On Sequential-Pattern-Based Intelligent Compilation of Aircraft Assembly Process

OUYANG Sen-shan¹, SONG Liang²*, LIU Xiang-feng¹, CHEN Chen¹, CHEN Zhen-yu²

¹. Chengdu Aircraft Industrial (Group) Co., Ltd, Chengdu, China, 610073;
². Sichuan Energy Internet Research Institute, Tsinghua University, Chengdu, China, 610042.

*Correspondence should be addressed to Song Liang; songliang@tsinghua-eiri.org

Abstract. The aviation industry which aircraft manufacturing dominates is an important sector to measure a nation's air power. The application of new technologies to optimize the process, increase the speed and upgrade the quality of airplane assembly, a crucial link in its manufacturing, is an effective method to improve the capacity. The combination of traditional processes and emerging technologies greatly improves its efficiency, reduces its labor costs and promotes the innovation and upgrade of traditional technologies. For aircraft manufacturers, the Assembly Outline (AO) refers to an essential document used for the guidance of the assembly process. Low efficiency and under-structuration of compilation, which is also difficult to upgrade, put a limit on the application of information and digitization technologies and has an impact on the overall airplane production cycle, due to present manual work in quantity and imbalance of informationization. Via sequential-pattern-based data mining introduced in this paper, patterns of AO processes are dug up to form a set based on which it is possible to quickly make a recommendation of the optimal process for the compiler, effectively reducing the duration of compilation and modification and improving the efficiency.

1. Introduction

With the rapid development of economic globalization and the boost in military high-tech, the world geopolitical layout is being quietly changed, and the competition among nations more and more fierce. Military strength, especially that of the air force, is a standard to weigh whether a nation is strong or not. Over the past 60 years since the founding of the PLA Air Force, its strategy has been evolving from territorial air defense to the capability of both offensive and defensive operations, and the development and production of the fourth-generation aircraft have been successfully achieved by PLA Air Force. However, a big gap still exists when compared with the United States. The increase in both capacity and output of aircraft production is an effective measure to empower PLA Air Force and to narrow the existing strategic gap.

Strict appearance requirements, frequent changes in designing, multiple product configurations, large number of components, compact internal space and complex internal structure characterize the modern aircraft. Therefore, the assembly, a difficult but significant multi-disciplinary process in aircraft manufacturing, has a direct effect on its development and production cycles.

In regard to aircraft assembly, Boeing’s Assembly Outline Management Mechanism [1] was introduced by most manufacturers in China to adapt to the characteristics that modern airplanes are usually manufactured in small batches separately. The assembly outline, with records on assembly
processes and specific modes of implementation, is consisted of administrative and descriptive pages, sketch maps and inventories of matching parts, including serial numbers, indexes, modification forms, issue numbers, vehicle controls, revision controls, customer controls, process numbers, brief descriptions, operation requirements and other contents. The outline is a manual for assembly workers, which they must strictly follow in their production process. With an efficient improvement in airplane manufacturing, this mechanism relatively requires more for the outline as well, the more rapid and accurate which is compiled, the faster aircrafts are assembled and the shorter their production cycle is. With the development of information technology, the informatization of the compilation is also on the way, from hand labor to computer work, which has led to the upgrade of structuration, readability and production speed of the documents, and finally a better guide to the process. However, the input of 3D data sets during the compilation mainly relies on technologists in a manual way, therefore the following problems still exist:

1. Repetitive compilation. Different assembly outlines may have same processes. However, most compilation modes are counting on manual inputs rather than selections, leading to much repetitive hand labor and much one-by-one modification while changes occur, which is inefficient and error-prone.

2. Different assembly outlines have same process combinations, which, relatively speaking, are generally solidified. Manpower-based compilation easily leads to different ways within the same combination due to the compiler’s personal habits that makes it difficult to form a unified process.

