

Research on Improvement of Structure Optimization of Cross-type BOM and Related Traversal Algorithm

XiuLin Sui¹, Yan Teng, XinLing Zhao and YongQiu Chen

School of Mechanical and Power Engineering Harbin University of Science and Technology, Harbin, China

¹*suixiulin@sina.com*

Abstract

Based on an analysis of the advantages and disadvantages of single-layer and multi-layer BOM structure, and combine with specific examples of products, a more practical—cross-type BOM structure method is presented. Aiming to solve the system efficiency problems brought by the BOM operation in manufacturing. Regarding the queue as a temporary table, we put forward an improved multi-layer BOM traversal algorithm, makes full use of the basic operation characteristics and setting marks in related fields queue and simplifies the complexity of the algorithm. Compared with the traditional recursive algorithm and hierarchical algorithm, the improved algorithm has the advantages of fast speed, low resource consumption, and high practicability. And the improved algorithm is verified through an example.

Keywords: *Cross-type BOM, Improved algorithm, Performance Analysis*

1. Introduction

Bill of material (Bill of Material, BOM) is the core basic data in manufacturing information system [1]. It plays an important role in product design, manufacturing, sales and after sale service and it reflects the information and sharing in enterprise data integration [2]. Although the structure of SBOM and MBOM are widely used due to the advantages of recursive procedure is clear, easy to read the program and easy to maintain. BOM involves a large variety of materials, large quantity, which make the decomposition and inverse checking of products very difficult [3]. Simple SBOM and MBOM structure is very difficult to meet the actual needs.

We consider the disadvantages of the SBOM structure increases the difficulty of ensuring the accuracy and with low efficiency and the disadvantages of MBOM with the big redundancy of the data items and reduplicate definition of component structure and is unable to display the tree structure of products clearly. We present the cross-type BOM structure which can define the product structure clearly and improve the system arithmetic speed and also with superior performance in all respects. Based on this, through the studies of recursive algorithm and the algorithm of hierarchical traversal and improved the previous traversal strategy, we get a optimization algorithm which with a high running efficiency, finally contrasts the arithmetic speed between the traditional algorithm and improved algorithm through a specific instance, then verify and analyze the algorithm.

2. The BOM Tree Structure of Products

In production, the structure of the product appears as a tree structure that represents the ownership relations and quantities of parts. From Figure 1, we can see that product A is composed of component B, component C, component D and component E (B, C, D and E is called A's child respectively), the former is the parent of the latter, and the component B is composed of component C and part F (B is C and F's parent). In Figure 1, the number on each box represents the quantity relationship between parent and child, this structure is called the bill of material of product A.

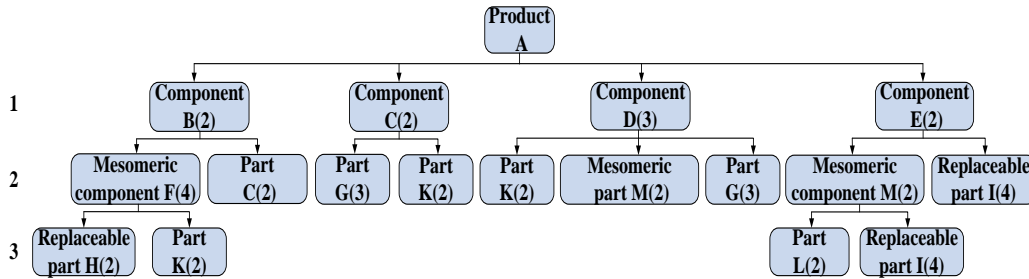


Figure 1. BOM Structure Diagram of Product A

3. The Optimization of the BOM Structure

3.1. The Tradition BOM

There are advantages and disadvantages during the maintenance process of SBOM. The advantages are that it has good logical relations and good adaptability; The disadvantages are that it will increase the difficulty of ensuring the accuracy, at the same time recursive algorithm will greatly reduce the operation efficiency of the system[5]. In the structure of MBOM, for the same parts, as long as they exist in different products, still need to be recorded once again [6]. And the big redundancy of the data items and reduplicate definition of component structure will increase the workload of entering basic data when the system is started and is unable to display the tree structure of products clearly.

