

АЛГОРИТМ ГЕНЕРАЦИИ ГРАФА СОСТОЯНИЙ СБОРКИ МЕХАНИЧЕСКОГО ИЗДЕЛИЯ

Ливанцов В.Э.

*Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет), Москва*

Ключевые слова: сборка, автоматизация сборки, гиперграф структуры изделия, диаграмма Хассе, граф состояний сборки, последовательность сборки.

Аннотация. Решение задачи поиска осуществимых последовательностей с проверки геометрической доступности для деталей требует высоких вычислительных затрат. Рациональные методы анализа состояний сборки позволяют снизить число потенциальных проверок и определить заведомо неосуществимые состояния при сборке. В работе описывается алгоритм генерации графа состояний сборки механического изделия. Приводится обоснование для параметров алгоритма, обсуждается связь с моделью структуры изделия в виде гиперграфа. Рассматривается применение графа состояний для задачи поиска осуществимых последовательностей сборки.

ALGORITHM FOR CONSTRUCTING ASSEMBLY STATES GRAPH OF A MECHANICAL PRODUCT

Livantsov V.E.

Bauman Moscow State Technical University (National Research University), Moscow

Keywords: assembly, computer-aided assembly planning, mechanical structure hypergraph, Hasse diagram, assembly state graph, assembly sequence.

Abstract. Solving the problem of finding feasible sequences by checking geometric accessibility for parts requires high computational costs. Rational methods of analyzing assembly states make it possible to reduce the number of potential checks and identify obviously impracticable states during assembly. The paper describes an algorithm for generating a graph of states of assembly of a mechanical product. The rationale for the algorithm parameters is given, and the relationship with the product structure model in the form of a hypergraph is discussed. The application of the state graph for the task of finding feasible assembly sequences is considered.

Введение

Одним из наиболее важных процессов на производстве является сборка. При подготовке этапа сборки выполняется организация и настройка роботизированных средств, осуществляется подготовка персонала и помещений, формируется технологическая документация и план монтажных работ. Современные САД-системы (*computer-aided design*) оснащены как встроенными инструментами, так и АРІ (*application programming interface*) для реализации методов и моделей автоматизации проектирования. Более специфичное направление исследований, связанное с решением задач по автоматизации планирования сборки, поиском осуществимых последовательностей и их генерации, носит название СААР (*computer-aided assembly planning*).

Реализация отношения деталей при сборке путем их базирования и закрепления отражается в графной модели структуры исключением связей между

узлами и сокращением их числа. Последовательность таких сокращений можно представить в виде графа состояний. Он является простейшим носителем доступных сборочных последовательностей, который содержит информацию об ограничениях порядка установки, разделяя в каждой своей вершине множество деталей на установленные и свободные. С ростом числа деталей значительно увеличивается число состояний в графе, причем анализ каждого состояния, как было сказано ранее, может являться ресурсоемкой задачей.

В работе освещается проблема поиска осуществимых секвенциальных (выполняется при помощи двух рабочих органов) и когерентных (в рамках операции реализуется связь). Обсуждается моделирование задачи поиска в виде игры двух участников, приводится алгоритм генерации параметризованного графа состояний, который имитирует его структурные особенности, возникающие в ходе преобразования структурной модели в граф состояний.

Моделирование последовательностей сборки

Популярной моделью [1,2] для моделирования механической структуры изделия является граф механических связей или механических контактов $G=(V,E)$ (*liaison diagram, mating graph*), где V – множество вершин, сопоставленных каждой детали изделия, E – множество рёбер, причем ребро ставится только при наличии контакта двух деталей. На рисунке 1 изображено резьбовое соединение в сборе и его граф связей.

