

УДК 658.512

АВТОМАТИЗОВАНЕ ПЛАНУВАННЯ ПОСЛІДОВНОСТІ СКЛАДАННЯ

Д. М. Проскурено, студент магістратури

E-mail: denxbr.proskurenko@gmail.com

О. В. Третьак, студент магістратури

E-mail: tr.13.elena@gmail.com

М. О. Демченко, кандидат технічних наук,

E-mail: dmariiaa@gmail.com

М. В. Філіппова, кандидат технічних наук, доцент

E-mail: m.filippova@kpi.ua

***Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»***

Анотація. Сучасне промислове виробництво потребує вдосконалення процесів складання, а отже підвищення самого рівня автоматизованого інтелектуального планування послідовності. Тому автоматизація послідовності складання виробів з метою створення більш швидких методів складання в промислових галузях є нині актуальними. Однією з проблем є створення графів процесу складання. Система планування складання може скоротити втручання людини у процес і, як наслідок, зменшити обчислювальні зусилля. Готова складальна одиниця містить в собі багато компонентів, які можна зібрати за допомогою безлічі послідовностей. Огляд методів з літератури показав, що вони хоч і підвищують рівень автоматизації, проте все одно не можуть бути застосовані в реальному виробництві, оскільки не беруть до уваги набутий досвід та знання, які можуть відіграти велику роль у плануванні і мають велику цінність. Було проаналізовано стан організації складання вузлів/виробів на сучасних виробництвах. Визначено, що планування складання здійснюється робітниками, які володіють знаннями в предметній області. Тому виникла потреба у дослідженні проблеми планування складання з метою практичної реалізації та стандартизації складальних планів. Оптимальна послідовність складання – цікавий аспект для промислового інженера, щоб мінімізувати час і вартість складання, що дає кількість рівнів складання та послідовність операцій складання. Послідовність складання з більшою кількістю паралельних можливих стабільних вузлів значно скорочує загальний час складання для великомасштабних виробів. Проаналізовано методи та підходи для оптимізації планування можливих послідовностей складання. У роботі представлена система для автоматичного створення альтернативних послідовностей складання, які можуть використовуватися проектувальниками продуктів на ранній стадії. Запропонована система генерує послідовності складання,

де одночасно можна зібрати декілька деталей. У цій роботі не використовувалися практичні обмеження, такі як сила тяжіння та необоротні складальні операції, такі як зварювання, постійне кріплення тощо.

Ключові слова: *блокуючий граф, граф зв'язку, послідовність, складання*

Актуальність. Промислові галузі в усьому світі стикаються з кількома проблемами через скорочення життєвого циклу продукції, жорстку глобальну конкуренцію та потребу в розробці складних, точних продуктів. Нині більшість операцій з розробки продукції на виробничих підприємствах автоматизовано завдяки інтеграції міждисциплінарних галузей знань. Проектування складальної одиниці та планування складання є прикладом такої інтеграції, коли продукти розроблені з урахуванням простоти складання. Мета полягає в тому, щоб проектувальники вчасно порівнювали альтернативні послідовності складання та вибирали оптимальний процес проектування. Планування складання є важливою діяльністю життєвого циклу в процесі розробки продукту, яка безпосередньо визначає час і вартість розробки продукту [1]. Складальна одиниця містить в собі багато компонентів, які можна зібрати за допомогою безлічі послідовностей. Нині планування складання здійснюється вручну за допомогою спеціалізованого технічного персоналу, що володіє глибокими знаннями в предметній області. Тому ефективний план складання, розроблений завчасно, може значно допомогти процесу розробки продукту. Інтуїтивно, вартість складання значною мірою залежить від кваліфікації персоналу та складності конструкції даного виробу. Приблизно два десятиліття тому були проведені значні дослідження в області планування складання для створення можливих послідовностей складання з геометрії CAD. У практичній реалізації вищесказаного було мало успіху. Тому виникає потреба у дослідженні проблеми планування складання з метою практичної реалізації та стандартизації складальних планів.

Аналіз останніх досліджень та публікацій. У минулому для генерації послідовності складання використовувалися два підходи: на основі знань і геометричний. Обидва були успішними до певної межі, але також мали припущення, які залишали їх неповними. Як продемонстрували Хомем де Мелло та

Сандерсон, використаний у ньому графік зв'язку був недостатнім для представлення послідовності складання. Вони розробили граф І/АБО для представлення послідовностей [2], який показує орієнтований граф, як І/АБО, необхідний для повної концептуалізації послідовностей складання. Санточчі та Діні ввели три матриці: матрицю інтерференції, матрицю контактів та матрицю з'єднання і використали їх для аналізу вузлів для створення послідовностей складання [3]. Ко і Лі запропонували «метод упорядкування частин», подібний до відношення пріоритету, щоб уникнути взаємодій/зіткнень між різними компонентами [4]. Щоб скоротити кількість етапів складання, Бен-Ар'є та Крамер розробили систему, яка генерує послідовності складання, враховуючи різні комбінації вузлів [5]. Хоча ці роботи були основою для дослідження, вони не були перевірені на складальних одиницях у складних обчислювальних задачах, а зв'язок між контактами та пріоритетом створювався вручну, що далеко від автоматизованого планування складання. У наступні роки такі дослідники, як Лапер'єр і Ель Марагі [6], Готтіполу і Гаус [7], Чжан та інші [8] запропонували декілька математичних та графічних моделей для ефективнішого створення послідовностей складання, враховуючи вектори контакту та інтерференції.