3.2. The Optimization of Cross-type BOM Structure

Considering the disadvantages of single-BOM and multi-BOM and the problems of various BOM structure running at the same time, we integrate two BOM construction methods, then presents a method of constructing cross-type BOM, in order to solve the conflicting issues of the system during the actual operation data maintenance and operational efficiency. The adoption of this cross-type BOM structure can not only improve the system efficiency but also define the product structure clearly.

In the actual production, the products are often more complex, parts perplexing, often need to use each other. Therefore, the single-BOM and multi-BOM are all difficult to fully represent the information of product structure, especially the complex products shows the cross status. For example, the composition of product A structure in Figure 1 is more complex, it is difficult to express by a common single BOM or multi-BOM. However, a cross-type BOM can display it very clearly, which is represented in Figure 2.

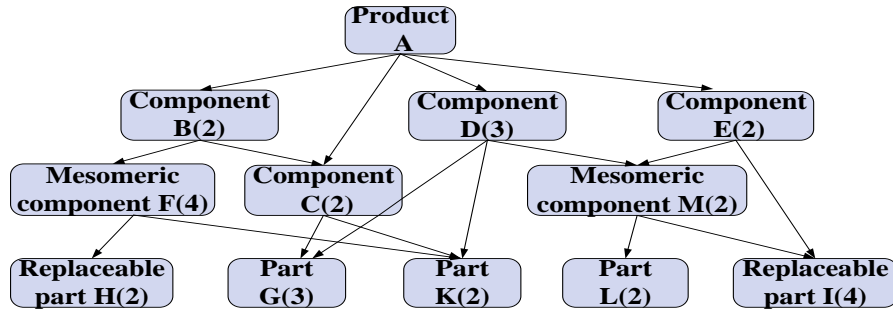


Figure 2. The Structure of Cross-style BOM

As can be seen from Figure 2, product A consists of 2 components B, 2 components C, 3 components D, 2 components E, component C is also the constituent part of component B. Each component B consists of 4 mesomeric components F and 2 components C; Each component C consists of 3 parts G and 2 parts K; Each component D consists of 2 mesomeric components M, 2 parts K and 3 parts G; Each component E consists of 2 mesomeric components M and 4 replaceable parts I; Each mesomeric component F consists of 2 parts K and 2 replaceable parts H; Each mesomeric component M consists of 2 parts L and 4 replaceable parts, in order to express and inverse checking conveniently, we introduce mesomeric component " F and M. Table 1 is the 3-level table of product A.

Cross-type BOM structure has an advantage in expressing the structure of more complex products. Meanwhile, cross-type BOM structure can be applied in larger fields compared with single-BOM, multi-BOM and composite BOM, especially for complex structure BOM, cross-type BOM has more advantages. Cross-type BOM reduces data redundancy, in the meantime, it accelerates the BOM query speed, and is convenient to modify and analyze the data, which also improves the efficiency of the system.

Table 1. 3-level Table of Product A

Number of Layers	Parent	Child	Number
1	A	B	2
		C	2
		D	3
		E	2
2	B	F	4
		C	2
	C	G	3
		K	2
	D	K	2
		M	2
E	G	3	
	M	2	
3	F	I	4
		H	2
	M	K	2
		I	4

4. The Optimization of BOM Traversal Algorithm

4.1. Recursive Algorithm

BOM traversal recursive algorithm adopts first-root traversal method, and fully demonstrates the hierarchical relation between the material and BOM tree structure, according to the recursive condition of whether the BOM tree materials have next layer sub-materials. BOM tree structure is shown in Figure 1, the traversal order of product A is:

$A \rightarrow B \rightarrow F \rightarrow H \rightarrow K \rightarrow C \rightarrow C \rightarrow G \rightarrow K \rightarrow D \rightarrow K \rightarrow M \rightarrow G \rightarrow E \rightarrow M \rightarrow L \rightarrow I \rightarrow I$

This recursive algorithm start from the root node to search down optionally a child, and select a sub to list exhaustively until reach the leaf node. Because every step of the program execution needs to preserve the stack, system memory is largely consumed, the operating efficiency of the system is very low for a material level-BOM structure. In practical application, the design of the general high-level language program must be set for the stack size, when the stack used by traversal procedure is greater than the set value, it will lead to the fail or breakdown of the system.