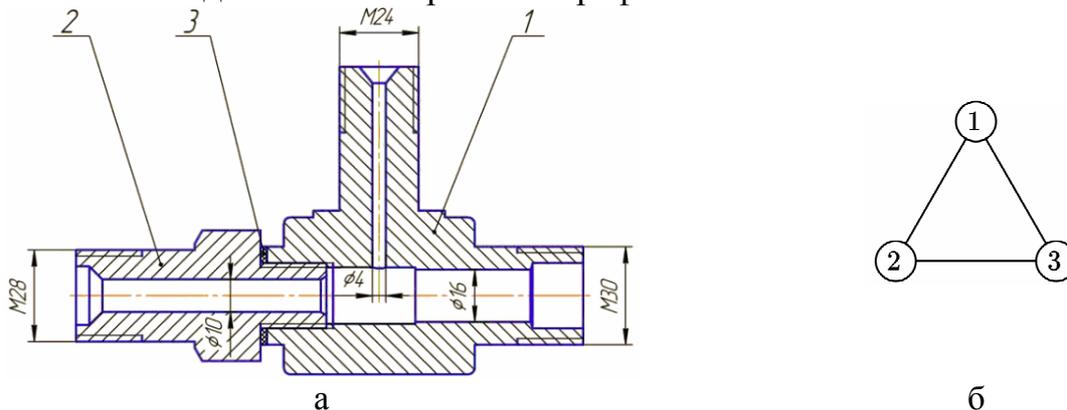


Рис. 1. а) резьбовое соединение; б) граф связей

Модель графа связей позволяет применить графоаналитические методы для поиска последовательностей, однако не учитывает отношения базирования и содержит избыточную информацию о соединениях. Этим недостатком лишена базирующаяся на G модель в виде гиперграфа $H=(V,C)$ [3], где V – множество вершин графа G , C – множество гиперрёбер, отражающий полный комплект конструкторских баз. Пример гиперграфа, сформированного из графа на рисунке 1б, изображен на рисунке 2,а.

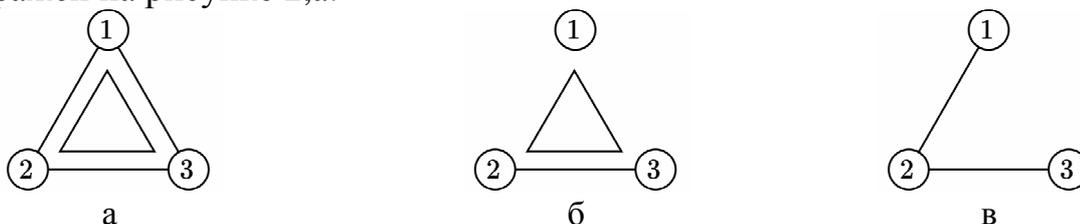


Рис. 2. Гиперграфы структуры резьбового соединения

Генерация гиперграфа происходит присоединением множества всех клик в графе к множеству рёбер исходного графа. Последовательность сборки на основании гиперграфа формируется как цепь стягиваний рёбер кратности 2, так как они отражают секвенциальную операцию, однако после генерации модель несет в себе избыточные гиперрёбра, которые порождают неопределенность при выборе следующей стягиваемой дуги. При выполнении линейного соотношения $|V|=|C|+1$ гарантируется отсутствие избыточности, а процесс удаления лишних гиперрёбер называется *линеаризация*. Линеаризованные гиперграфы продемонстрированы на рисунок 2,б и 2,в.

Граф состояний $S=(F,T)$ где F – множество вершин – некоторый собранный фрагмент или фрагменты изделия, выраженные множеством установленных деталей, T – множество рёбер, каждое из которых сопоставлен выполнению когерентной сборочной операции. Граф строится по всем возможным последовательностям нормальных стягиваний заданного гиперграфа, не учитывая их осуществимость. На рисунке 3 показаны графы состояний по гиперграфу структуры с рисунка 2,в в виде диаграммы Хассе.