У той час як ці дослідники розглядали планування складання знизу вгору, інші використовували стратегії «декомпозиції» (зверху вниз), в якій послідовність складання генерується, починаючи з готового продукту [9]. Ву і Датта створили цю систему, використовуючи бінарний підхід розбирання (тобто одну частину за раз) за допомогою однокрокового переміщення [12]. Лі та Кумара запропонували підхід до просторів свободи та інтерференції для створення послідовності індивідуальної та групової декомпозиції [11]. Вілсон і Латомбе ввели ненаправлений блокуючий граф, щоб представити комбінаторний набір інтерференційного (передовного) зв'язку [12]. Спираючись на цей підхід, Ромні зі своєю командою запропонував систему «СТААТ», яка використовує геометричні міркування для автоматичного створення зв'язків пріоритету та послідовностей складання з файлу CAD [13]. У попередніх дослідженнях використано метод розгорнутого об'єму для генерування інтерференційних матриць [14]. Ці матриці були використані для створення

можливих послідовностей складання. Вони запропонували вдосконалений підхід генетичного алгоритму для подальшої оптимізації цих можливих послідовностей, враховуючи критерій оптимізації. Вони використовували геометричні обмеження складання та обмеження процесу складання, щоб визначити співвідношення пріоритету.

Планування послідовності складання в теорії є NP – повною проблемою для збірок, що містять велику кількість деталей. Су показав, що в порівнянні з бінарним плануванням складання (по одній частині) планування складання всього продукту набагато дорожче обчислень [14]. Складність задачі зростає в геометричній прогресії зі збільшенням кількості компонентів у складальній одиниці. Ідентифікація вузлів ділить всю складальну одиницю на обмежену кількість менших вузлів, що зменшує простір для пошуку можливої послідовності складання. Вузол є важливим критерієм для оцінки послідовності складання [15]. Його можна виконати на основі кількох обмежень складальної одиниці, таких як топологічні, геометричні та обмеження процесу складання. Важливим критерієм для вузла є те, що він повинен бути стабільним. Подібно до планування послідовності складання, проблема ідентифікації вузлів досліджувалася з використанням підходів зверху вниз і знизу вгору. Обидва підходи використовують граф пріоритету та граф підключення. Джін та ін. запропонував орієнтований граф зв'язку для представлення процесу складання з інформацією про пріоритет вузла [16]. Хомем де Мелло і Сандерсон використовували метод «вирізки» для створення всіх вузлів складальної одиниці за допомогою графа зв'язку [17]. Перевага цього методу полягає в тому, що ідентифікація вузлів може бути здійснена за допомогою алгоритмів пошуку графів. Кан з командою використовували алгоритм пошуку графів з графом пріоритету розбирання відповідно до графа І-АБО складальної одиниці [18]. Онг і Ван запропонували метод, за допомогою якого вони використовують функціональні зв'язки між деталями, стабільність складальної одиниці та базовий компонент для ідентифікації вузла [19].

Мета дослідження. У цій статті пропонується вдосконалити систему планування складання для створення графіка зв'язку та графіка пріоритету складання, які будуть

використовуватися для генерування можливих послідовностей складання з ідентифікацією вузлів. На відміну від вищезгаданих методів, запропонована система генерує послідовності складання, де одночасно можна зібрати декілька деталей. Далі розглядатиметься запропонована методологія з системою, що має модульну структуру. Для оцінки продуктивності системи наводяться два приклади генерації послідовності складання та ідентифікації вузлів.

