLC的思想,改进了BOM遍历算法。通过设计一个临时队列存储父子关系和层次码,使得该算法不受递归的限制,随着产品结构复杂程度增加更具有优越性。应用该算法对多层次通用件进行对应父项的分类统计,实现了物料总数量的计算,针对BOM网状结构提出了有效解决方案。

**关键词:**物料清单;遍历算法;低层码;多层次通用件

**中图分类号:**TP391

**文献标识码:**A

**文章编号:**1673-629X(2009)06-0102-04

## An Improved Traverse Algorithm Based on Multi-level Generic Pieces and Its Application of Calculation

FANG Xia<sup>1</sup>, PAN Mei-sen<sup>1</sup>, WANG Xi-fu<sup>2</sup>

(1. Dept. of Computer Science and Technology, Hunan University of Arts and Science, Changde 415000, China;

2. School of Traffic & Transportation, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** As the multi-level structure, code-level structure and composite structure of the BOM's storage forms, discussed the commonly used recursive traversal algorithms and the level traverse algorithms of BOM. To resolve the problem of duplication and omission which is associated with the multi-level generic pieces, based on the idea of lower-level code as LLC, an improved traverse algorithm of the BOM is put forward. By designing a temporary storage queue to restore the relationship of father and son and the level code, the new algorithm has not the restrictions of recursive algorithms. And it has more advantages with the increased complexity of the product structure. In application by calculating the multi-level generic pieces' classification of statistics, this algorithm has successfully achieved the total number of materials. As the network structure of BOM, this algorithm made effective solutions.

**Key words:** bill of material; traverse algorithm; lower level code; multi-level generic pieces

## 0 引言

ERP系统中物料清单BOM(Bill Of Material)与企业各部门的业务活动紧密联系,其变化与维护对生产产生最直接的影响,调度部门根据产品BOM计算后的结果来安排生产,库房根据BOM安排入库,采购部门根据BOM进行原材料采购等。BOM的数据结构及遍历算法计算是ERP系统数据模型的核心<sup>[1,2]</sup>。BOM中存在着同一种物料项目同时出现在产品不同层次的现象,这种项目称为多层次通用件。一个多层次通用件可能出现在同一产品的不同层次上,也可能

出现在不同产品的不同层次上。多层次通用件涉及不同层次上的总需求量计算,有时出现重复或遗漏计算问题。针对该情况,文献[3]阐述了产品结构树的这种多对多关系,以及网状结构与层次结构的转换,并采用二叉树形式的产品结构树实现了BOM操作,解决了树形BOM存储的大量数据冗余问题。文献[4]针对BOM结构中出现的多对多关系利用邻接表建立联系,构造相关表单结构,对于一般网状数据关系的BOM有通用性,但BOM展开不够灵活。针对多层次通用件计算的重复和遗漏问题,有效的解决办法就是引入层次码和最低层代码叫低层码(Lower Level Code, LLC)<sup>[5]</sup>。文献[6]设计了低层码算法,采用一个临时堆栈表,使得BOM遍历算法对于比较复杂的产品结构有优越性,但存储空间冗余度较大。文献[7,8]研究了BOM遍历算法和BOM总数量计算等关键技术,针

收稿日期:2008-09-17;修回日期:2008-12-12

基金项目:湖南省自然科学基金资助项目(07JJ6141)

作者简介:方霞(1972-),女,湖南常德人,硕士,讲师,研究方向为计算机应用、智能控制与优化;王喜富,教授,博士生导师,研究方向为系统工程、物流信息技术。

对多层次通用件的有关问题没有得到很好的解决,文中基于 LLC 进行改进,实现 BOM 遍历算法及物料准确计算,特别对于重复和遗漏问题提供详细的解决方案。

1 产品结构树和 BOM 表的存储结构

实际应用中,不同产品采用不同的存储结构来实现<sup>[9]</sup>。由于多层次通用件的存在和经常使用,产品的结构是一个典型的网状结构。从物理装配和组成关系的角度来分析,产品与其构成的部件、部件与其子部件具有层次关系。由于图形结构的复杂性在目前关系型数据库中无法对其进行有效便捷的操作,而由产品的特征,对图形结构进行适当修改,将被多次指向的节点(半成品)进行多处复制,图结构就变成了树的层次结构。实例参考图 1 产品 A 结构转换,同时添加上相应用量。