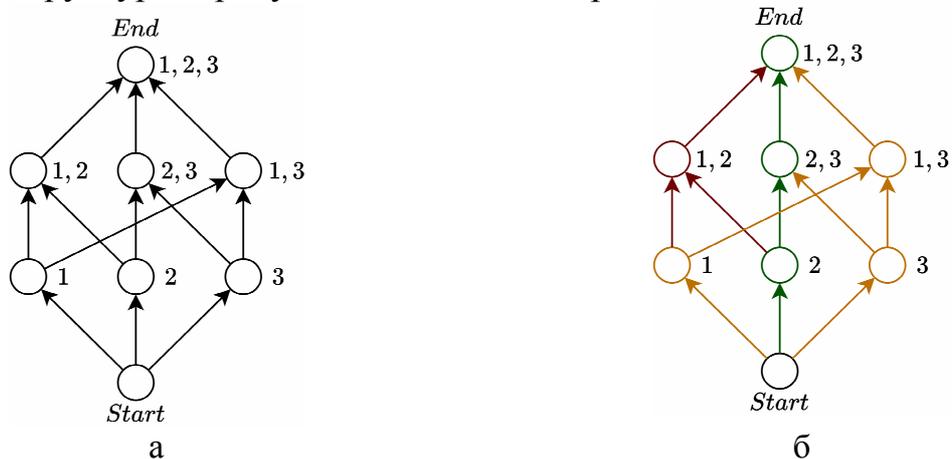


Рис. 3. Графы состояний сборки резьбового соединения

Установка детали 1 на деталь 2 блокирует возможность установки детали 3, поэтому состояние (1, 2) отмечено красным, как и инцидентные ему рёбра. Оранжевым отмечены состояния, которые требуют закрепление прокладки (3) на детали 2, что для представленной сборки не является проблемой, однако подобные ситуации при сложных формах деталей потребуют дополнительных фиксаций или инструментов. Зеленым отмечена осуществимая последовательность, операции которой секвенциальны и когерентны. В работе рассматривается алгоритм метода, который предлагает отбраковку красных состояний, однако для исключения оранжевых в моделях недостаточно информации.

Метод снижения числа геометрических проверок состояний при поиске последовательностей на выводе: если состояние невозможно, т.е. вычислительный эксперимент по анализу геометрии детали и собранного фрагмента изделия показал неизбежность коллизии детали и окружения, то все состояния, образующие конус и следующие из него вверх по графу состояний невыполнимы, как в случае $(1,2) \rightarrow (1,2,3)$. Однако если оно выполнимо, то все

вершины нижнего конуса объявляются выполнимыми. Предлагается раскрашивать осуществимые вершины-состояния в белый, неосуществимые – в черный. Процесс раскраски представляется в виде игры человека и природы, в которой ответы последней учитывают вероятность раскраски. Подробнее метод изложен в [4].

Алгоритм генерации графа состояний

Структурно, в графе состояний можно выделить три части: расширение, тело и сужение. Расширение может быть обусловлено списком деталей, с которых нельзя начинать сборку. Предполагается, что число строк расширения может быть примерно равно числу строк сужения. Количество инцидентных вершин сверху определяется накопленной степенью вершины при стягивании гиперграфа: она показывает число возможных деталей для присоединения. С учетом этого параметрами алгоритма выступают: высота графа h , число соседей сверху n_{up} , диапазон ширины тела r_w , задающийся кортежем из двух чисел и доля строк от половины графа для сужения или расширения r_e .