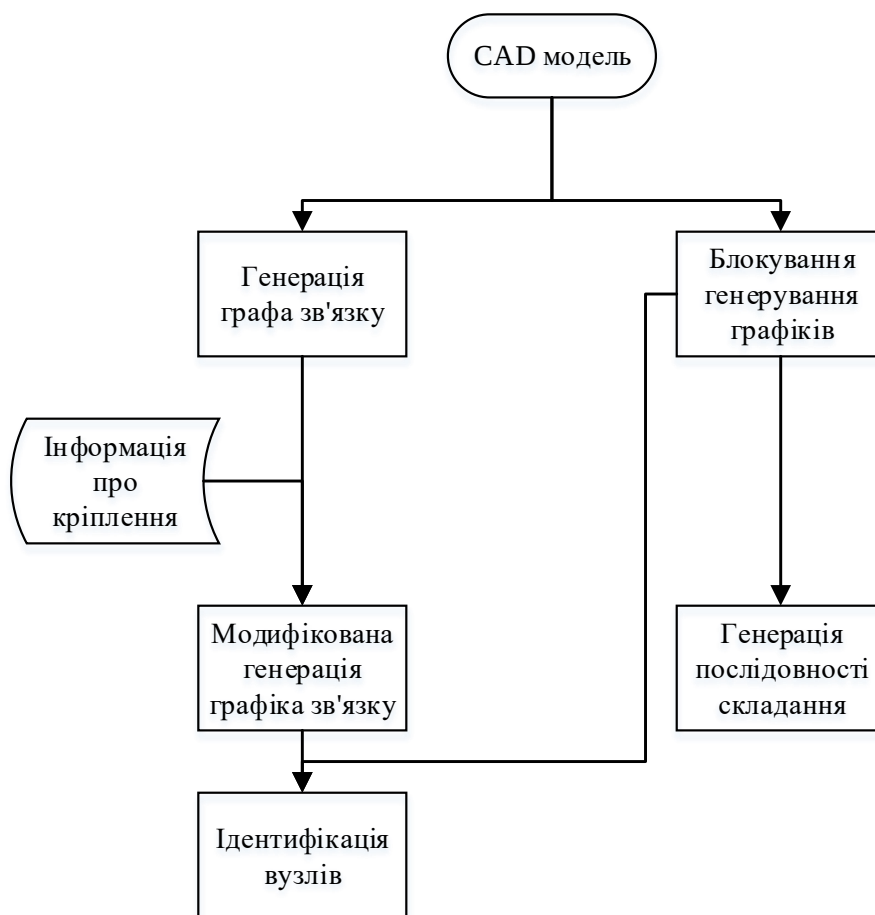


Рис. 1. Блок-схема системи планування складання

Матеріали і методи дослідження. Запропонована система є модульною за архітектурою, що складається з двох основних модулів – генерування послідовності складання та ідентифікації вузлів. Модуль послідовності складання приймає файл CAD (файл STP) складальної одиниці та генерує можливі послідовності складання. Цей модуль далі підрозділяється на три підмодулі: генератор графів зв'язку, генератор блокувальних графів і генератор послідовності. Отримавши інформацію про кріплення для складальної одиниці, модуль вузлів генерує модифікований графік зв'язку та блок

блокування, щоб ідентифікувати всі можливі вузли. Цей алгоритм пояснюється у вигляді блок-схеми на рис. 1.

Результати досліджень та їх обговорення. *Модуль генерації послідовності складання.* Використовується підхід «декомпозиції» для створення можливих послідовностей. Тому припускаємо, що складання є процесом, зворотним процесу декомпозиції. Цей підхід використовує геометричні міркування для створення зв'язку з пріоритетом. Цей зв'язок пріоритету потім використовуються для визначення послідовності розбирання. Доступні послідовності складання потім генеруються шляхом повернення цих послідовностей декомпозиції. Будемо використовувати Open Cascade на платформі C# як пакет CAD для виконання різних геометричних операцій. Open Cascade Technology – це платформа для розробки програмного забезпечення з відкритим кодом для 3D CAD, CAM, CAE тощо . Модуль отримує файли САПР складання як вхідні дані. Потім вхідні дані попередньо обробляються для індексації компонентів, включених до складальної одиниці. Усі підмодулі цього модуля описані нижче.

Графік зв'язку – це відображення контактної інформації між будь-якими двома частинами даної складальної одиниці. У графі зв'язку кожна частина складальної одиниці представлена у вигляді вузла. Між будь-якими двома вузлами на графі зв'язку буде межа, якщо частини, що відповідають вузлам, контактують одна з одною в кінцевому виробі. На рис.3 показано один із таких прикладів графа зв'язку, де можна бачити, що частини 1, 2 складальної одиниці з 5 частин (див. рис.2) з'єднані, а отже, у графі зв'язку вони з'єднані ребром.

Представлення у вигляді графу зв'язку займає $\Theta(V + E)$ місця, тоді як матричне представлення займає $\Theta(|V|^2)$ місця [20]. Тому використовуємо графічне представлення, щоб зменшити простір для зберігання, необхідний для великих складальних одиниць. Щоб визначити контакт між двома частинами, перевіряємо, чи мають дві частини контактні грані. Перетворюємо кожен частину в плоскі грані та перевіряємо 3d-перетин відповідних пар цих граней. Якщо будь-яка пара перетинається в межах заданого допуску, робимо висновок, що ці частини

контактують. Цей процес повторюється до тих пір, поки не буде вичерпано всю комбінацію частин.