4.2. The Algorithm of Hierarchical Traversal

The principle of BOM hierarchical traversal algorithm is to start from the root materials to find the needed BOM, layer by layer down to traverse until you find the material in the bottom layer, and in the end merger the same code material from each layers to get the needed BOM [7].

BOM tree structure is shown in Figure 1, the traversal order of product A is $A \rightarrow B$ 、 C 、 D 、 $E \rightarrow F$ 、 C 、 G 、 K 、 M 、 G 、 $I \rightarrow H$ 、 K 、 L 、 I . Hierarchical traversal will search out all the node sub items, which greatly improves the search speed, it can traverse a massive BOM tree structure according to the hierarchy. Due to the hierarchical search features, the storage order of materials fails to show the relationship between parent and child, while in the replacement and deletion of actual material structure, we must know the parent-child relationship between materials, so hierarchical BOM traversal algorithm cannot meet this requirement [8].

4.3. The Improvements of Traversal Strategy

4.3.1. The Targets of Improved Algorithm: The goal of improved algorithm is not only like recursive algorithm can show the parent-child relationship, quantitative relations between the material and the BOM tree whole structure to check nested error of material structure operation conveniently, but also like hierarchical traversal can find out all the child of the node in once search process, greatly improves the searching speed; and can overcome the contradictions that storage space size of the two algorithms cannot be confirmed [9].

The Targets of Improved Algorithm: In this algorithm, the queue is used as a temporary table. It is a first-in-first-out linear table, inserted in the end of the table, deleted in the head of the table. The queue operation is shown in Figure 3 for the tree structure of the BOM in Figure 1. An empty queue queue is initialized, front is the head pointer, and rear is the tail pointer, first put A into the team, meanwhile, according to the record information of A to find

B, C, D, E, which are the components of A, put B, C, D, E into the queue. Hierarchical number of A plus 1. And output the related information of A. A is out, the queue is not empty, then the sub member F, C of B enter into the queue. After B is out, check the component H, K of F. Circulating like this until the queue is empty, and this traversal ends. During the algorithm runs sequentially dequeue sequence is A B C D E F C G K K M G M I H K L I.

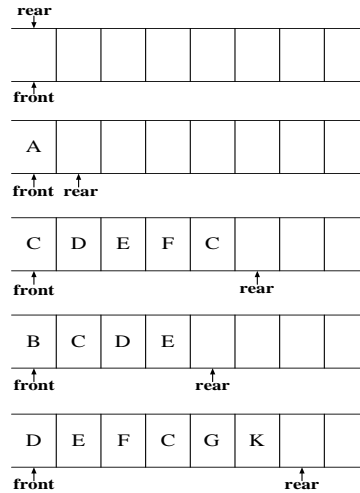


Figure 3. Order Storage and operation of Queue

According to the description of the algorithm, the process of the algorithm is shown in Figure 4.