As a result, for the purposes of solving the above, discovering same processes or their combinations in different assembly outlines, optimizing and speeding up compilation process, and better helping the improvement of airplane manufacturing, the sequential pattern algorithm proposed in this paper takes all outlines as a sequence from which combinations of processes are dug up and standard operations of each process based on them are bound up, to form a knowledge library of assembly outlines which makes recommendations of processes and modes during compilation to reduce the errors caused by manual input, shorten the cycle, improve the efficiency, accumulate the experience and promote the industrial revolution.

2. The Sequential pattern

The rapid progress of data acquisition and storage technology boosts the data in different sectors. The use of data mining technology to explore values from massive data sets has become a hot topic in the data field. Frequently-used algorithms in data mining include the supervised, and the unsupervised which contains the clustering, the association and so on. Traditional association rules algorithms were first proposed by Agrawal R, Imielinski T, and Swami A, [2] to discover interesting connections between items in shopping-basket data transactions, and later used in bioinformatics, medical diagnostics, web mining, and scientific data analysis. The concepts of support and confidence are defined in algorithms of association rules.

If X and Y are disjoint item sets, the expressions of support and confidence rules are described as follows:

\[ s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \]
\[ c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \]

\( \sigma(X) \), showing the count of X support, is the number of times X appears in all transactions. Support is an important measure which is usually used to remove meaningless rules. The measure of confidence is to infer its reliability through rules. Rules that are both larger than the thresholds of support and confidence are usually chosen as association rules, and the concept definition of them lays a foundation for the popularization of association rule algorithms.

Association rule algorithms emphasize the simultaneousness of occurrence rather than its time sequence which may be of great value [3] in identifying the re-occurrence of dynamic systems and predicting the future occurrence of specific events. As sequential information of transaction sequences
with time series cannot be mined through association rule algorithms, sequential pattern algorithms, defining the concept of sequential pattern and putting forward the AprioriAll, AprioriSome and DynamicSome algorithms, were proposed by Agrawal and Srikant in their document [4]. The sequential pattern algorithm, which matters in data mining, figures out that sequential transaction data problems cannot be solved through association rule algorithms, and has been widely applied in customer behavior analysis, computer and wireless remote sensing network, natural disaster prediction and so on. The sequential pattern refers to a given set of different sequences among which each one is lined in order with different elements (transactions) that is composed of different items and is given a minimum support threshold specified by one user. The purpose of sequential-pattern-based mining is to find all frequent subsequences whose frequency of occurrence in the sequence set is no less than user-specified minimum support threshold.

GSP algorithm [5], also proposed by Agrawal and Srikant upon the above mentioned, is an extension of AprioriAll, which, with the introduction of time constraint, sliding time window and classified hierarchy technologies and the application of hash trees to store candidate sequences, reduces the number of sequences to be scanned. Both MFS [6] put forward by Hang M, Kao B and Yip C and PSP by Masseglia, Cathala and Poncelet, to improve the efficiency of GSP algorithm [7], are a class of Apriori algorithm which, with the basic idea that a frequent item set means all its subsets are frequent, is a classical mining method for frequent item sets in association rules. The inverse-negative proposition, used for the pruning of frequent itemset mining, brings sharp reduction of the traversal duration and great improvement of the efficiency. The classical idea of Apriori is also used for sequential pruning in a sequential pattern. However, this class has a disadvantage that its computational complexity is largely influenced by threshold of support, number of items, number and average width of transactions, making the time complexity of producing just one frequent item set to be O(Nw), where N is the total number of transactions and w is the average width. Zaki took a study of SPADE algorithm based on a vertical format [8] to improve the execution efficiency of the class Apriori algorithm. FreeSpan algorithm [9] put forward by Han J, Pei J and Mortazvi – asl B and PrefixSpan [10]by Pei J and Han J solved the problem of low efficiency in long-sequential pattern mining, reducing the search space with projected databases and thus improving their performance. In addition, the memory-based index algorithm MEMISP[11] by Lin Ming-yen and Lee S Y greatly improves its efficiency.

3. Mining of Sequential-Pattern-Based Process Combinations

Frequently-used assembly outline combinations of processes are mined based on sequential pattern algorithm in this paper for the aircraft assembly sector. In this regard, a sequential order of processes is a timestamp for an assembly outline that can be defined as a sequence of processes, on top of which traditional sequential pattern mining algorithms are used to discover process combinations that often appear as sequences in an assembly outline.