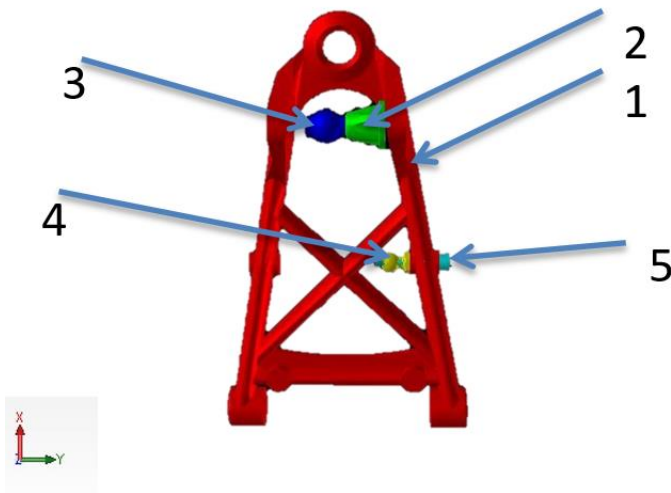


Рис. 2. Приклад складання з 5 частин

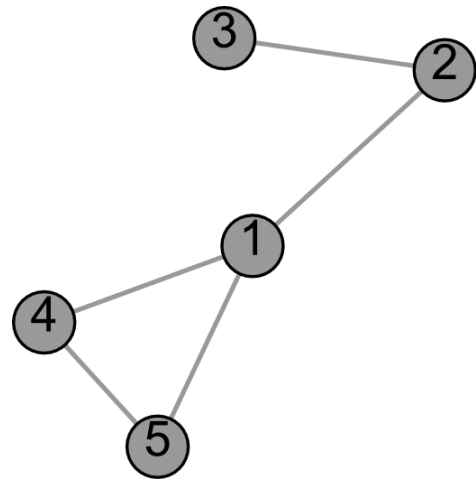


Рис. 3. Граф зв'язку для складання з 5 частин

Блок блокування – графік блокування підсумовує взаємозв'язок блокування всередині збірки для заданого напрямку складання. Технічно блокуючі графіки можна генерувати для будь-якого визначуваного напрямку, але через підвищену складність обчислень використовується вибіркова підмножина векторів напрямку. У даній роботі припускаємо, що складання виконується в напрямках $\pm x$, y , z . Сам блокуючий граф є орієнтованим графом, де вузли є компонентами виробу, а попередники вузла блокуються відповідним вузлом у цьому конкретному напрямку. Приклад графіка блокування для виробу з 5 компонентами показано на рис.4. Показаний графік блокування для напрямку $+x$, де можна бачити, що, оскільки частини 1, 4, 5 блокують один одного в напрямку $+Z$, і, отже, вони з'єднані двонаправленими ребрами в графі. Частини 2, 3 не блокуються жодними компонентами під час складання у відповідному напрямку, а отже, не з'єднані жодним ребром у графі блокування.

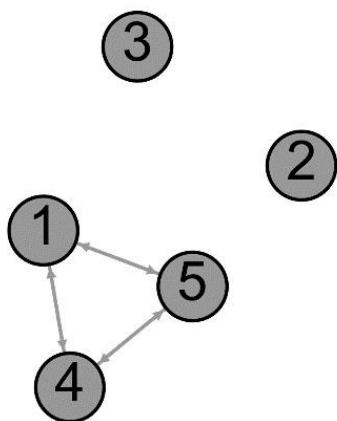


Рис. 4. Граф блокування в напрямку +x

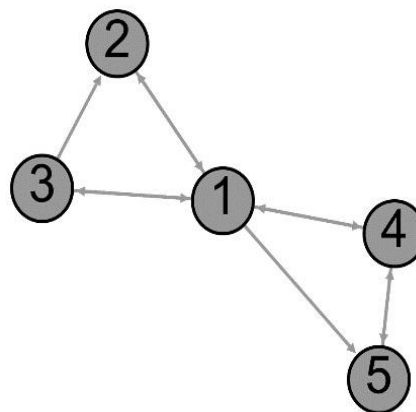


Рис. 5. Граф блокування в напрямку +y

Побудова блокувального графа відповідає методології, подібної до генерації графа зв'язку. Проектуємо промені від граничних точок граней однієї частини (P_i) на відповідну грань іншого компонента (P_j). Якщо промені перетинають грані P_j , робимо висновок, що P_j блокує компонент P_i . Тому додаємо ребро від P_i до P_j . Приклад такого типу перетину показаний на рис.6. Цей модуль генерує 6 різних графіків блокування, що відповідають кожному напрямку.

Генерація послідовності складання. У цій роботі розглядаємо лише блокуючий графік для створення зв'язку пріоритету. Граф зв'язку використовується в другому модулі для ідентифікації вузлів. Основною ідеєю створення послідовності розбирання є видалення деталей, які не блокуються іншими деталями. Після створення всіх блокуючих графів створюємо список вузлів (R), які не мають наступників (вузлів з вихідними ребрами). Якщо набір частин блокує один одного у всьому блокувальному графі, це означає, що їх потрібно видалити разом, тому в послідовності ці частини групуються разом. Частина(и) потім випадковим чином вибирається для видалення з R , і всі шість блокуючих графіків оновлюються. Процес повторюється до тих пір, поки в блокуючих графах не залишиться жодних вузлів. Після створення послідовностей розбирання вони повертаються, щоб визначити послідовності складання.