Для определения значений потребуется ввести несколько дополнительных функций.

$$leq(v, l) = \begin{cases} v, v \geq l; \\ l. \end{cases}, \quad geq(v, g) = \begin{cases} v, v \leq g; \\ g. \end{cases}, \quad odd(v, l) = \begin{cases} 1, v \bmod 2 = 0; \\ 0. \end{cases} \quad (1)$$

Число строк сужения и расширения, используя (1):

$$h_e = \lfloor leq(h - 3 - odd(h), 2) \cdot r_e \cdot 2 \rfloor. \quad (2)$$

Далее необходимо вывести индексы границ частей графа состояний. Деление на части обусловлено разными группами контактов: для изделий, в гиперграфе структуры которых степень вершин сильно отличается, характерно высокое число рёбер в графе состояний. Для изделий, имеющих структуру, напоминающую цепь, граф состояний будет более разреженным, а ширина каждой его строки будет лежать в узком диапазоне. Иллюстрация разбиения графа состояний на части приведена на рисунке 4,а. Для удобства отображения модель изображена без рёбер. Блок-схема алгоритма изображена на рисунке 4,б.

Параметры алгоритма для генерации подобного вида графа: $h = 18$, $r_w = [4; 6]$, $r_e = 0.5$. В столбце слева подписаны индексы строк; как и узлы соответствующих строк они имеют цветовую индикацию зоны. Фиолетовым изображена зона расширения с диапазоном строк $[g_b, g_e)$, зеленым – тела с диапазоном $[b_b, b_e)$, синим – сужения с диапазоном $[s_b, s_e)$. Ширина строки окончания расширения и начала сужения w_{avg} , а также шаг изменения ширины вне тела Δw рассчитываются по формулам:

$$w_{avg} = \left\lfloor \frac{\min r_w + \max r_w}{2} \right\rfloor, \quad \Delta w = \left\lfloor \frac{w_{avg}}{h_e} \right\rfloor. \quad (3)$$

Поскольку высота меньше 4 может не вмещать все зоны графа, для индексов границ зон частей введено специально значение -1: оно означает невыполнимость генерации частей. Расчеты сведены в таблицу 1.

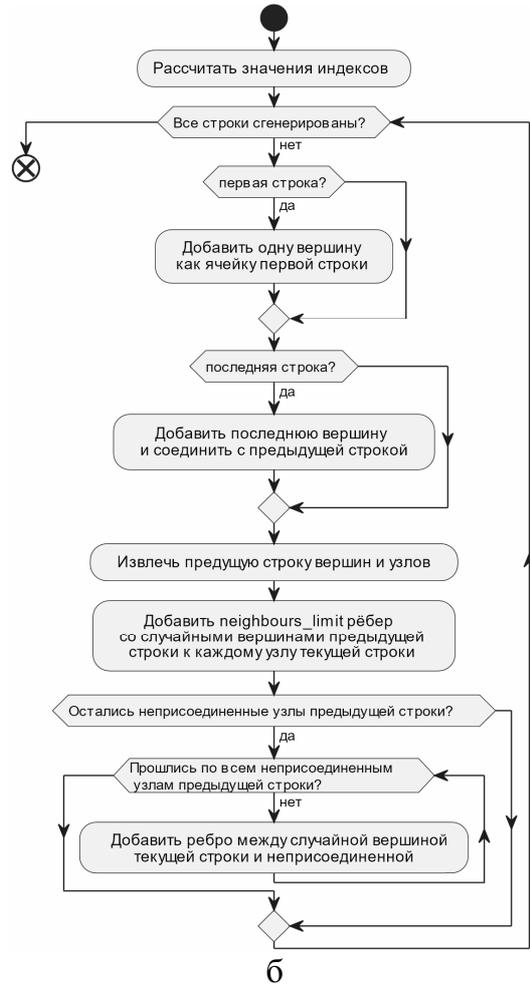
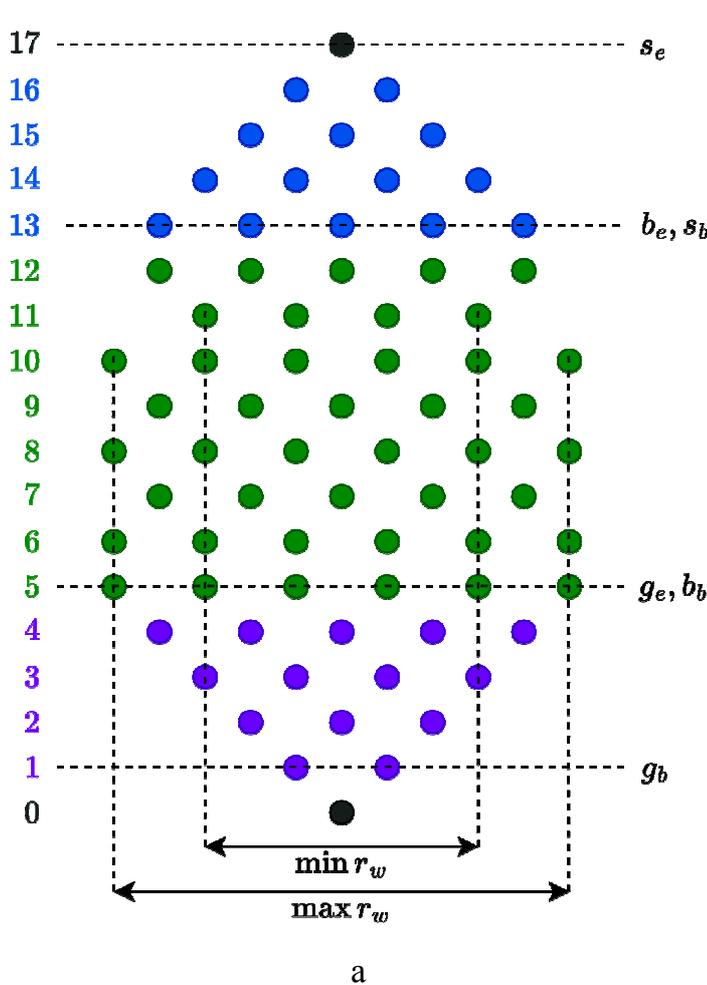