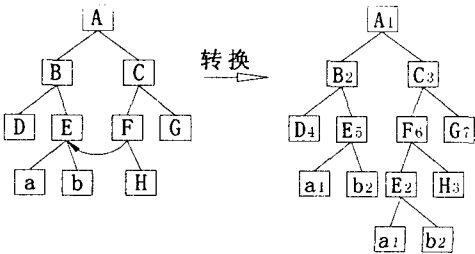


图 1 产品 A 网状结构向树结构的转换

1.1 多层 BOM 结构分析

多层 BOM 结构采用“单父多子”数据结构,同样的零部件,只要存在于不同的产品中,也需要再记录一次。多层 BOM 结构的特点是产品间结构相互不影响,各个产品之间的数据记录没有交叉,因此维护方便。但需要递归程序来显示其结构,大大降低了系统的运行效率。

1.2 按层次码排序的 BOM 结构

按层次码排列 BOM 结构的冗余度较大,层次码中隐含了零部件的所在层次和低层码,非常容易显示产品的结构树和进行 BOM 结构分解,也可以快速准确地反查一个零部件的归属情况。但是随着产品结构复杂程度的增加,层次码的空间占用较大。

1.3 改进的 BOM 结构

综合 BOM 多层结构和层次码结构的优缺点,根据文献[3],采用一种复合式 BOM 结构,来缓解系统实际运行过程中数据维护和运行效率之间的矛盾。创建的 BOM 数据表结构如表 1 所示,结合产品 A 其内容如表 2 所示。

复合式 BOM 结构与多层结构相似,主要提供了层次字段,该字段初始为空,利用 BOM 结构生成其层

次码,然后获得低层码 LLC(Lower Level Code)。所谓低层码是指某个物料在所有产品结构树中所处的最低层次码,每个物料有且仅有一个低层码。如图 1 产品 A 的结构树中,C 分别处于产品结构树的第 1 层和第 3 层,于是 C 的低层码为 3,如果还有其他零部件使用 C,且 C 的层次码  $n$  大于 3,那么 C 的低层码更改为  $n$ 。产品 A 的对应表 2 中 L 表示叶子结点,为 BOM 展开算法所用。通过 LLC 可以控制 ERP 计划的编制顺序,使得各物料的 ERP 运算不重不漏,提高系统的效率。

2 BOM 遍历算法

ERP 系统包含一些 BOM 的基本操作,如展开查询、多级查询、以及工艺文件的制定,还有物料需求计划的反查、BOM 物料数量的计算等,都需要在 BOM 结构表中检索数据。为了确保相关计算的顺利和有效进行,要进行 BOM 的合法性检查和计算属性生成,也需要对 BOM 树进行遍历,而且该类操作使用频繁,因此如何快速查到目标物料,所用到的 BOM 遍历算法非常重要。常用的遍历算法采用深度优先的递归算法,或者采用层次遍历算法<sup>[10]</sup>。

表 1 物料清单数据表结构

序号	字段	类型	长度	是否为空	缺省值	说明
1	father	varchar	30	Y		父项代码
2	Son	varchar	30	N		子项代码
3	Dosage	float	10	Y		用量
4	Level	Int	10	Y	0	层次号
5	Llc	Int		Y	0	低层码
6	...	...	...	...	...	其他字段信息

表 2 产品 A 的复合式 BOM 表主要信息

记录号	父项代码	子项代码	用量	层次号	低层码
1	A	B	2	1	1
2	A	C	3	1	1
3	B	D	4	2(L)	2
4	B	E	5	2	3
5	C	F	6	2	2
6	C	G	7	2(L)	2
7	E	a	1	3(L)	4
8	E	b	2	3(L)	4
9	F	E	2	3	3
10	F	H	3	3(L)	3
11	E	a	1	4(L)	4
12	E	b	2	4(L)	4

2.1 常用的递归算法

针对图 1 所示产品 A 的 BOM 结构,递归查找的搜索序列是 A(1), B(2), D(8), E(10), a(10), b(20), C(3), F(18), E(36), a(36), b(72), H(54), G(21)。递归算法采用树的先根遍历,较好地展现了物

料层次之间的层次关系和 BOM 树的结构,优点是程序结构简单,但是使用递归,程序执行每一步都需要将临时节点信息压入堆栈,当系统结构复杂度加大时,系统资源占用较大,系统的运行效率明显降低。

## 2.2 常用的层次遍历算法描述

针对图 1 所示 A 的 BOM 结构,递归查找的搜索序列是 A(1), B(2), C(3), D(8), E(10), F(18), G(21), a(10), b(20), E(36), H(54), a(36), b(72)。层次算法实质是非递归算法,利用临时表(队列结构)存放中间信息,不需使用内存堆栈,运行速度快,能较好地实现大规模 BOM 树结构展开。但是由于层次查找的特点,不能很好地展现物料之间的父子关系,在物料结构的取代、删除以及检查嵌套错误等方面需要知道物料之间的父子关系,利用层次搜索将出现操作上的困难,无法展示产品 BOM 的完整树形型结构<sup>[3]</sup>。