Ідентифікація вузлів. Алгоритм приймає інформацію про кріплення як вхідні дані і генерує стабільні вузли як вихідні дані. Основна ідея цього алгоритму полягає в тому, щоб згрупувати компоненти, які скріплені разом як можливі вузли, і

визначити, які з цих можливих вузлів є стабільними. Стабільний вузол – це набір компонентів, які мають скріплені контакти і не можуть бути легко зняті окремо, але загальний набір можна зняти разом на окремих етапах розбирання.

Цей алгоритм бере граф зв'язку, згенерований з попереднього модуля та інформації про кріплення, щоб створити модифікований графік зв'язку, який визначає, чи є з'єднання закріпленням з'єднанням чи ні [21]. Цей модифікований граф зв'язку генерується зміною ваги ребер графа. Якщо будь-які два компоненти з'єднані кріпильним з'єднанням, то вага відповідного краю в графі зв'язку збільшується на 1. Алгоритм передбачає, що всі деталі/вузли зібрані на базовому компоненті. Передбачається, що базовий компонент є окремою частиною, щоб видалити тривіальне рішення, яке полягає в тому, що повна складальна одиниця також є збіркою.

Базовий компонент — це компонент, що відповідає вузлу з найвищим ступенем зв'язку, тобто вузлу, що має найвищий ступінь у модифікованому графі зв'язку (див. рис.6).

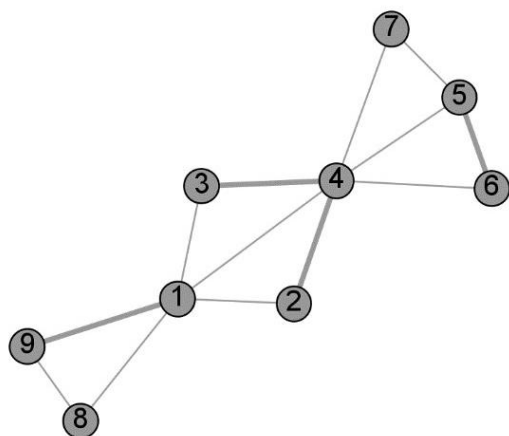


Рис. 6. Змінений графік зв'язку

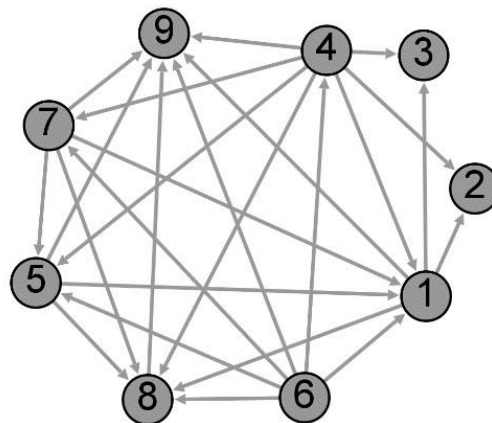
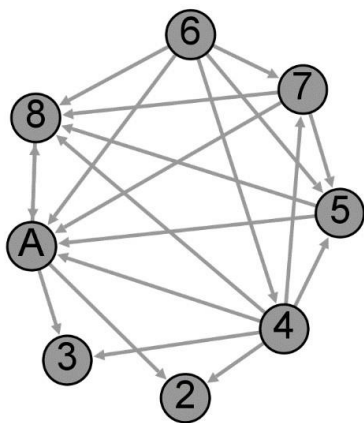


Рис. 7. Блок блокування для

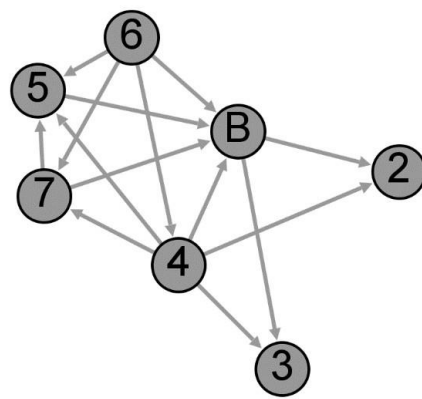
для складальної одиниці з 9 частин

складальної одиниці з 9 частин

Компоненти, що мають кріпильні з'єднання, потім групуються в набори S . На рис.6 з'єднання кріпильних елементів показано з товстими краями. Вузли $\{2, 3, 4\}$, $\{5, 6\}$, $\{1, 9\}$ з'єднані за допомогою кріпильних з'єднань і, отже, будуть згруповані в набір S . Отже, $S = \{\{2, 3, 4\}, \{5, 6\}, \{1, 9\}\}$. Аналогічне групування виконується в блокуючих графах, тобто ці вузли замінюються одним вузлом у графі (див. рис.8).