Рис. 4. Визуализация частей графа и блок-схема алгоритма

Табл. 1. Расчет индексов частей графа

Имя	Описание	Формула
g_b	Начало расширения	$\begin{cases} 1, h > 4; \\ -1 \end{cases}$
g_e	Конец расширения	$\begin{cases} 1 + h_e, g_b > -1; \\ -1 \end{cases}$
b_b	Начало тела	g_e
b_e	Конец тела	$\begin{cases} s_e - h_e, s_e > -1; \\ -1 \end{cases}$
s_b	Начало сужения	b_e
s_e	Конец сужения	$\begin{cases} h, h > 4; \\ -1 \end{cases}$

В зависимости от значения текущего индекса i строки в процессе наполнения графа состояний вычисляется её ширина w , итоговая формула:

$$w = \begin{cases} 1, i = 0; \\ geq(\Delta w \cdot (i - g_b), \max r_w), g_b \leq i < g_e; \\ rand(\min r_w, \max r_w), b_b \leq i < b_e; \\ leq(w_{avg} - \Delta w \cdot (i - s_b), 1), s_b \leq i < s_e; \\ 1, i = h - 1. \end{cases}, \quad (4)$$

где $rand(a,b)$ – функция, возвращающая случайное число в диапазоне $[a,b]$.

На рисунке 5 продемонстрированы примеры работы алгоритма.