Specific steps to implement the algorithm are as follows as the explanation of algorithm constraints and necessary definitions is the first:

Constraint 1. Process names involving all assembly outlines have been standardized, that is, there are no two names for one process.

Definition 1. All events related to object A are arranged in a timestamp-based ascending order, resulting in a sequence s of object A.

Definition 2. All assembly outlines in study are defined as set S, then \( S = \{s_1, s_2, ..., s_n\} \), where \( s_i \), \( i \in [1, n] \) represents the process sequence of order \( i \), \( n \) is the count of orders, \( s_i = <t_{i_1}, t_{i_2}, ..., t_{i_j}, ..., t_{i_m}> \), \( j \in [1, m] \), \( m \) is the total number of processes in \( s_i \), and \( t_i \) is process \( j \) in \( s_i \).

Definition 3 (Subsequences). Sequence \( t \) is a sub of another sequence \( s \), if every element in \( t \) is a subset of an ordered element in \( s \).

Definition 4 (Supports). \( \forall \) process sequence \( \omega \), support count \( \sigma(\omega) \) represents that sequence \( s \) in \( S \) contains the count of sequence \( \omega \).
Definition 5 (Sequential patterns). Given sequence set $S$ and minimum support $\text{minsupport}$, if sequence $\omega$’s support count $\sigma(\omega) \geq \text{minsupport}$, $\omega$ is called a sequential pattern.

Definition 6 (Merged process sequences). The merging strategy of assembly outline process sequences adopts the classic one $\text{apriori-gen}$. Sequences $s_1$ and $s_2$ merge, only if one obtained by removing the first event from $s_1$ is identical to another obtained by removing the last event from $s_2$.

Specific steps to implement sequential-pattern mining of assembly outline process, upon the above constraint and definitions, are as follows:

First, define the minimum support $\text{minsupport} = p$, and implement the following algorithm:

1. Set $k = 1$, $C_1$ represents candidate $k$-sequence and $F_1$ frequent $k$-sequence

   1. Find all frequent 1-sequence sets $1F_1$.

      For all sequence $s_i = \langle t_{i_1}, t_{i_2}, \ldots, t_{i_m} \rangle, i \in [1,n], j \in [1,m]$ in set $S$, candidate 1- sequence set $C_1 = \{c_{1_1}, c_{1_2}, \ldots, c_{1_m}\}, l \in [1, p]$ is defined, where $c_{1_j}$ is candidate 1- sequence and represents a process, then $C_1 = s_1 \cup s_2 \cup \ldots \cup s_n$ and $\forall c_{1_j}, \exists t_{i_j}, s.t. c_{1_j} = t_{i_j}$.

      Then frequent 1 sequence is $1F_1 = \{c_{1_j} | c_{1_j} \in C_1 \land \sigma(c_{1_j}) \geq p\},$ and $F_1 \subseteq C_1$.

   2. Set 2 to generate frequent 2 sequence $F_2$.

      1) Merge the two frequent 1 sequences into candidate 2 sequence through the classical algorithm $\text{apriori-gen}$. Candidate 2 sequence set $C_2 = \text{apriori-gen}(F_1)$.

      2) For the merged set $C_2$, define set $C_2$ to satisfy the following condition $C_2 = \{c_{2_j} | c_{2_j} \in C_2\}$, and $c_{2_j} \in C_2, \exists s_i \in S, c_{2_j}$ must be a sub of $s_i$. Each sequence here in $C_2$ is one with 2 processes.

      3) Since sequences in $C_2$ are all subs of the one in $S$, their support counts $\sigma(c_{2_j})$ are respectively calculated.

      4) Produce frequent 2 sequence $F_2$, $F_2 = \{c_{2_j} | c_{2_j} \in C_2 \land \sigma(c_{2_j}) \geq p\}$.

   3. Set $k = k + 1$ and repeat step 2 till $F_k = \emptyset$.

   4. Now we have sequential pattern $(F_1, F_2, \ldots, F_k)$, where $F_i$ represents the set of all frequent $i$ sequences, which means the set of all sequences with $i$ processes.