## 2.3 改进的基于 LLC 的 BOM 遍历算法

提出的改进算法针对初始形成的 BOM 结构表,设计了层次号、低层码等域,初始都置空。利用数据库中索引的相关概念,设计一个临时表 TempQ 在计算层次码时作为临时队列使用,表中主要有三个字段(father, son, level),分别存储父子关系和层次码。改进算法基于层次遍历的思想,通过一些改进得到遍历序列,并同时计算各种物料的层次码。

### 算法 1 伪代码描述:

```

创建临时数据 TempQ(father, son, level);
将要查找的物料展开第一层物料信息放入临时表 TempQ 中;
置该层物料的 level=1,且修改主表 P_bom 中低层码字段 llc(初始为 0)为 level;
While 表 TempQ 中数据不为空
    Begin
        取出第一条物料信息;
        更新主表中对应物料的 level 值;
        删除临时表中当前物料记录;
        把当前结点的所有子结点放入临时表 TempQ;
        如果有子结点则 level 增值,并更新主表中对应记录的 llc 为新的 level;
    End;
在主表 P_bom 中针对相同编号的物料进行低层码处理;

```

相较于常用递归算法和层次算法,该改进算法需要的临时空间小,而且主要环节只需要一重循环就能实现。通过 LLC 的使用,及时实现了主表数据的反填更新。然后实现层次码的统一,可以有效保证其完整性和数据的一致性。该算法效率较高,可以解决较大规模的产品结构计算问题。

## 2.4 基于 LLC 计算物料总数量

在最低层次码已经计算完毕的基础上将进行物料

的准确计算,该关键步骤在新产品提交、更改产品提交以及 ERP 运算前的 BOM 合法性检查等功能模块中都具有重要作用。针对多层次通用件,重复和遗漏问题是需要计算中重点考虑的。算法 1 的本质是层次遍历,由于低层码的使用,可以将父子关系控制问题进行改进。

### 算法 2:实现物料数量计算的伪代码:

步骤(1)定位于第一条记录(根结点),置其总数量为 1,变量  $N = 1$  记录最低层次码;

步骤(2)将最低层次码 =  $N$  的相关记录的 BOM 编号顺序存入分段数组;

步骤(3) 如果对应层次  $N$  段数组不为空,顺序对数组中的 BOM 编号进行下列操作;否则转步骤(4)。

1) 查找相应的父项总数量。

2) 计算本项物料的总数量:本项物料的总数量 = 用量(父子相对用量字段信息) \* 父项总数量。

3) 如果  $N$  段数组的所有数据计算完毕,则  $N = N + 1$ ,转步骤(2);否则转步骤(3)。

步骤(4) 针对编号相同的物料进行统计汇总,总数量计算完成。

算法 2 的说明:在实际计算中对于同一物料项目同时出现在产品不同层次的多层次通用件计算有一定的难度,需要进行特别处理。由于算法 1 本质上是基于层次遍历的一种改进,基于低层码的特征,在算法 2 中计算结果父件和子件都在相邻层次,可以体现上层项目与下层项目之间的父子关系,利用数组的顺序处理过程,可以实现准确计算。在寻找父项总数量时,统计某物料的父项个数  $m$ ,如果  $m = 1$  则直接进行计算;否则  $m > 1$  表示存在多层重用问题,需要根据  $m$  值进行分组,子项分别寻找自己的对应父项进行数量统计。

如何寻找对应父项是解决多层次通用件重复和遗漏计算问题的核心,通过设计一个标志临时表,分别存放多层次通用件的父项、第一个子项、分组值和当前组等信息,其结构如图 2 所示。通过对该表的组别和当前值进行依次搜索,可以有效实现对应父项的查找和分类统计。该方法考虑的多层次通用件重复和遗漏问题周全,不仅解决了文献[7,8]中提到的多层次通用件的计算问题,针对其未能涉及到的情况也能有效解决。

父项	第一个子项	分组值	当前组	.....
----	-------	-----	-----	-------

图 2 标志临时表结构

## 3 实例数据

在 ERP 功能模块的开发中,结合 BOM 的展开计算功能对上述算法进行了应用。运行环境操作系统是

WinXP,数据库采用 SQL Server2000。

### 3.1 算法性能分析

理论上几种遍历算法将所有结点均访问了一遍,时间复杂度  $T(n) = O(n)$ 。

改进算法1和常用算法相比,其空间复杂度很明显。常用递归算法需要内存堆栈,辅助内存为产品结构树的深度  $k$ ,最坏情况可达到  $O(n)$ 。当BOM的复杂度增加,特别是随着产品结构树复杂度的增加,会造成系统崩溃。常用非递归算法需要使用临时数据表,外存空间最多为  $O(n)$ 。