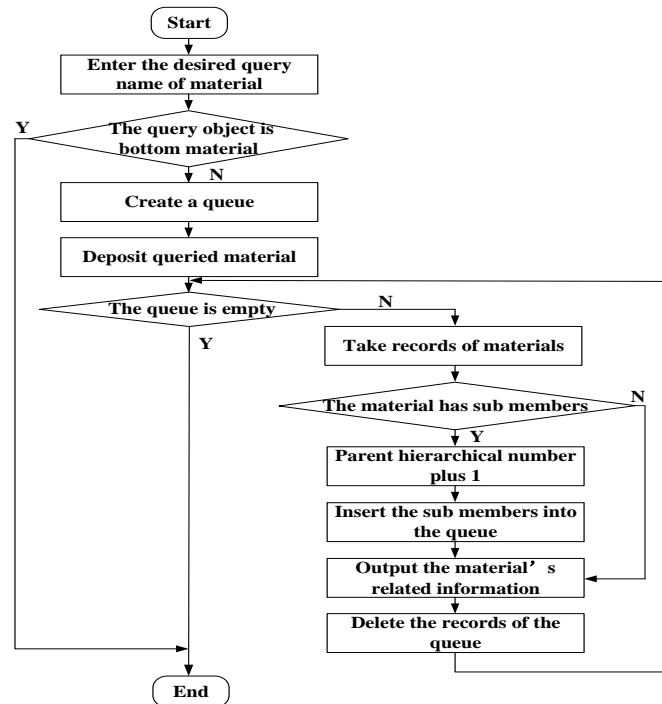


Figure 4. Flow Chart of Improved Algorithm

Improved traversal algorithm can better show the hierarchical relationship of parent and child between the material and BOM tree structure, the using of queue operations makes the program simple, and it also inherits the advantage that hierarchical traversal can find out all the children of the node in once search process, greatly increases the searching speed. It can also better traverse a massive BOM tree structure according to hierarchy, and can solve the disadvantages of the storage space size cannot be defined, which the two algorithms are inevitable.

4.3.3. The Targets of Improved Algorithm: The running speed is improved obviously in the aspect of time complexity [10]. In order to analysis product structure conveniently, assume that the product structure tree is similar to full binary tree, r is defined the quantity of tree forks, k is the depth of tree. So by unwrapping recursive algorithm of BOM, we can gain the recurrence formula: $T(1)=1, T(k)=r \cdot T(k-1)+1$. Then after it is extended and calculated, the formula changes into $T(k)= (r^k -1)/(r-1)$. By the above formula, we can make two conclusions: ① Time complexity is $T(k)=O(r^k)$, it shows an exponential growth tendency with the increase of the tree's depth. ② In actual application, the quantity of r is not equal in itself material decomposition, the product structure tree is not ideally full- r forks tree. So must be cautious to choose the suitable quantity of material branches r and control the whole product tree's depth k . What's more, the opening up of memory stack needs the operation system to distribute time, but this work is transparent to user. In actually, stack size is closely related to tree's depth k .

Comparing the traditional algorithm with improved algorithm in the aspect of space complexity, the traditional algorithm needs memory stack, generally spatial assistant item is $O(\log n)$. When the BOM complexity increases, especially material quantity n increases, it will leads to the system collapse; The traditionally non-recursive algorithm needs to use temporary data table, generally is $O(n)$. The auxiliary space of improved algorithm uses queue as temporary table, and makes full use of the main table information, reducing the redundancy information. The maximum space of a temporary table is $O(3n/m)$, m represents the fieldcount of BOM data main table.

5. Instance Verification

We contrast the application effect of traditional algorithm and improved algorithm through the instance. The operating environment of the ERP system of a car manufacturer is WIN2000 operating system, and database uses the MS SQL Server2000, the algorithm application modules is BOM sequential check, inverse check and the calculation of preventing the nested error and the low-level. In the aspect of operating time and the amount of processing data, the comparison of traditional algorithm and improved algorithm is shown in Figure 5. Table 2 expresses the algorithm processing capability comparison of material expands.