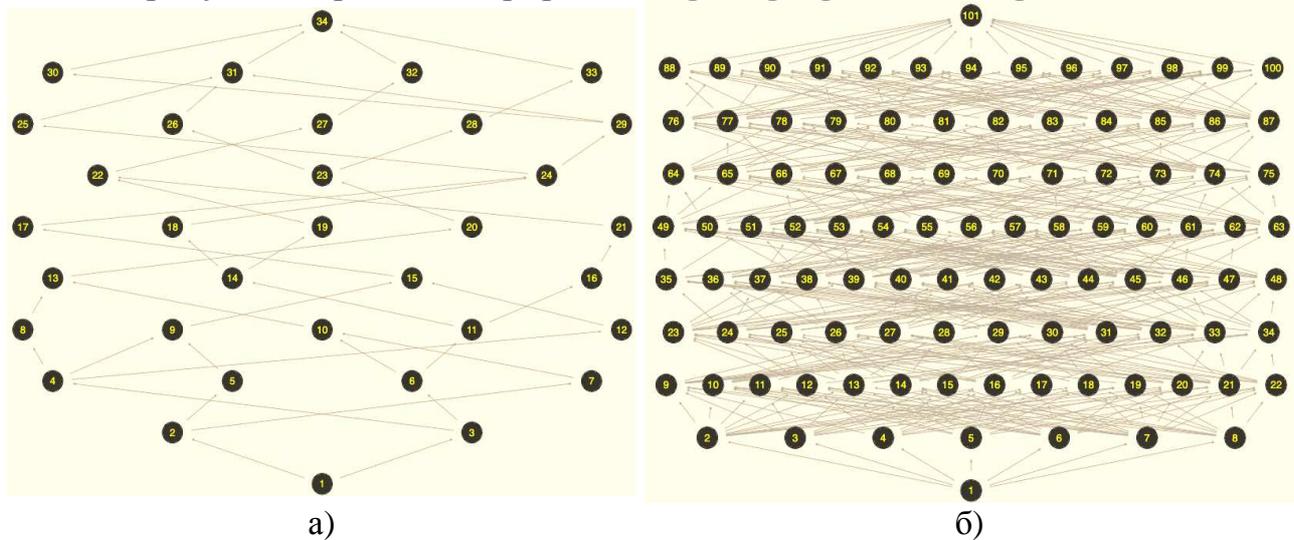


Рис. 5. Сгенерированные графы состояний

Для реализации использовался язык python. Визуализация сделана при помощи библиотек network [5] и matplotlib [6]. Визуально на рисунках 5,а и 5,б заметна разница структуры при разных параметрах, но одинаковой высоте $h = 10$ и $r_e = 0.6$: для графа (а) – $n_{up} = 1$, $r_w = [3;5]$, для графа (б) – $n_{up} = 5$.

Заключение

Представлено описание моделей для описания структуры механического изделия, показан алгоритм, приведены расчеты его внутренних параметров, приведено краткое описание их назначения и обоснование их введения. Продемонстрированы примеры работы алгоритма при разных параметрах запуска. Генерация подобных графов с разными структурными свойствами позволит провести ряд вычислительных экспериментов по сравнению критериев выбора вершин при раскраске графа состояний.

Список литературы

1. Xing Y., Chen G., Lai X., Jin S., Zhou J. Assembly sequence planning of automobile body components based on liaison graph // Assembly automation. 2007, vol. 27, no. 2, pp. 157-164.
2. Sambhoos K., Koc B., Nagi R. Extracting Assembly Mating Graphs for Assembly Variant Design // Tenth Industrial Engineering Research Conference Dallas TX IIE Transactions on Design and Manufacturing IIE Transactions on Design and Manufacturing. 2001, vol. 37, pp. 1-15.
3. Bozhko A.N. Hypergraph model for assembly sequence problem // IOP Conference Series: Materials Science and Engineering. 2019, vol. 560, no. 1, p. 012010.
4. Божко А.Н. Игровое моделирование геометрического доступа // Машиностроение и компьютерные технологии. – 2009. – №. 12. – С. 5.
5. Hagberg A., Swart P., S Chult D. Exploring network structure, dynamics, and function using NetworkX. – Los Alamos National Lab. (LANL), Los Alamos, NM (United States), 2008. – №LA-UR-08-05495; LA-UR-08-5495.
6. Hunter J.D. Matplotlib: A 2D graphics environment // Computing in science & engineering. 2007, vol. 9, no. 03, pp. 90-95.

Сведения об авторе:

Ливанцов Виктор Эдуардович – аспирант.