改进算法的辅助空间里采用了数据库的创建索引思想,充分利用主表信息,减少信息冗余,临时表空间最大是  $O(3n/m)$ ,  $m$  表示 BOM 数据主表的字段数。

### 3.2 算法实现

采用表2所示的记录结构作为BOM的数据表模式,基于SQL Server2000,这里给出其算法1的核心简单实现部分:

```
CREATE PROCEDURE Expand(@TopItem varchar(30)) as
SET NOCOUNT ON
DECLARE @level int,@father varchar(30),@son varchar(30)
CREATE TABLE #TempQ(father varchar(30),son varchar(30),
level int)
INSERT INTO #TempQ SELECT father,son,1 FROM P_bom
WHERE father=@TopItem
UPDATE P_bom SET llc=1 WHERE father=@TopItem
WHILE EXISTS(SELECT * FROM #TempQ)
BEGIN
SELECT @father=father,@son=son,@level=level FROM #
TempQ
UPDATE P_bom SET level=@level
WHERE father=@father AND son=@son AND llc<>0
//将求得的层次号填入 P_bom 主表,利用 llc 先做临时标志位
DELETE FROM #TempQ WHERE father=@father AND son=
@son AND level=@level //删除临时表中的当前结点
INSERT INTO #TempQ SELECT father,son,@level+1 FROM
P_bom WHERE father=@son //将其子结点放入临时表中,并
计算其层次码
IF @@ROWCOUNT>0
UPDATE P_bom SET level=@level+1 WHERE father=@son
END
```

算法2的物料数量计算准确,较好地解决了层次分析法中父子关系不能较好对应的问题。利用一个分段数组实现了低层码的物料计算,针对多层次通用件的计算问题,考虑到层次中展现父子关系不很清晰,设计使用一个标志临时表实现查找对应父项,应用效果好。

实际应用中单个产品实验数据一般不超过3万条记录,层次在15层以内。在同一台机器上运行比较得到改进算法在实际的应用中取得了明显的效果,极大地提高了系统的效率,减少了数据库系统资源的占用,提高了数据库的性能和响应能力。

## 4 结束语

(1)通过分析比较三种典型的BOM结构模型,综合多层结构和层次结构的优缺点,决定采用复合式BOM结构。

(2)通过总结BOM核心计算相关算法的实现,提出一种新的基于低层码的改进算法,基于SQL Server2000,给出了算法的实现。充分利用存储过程以及相关字段设置标志,简化了算法的复杂度。

(3)对于多层次通用件的计算求解,结合层次遍历中父子关系不够明确的问题提出了较好的解决方案,利用分段数组和标志临时表得以实现。该方法在物料的取代、删除等方面实现还有待进一步探讨。

### 参考文献:

- [1] 汪洋,张宇.基于STEP标准的统一BOM模型研究[J].计算机技术与发展,2006,16(10):185-188.
- [2] Steveson M, Hendry L C, Kingsman B G. A review of production planning and control: the applicability of key concepts to the make-to-order industry[J]. International Journal of Production Research, 2005, 43(5): 869-898.
- [3] 梁书峰,郭顺生,熊志勇.使用递归算法的N级结构的BOM的设计[J].计算机应用,2003,23(12):274-294.
- [4] 张军,万兴宇,徐明秋.一种多对多BOM结构的实现[J].机械设计与制造工程,1999,28(6):51-53.
- [5] 程控,革杨. MRPII/ERP原理与应用[M].第2版.北京:清华大学出版社,2006.
- [6] 余锐林,吴顺祥.一种改进的BOM展开及低层码生成算法[J].计算机工程与应用,2005(27):100-102.
- [7] 杜杰,陆金桂.一种改进的多级性BOM遍历算法[J].工程设计CAD与智能建筑,2002(9):65-67.
- [8] 张玲,董天阳,曹玉华,等.ERP中BOM结构设计及其关键技术研究[J].浙江科技学院学报,2005,17(4):268-272.
- [9] Ji GuoLi, Gong DaXin, Tsui F. A Tree Structure Storage Model of BOM[J]. Journal of Systems Science and systems engineering, 2002, 11(1): 55-60.
- [10] Clement J, Coldrick A, Sari J. Manufacturing data structures, building foundations for excellence with bills of materials and process information[M]. New York, NY, USA: Oliver Wight Publications, Inc., 1992.