**Рис. 8. Згрупований графік
блокування**



**Рис. 9. Остаточний графік
блокування**

Ця зміна в графі блокування може створити цикли в графі. Наявність циклу вказує на те, що компоненти, що становлять цикл, не можуть бути видалені окремо, і тому їх необхідно розібрати разом. Якщо граф блокування містить цикл, то всі компоненти, що утворюють цикл, додаються до S . Цей процес повторюється до тих пір, поки в графах блокування більше не буде циклів. Відповідно до визначення базового компонента, 4 є базовим компонентом. Оскільки 4 є базовим компонентом, тому він не може бути частиною будь-якої підгрупи, і будь-який набір в S , що має базовий компонент, видаляється з S , коли в графі блокування більше немає циклів. Отримана множина S являє собою множину всіх стабільних вузлів.

Вивчення проблеми. Виріб (див. рис.10), що складається з 23 деталей, був обраний в якості прикладу для генерації послідовності складання, яка є частиною збірки RC автомобіля. Створені системою графіки зв'язку та блокувальні графіки показані на рис. 11, 12, 13.

Використовуючи наведену вище інформацію, система дає бажану кількість можливих послідовностей складання на виході за 247 секунд. Через складність обчислень цієї проблеми дуже мало дослідників представили час, необхідний для створення зв'язку пріоритету. Тому важко порівнювати результати літератури. Одна з можливих послідовностей, створених запропонованим алгоритмом, наведена нижче в таблиці. Ця інформація зберігається у форматі xml, який можна використовувати для визначення вартості складання та інших критеріїв оцінки складання.