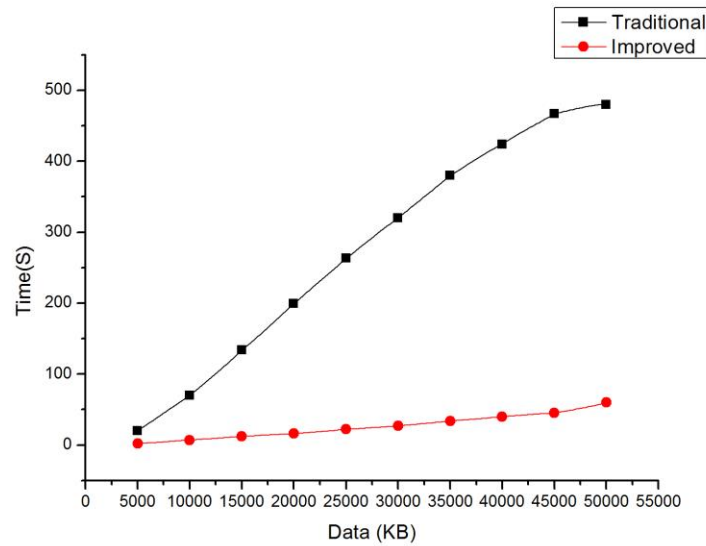


Figure 5. The Comparison of Traditional Algorithm and Improved Algorithm

Table 2. The Algorithm Processing Capability Comparison of Material Expand

Data Recording	Processing Capability	
	Traditional Algorithm	Improved Algorithm
5×10^4	Y	Y
10×10^4	Y	Y
15×10^4	N	Y
25×10^4	N	Y
40×10^4	N	Y

In different computers, different performance and configuration may lead to the results have a little different, but from the operation on the same machine, the obvious effects of improved algorithm in actual application can be seen. It improves the efficiency of the system greatly, reduces the occupation of software for large-scale BOM traversal time and database system resources, improves the database performance and response ability greatly.

6. Conclusions

This paper through analyzing and comparing the structure characteristic of MBOM and SBOM, presents a method of constructing cross-type BOM, overcomes the disadvantages of the large data redundancy of MBOM, unclear product structure definition and low decomposition efficiency of SBOM. Therefore, cross-type BOM is a practical BOM constructing method. Based on analyzing the data structure of product, this algorithm not only like recursive algorithm can show the parent-child relationship, quantitative relations between the material and the whole structure of BOM tree to check nested error operation of material structure conveniently, but also like hierarchical traversal can find out all the child of the node in once search process, greatly improves the searching speed; and can overcome the contradictions of storage space size of the two algorithms cannot be confirmed.

Acknowledgements

This research was sponsored by Excellent Academic Leaders Project of Harbin science and technology innovation talents of special fund (2013RFXXJ064) and Heilongjiang province education department natural science and technology item (11531036).

References

- [1] F. H. Zeng and L. H. Zhou, "BOM configuration based on genetic algorithm", vol. 156-157, (2011), pp. 529-533.
- [2] W. M. Yan and W. M. Wu, "Data Structure", Qinghua University Press, Beijing (1997).
- [3] J. C. Zhang, F. J. Kang, H. X. Wu and W. Huang, "BOM ontology-based composite modeling approach for simulation model", vol. 323, no. PART 1, (2012), pp. 37-45.
- [4] G. Q. Feng, D. L. Cui, C. G. Wang and J. P. Yu, "Integrated data management in complex product collaborative design", Computers in Industry, vol. 60, no. 1, (2009), pp. 48-63.
- [5] A. P. Li, F. Fen and L. Y. Xu, "A Sort of Improved Circular Algorithmic Implement of BOM Expandedness", Mechanical and Electrical Integration, vol. 6, (2007), pp. 30-42.
- [6] M. Imran, A. Rassul, V. Vladimir and M. Farshad, "Fairness Verification of BOM-based Composed Models Using Petri Nets", (2010).
- [7] Y. Zang and Z. Y. Cai, "Research on Structure Design and Algorithm of BOM for ERP System", Machine Tool and Hydraulics, vol. 36 (2008), pp. 22-25.
- [8] X. Fang and M. L. Pan, "An Improved BOM Expanded Algorithm Based on the Lower Level Code", Development and Innovation of Machinery and Electrical Products, vol. 23, (2010), pp. 100-103.
- [9] S. Zlatko, O. Tihomir and L. Alen, "In search of an improved BOM and MRP algorithm", ITI, (2009), pp. 665-670.
- [10] T. Yasemin, T. Isa, A. Sevgi, D. Oguz and B. Ismail, "Using BOM in conceptual modelling of distributed simulation projects", vol. 2, (2007), pp. 1595-1613.