4. Implementation and Application

Sequential pattern $(F_1, F_2, \ldots, F_k)$, obtained through the above algorithm, is absorbed by the computer-aided process planning (CAPP) as an item of knowledge library. A recommended associated process, directly chosen and produced once meeting the compiler’s expectation, is automatically chosen by the system from the knowledge library in case of the appearance of a process in the sequence mode $(F_1, F_2, \ldots, F_k)$ during the compilation of assembly outlines, which leads to the reduction of labor and upgrading in efficiency. On the other hand, a recommended process combination brings an immediate choice and production of multi processes, meaning the multiplication of efficiency upgrading and effective reduction of possible errors or irregularities caused by manual work.
Figure 1 Traditional manual compilation vs. recommended process compilation

An obvious reduction of manual-compiled assembly outlines through the recommended method as shown in figure 1 helps come to the conclusion that the more processes involved in one outline, and the more frequently they appear in a sequence pattern, the more possible they are recommended by the outline, which results in a sharp reduction of manual labor and a high value of implementation and application.

5. Summary and Prospect
Sequential pattern mining, used for the discovery of high frequency subs of sequences, is the first time in a paper being applied in assembly outline processes for aircraft manufacturing to dig up sequential patterns that often appears at the same time with the specific pattern as (F1, F2,…, Fk). Sequential patterns mined from an assembly outline could be recorded and stored for immediate recommendation once involved in a process in a same outline compiled by the technologist. Time for inputting is sharply shortened, if the patterns are bound with standard operation instructions, and multi expressions of one process is also avoided to improve the efficiency and optimize the quality of compilation.

The mining of a sequential pattern of processes and a frequent sequence through the traditional algorithm AprioriAll, and the potential mining and visualization of a sequential pattern through FP tree generation with FP-growth in this paper give a good show of the relationship among process sequences. In addition, the following sequence shall be defined if descriptions of processes are all structured and standardized: set processes of an assembly outline as a sequence, each process an element and each description an event. An in-depth mining of sequential patterns based on the above definitions helps to figure out whether or not same procedures exist among different processes, upon which a map of operations and processes is created to update all relevant processes in the case of one change, greatly reducing the workload of technologists.

Acknowledgments
This work is supported by the National Key Research and Development Program of China (2019YFB1707402).

References
[1] YAN Yi-jun. Research and Development of AO-based Airplane Assembly Management System [D]. NUAA. Nanjing, 2007(In Chinese)
[2] Agrawal R, Imielinski T, and Swami A. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering[C], 1993.

[3] Tan P, Steinbach M, Kumar V. Introduction to Data Mining[M]. 2005.

[4] Agrawal R, Srikant R Mining sequential pattern Proc[C]. Of the 11th International Conference on Data Engineering. Taipei, 1995.

[5] Srikant R, Agrawal R Mining sequential patterns: Generalizations and performance improvements Proc[C]. Of the 5th International Conference on Extending Database Technology. Avignon, 1996.

[6] Zhang M, Kao B, Yip C, et al. A GSP-based efficient algorithm for mining frequent sequences Proc[C]. Of International Conference on Artificial Intelligence. Nevada.

[7] Masseglia F, Cathala F, Poncelet P. The PsP approach for mining sequential patterns[C] Proc. of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery. Berlin: Springer-Verlag.

[8] Zaki M J. SPADE: An efficient algorithm for mining frequent sequences[J]. Machine Learning, 2001, 41(1): 31—60.

[9] Han J, Pei J, Mortazavi—asl B, et al. FreeSpan: frequent pattern projected sequential pattern mining[C] Proc. Of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2000.

[10] Pei J, Han J. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth[C] Proc of the 7th International Conference on Data Engineering. Washington Dc: IEEE Computer Society, 2001.

[11] Lin Ming—yen, Lee S Y. Fast discovery of sequential patterns by memory indexing[C] Proc of the 4th International Conference on Data Warehousing and Knowledge Discovery. London, UK: Springer-Verlag, 2002.