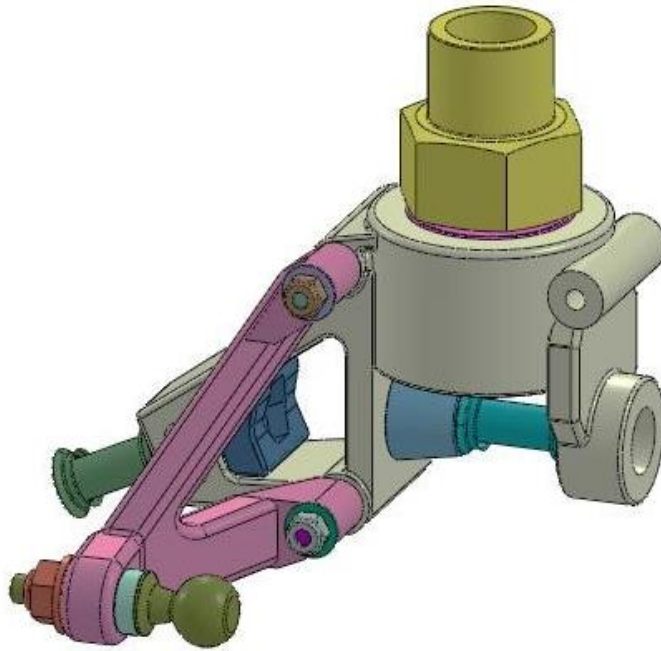


Рис. 10. Збірка з 23 деталей

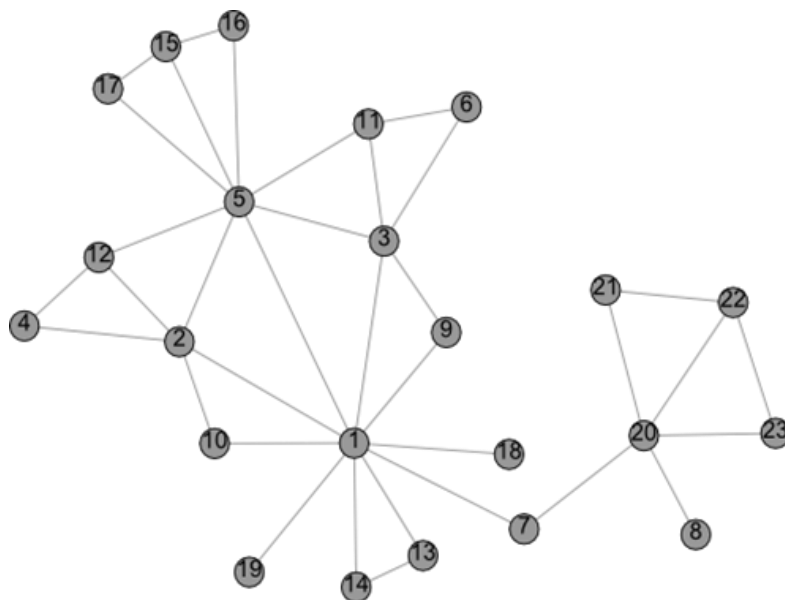


Рис. 11. Графік зв'язку для складання з 23 деталей

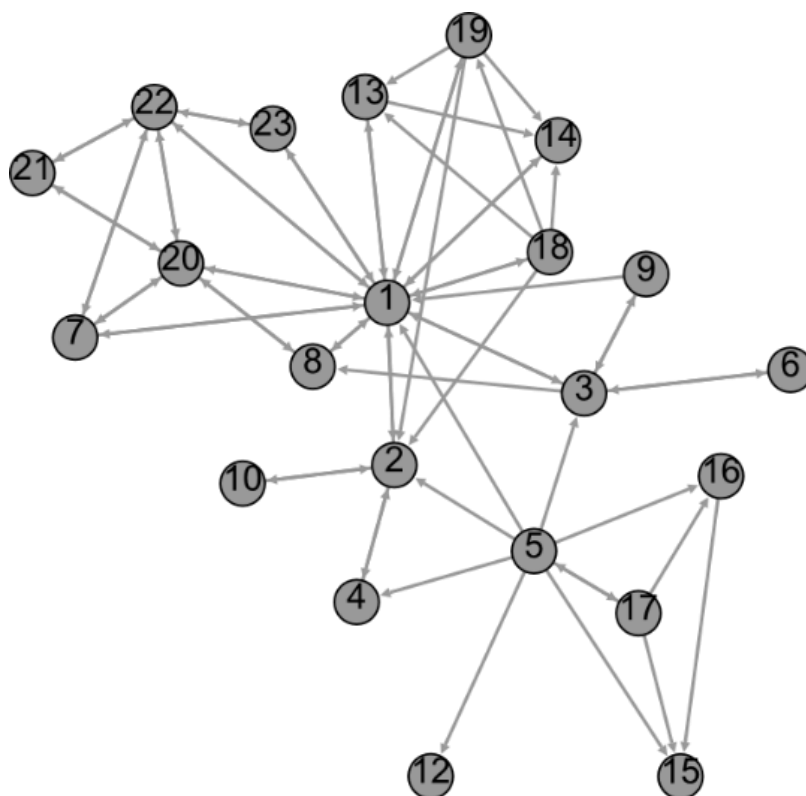


Рис. 12. Графік блокування в напрямку $+x$

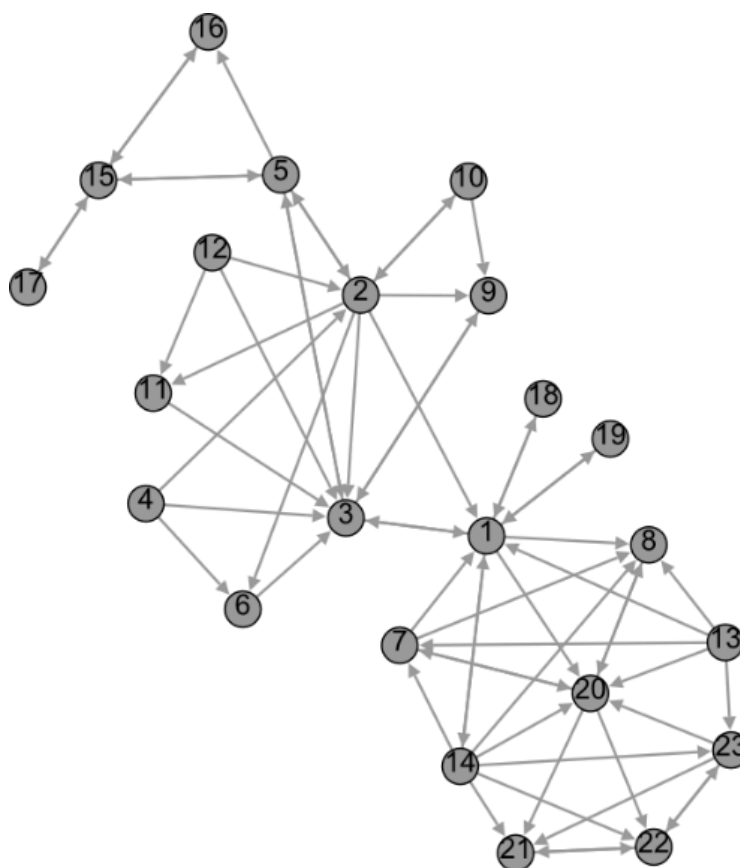


Рис. 13. Графік блокування в напрямку $+y$

Можлива послідовність складання

Частини	2	10	3	1	7	19	9	5
Напрямок	$-x$	$-z$	$-x$	$-z$	y	$-z$	$-z$	$-z$
Частини	14	18	13	8	23	20	22	21
Напрямок	$-z$	$-z$	$-z$	$-y$	$-y$	$-y$	$-y$	$-y$
Частини	12	4	17	16	15	6	11	
Напрямок	$-x$	$-x$	$-x$	$-x$	$-x$	$-x$	$-x$	

Висновки і перспективи. У цій статті представлено систему для автоматичного створення альтернативних можливих послідовностей складання, які можуть використовуватися проектувальниками продуктів на ранній стадії. Альтернативні послідовності допомагають проектувальникам у виборі найкращої послідовності на основі різних сценаріїв. Поточна система генерує послідовності складання ефективним і надійним способом.

Однак у міру того, як кількість частин наближається до 100, час, який витрачається на створення графіка блокування, збільшується експоненціально. У цій роботі не використовувалися практичні обмеження, такі як сила тяжіння та необоротні складальні операції, такі як зварювання, постійне кріплення тощо. Використовуючи систему, засновану на знаннях, користувач може визначити кілька правил для послідовності складання на основі різних операцій складання, і ці правила будуть запускатися щоразу, коли буде виконана певна умова. Ця інтегрована система може не тільки значно скоротити час, необхідний для створення послідовностей складання, але й практичність послідовностей буде кращою.

Список використаних джерел

1. Boothroyd G., Dewhurst P., Winston K. Product Design for Manufacture and Assembly. Boca Raton: CRC Press, 2010. 712 p.
2. Homem de mello L. S., Sanderson A. C. Representations of mechanical assembly sequences. IEEE Transactions on Robotics and Automation. 1991., No. 2 Vol. 7. P. 211-227.
3. Santochi M., and Dini G. Computer-aided planning of assembly operations: the selection of the assembly sequences. Int. J. Robot. Comput. Integr. Manuf. 1992. Vol. 9, no. 6. P. 439–446.

4. Ko H., Lee K. Automatic assembling procedure generation from mating conditions. *Computer-Aided. Design*. 1997. Vol. 19, no. 1. P. 3–10.
5. Ben-Arieh D., and Kramer B. Computer-aided process planning for assembly: generation of assembly operations sequence. *Int. J. Prod.* 1994. Vol. 32, no. 3. P. 643–656.
6. Laperrière L., and ElMaraghy H. A. GAPP: a generative assembly process planner. *J. Manuf. Syst.* 1996. Vol. 15, no. 4. P. 282–293.
7. Gottipolu R. B., and Ghosh K. A simplified and efficient representation for evaluation and selection of assembly sequences. *Comput. Ind.* 2003. Vol. 50, no. 3. P. 251–264.
8. Zhang Y. Z., Ni J., Lin Z. Q., and Lai X. M. Automated sequencing and sub-assembly detection in automobile body assembly planning. *J. Mater. Process. Technol.* 2002. Vol. 129, no 1. P. 490–494.
9. Halperin D., Latombe J.-C., and Wilson R. H. A general framework for assembly planning: The motion space approach. *Algorithmica*. 2000. Vol. 26, no. 3. P. 577–601.
10. Woo T. C., and Dutta D. Automatic disassembly and total ordering in three dimensions. *Journal of Engineering for Industry*. 1991. Vol. 113, no. 2. P. 207–213.
11. Lee Y. Q., and Kumara S. Individual and group disassembly sequence generation through freedom and interference spaces. *Journal of Design and Manufacturing*. 1992. Vol. 2. P. 143–154.
12. Wilson R. H., and Latombe J.-C. Geometric reasoning about mechanical assembly. *Artif. Intell.* 1994. Vol. 71, no. 2. P. 371–396.
13. Romney B., Godard C., Goldwasser M., Ramkumar G. An efficient system for geometric assembly sequence generation and evaluation. *Comput. Eng.* 1995. P. 699–712.
14. Su Q. A hierarchical approach on assembly sequence planning and optimal sequences analyzing. *Robot. Comput. Integr. Manuf.* 2009. Vol. 25, no. 1. P. 224–234.
15. Lee H.-R., and Gemmill D. D. Improved methods of assembly sequence determination for automatic assembly systems. *Eur. J. Oper. Res.* 2001. Vol. 131, no. 3. P. 611–621.
16. Jin S., Cai W., Lai X., and Lin Z. Design automation and optimization of assembly sequences for complex mechanical systems. *Int. J. Adv. Manuf. Technol.* 2010. Vol. 48, no. 9. P. 1045–1059.
17. Homem de Mello L. S., and Sanderson A. C. AND/OR graph representation of assembly plans. *Robot. Autom. IEEE Trans.* 1991. Vol. 6, no. 2. P. 188–199.
18. Kang J.-G., Lee D.-H., Xirouchakis P., and Persson J.-G. Parallel disassembly sequencing with sequence-dependent operation times. *CIRP Ann. Technol.* 2001. Vol. 50, no. 1. P. 343–346.
19. Ong N. S., and Wong Y. C. Automatic Subassembly Detection from a Product Model for Disassembly Sequence Generation. *Int. J. Adv. Manuf. Technol.* 1999. Vol. 15, no. 6. P. 425–431.
20. Філіппова М. В., Вислоух С. П. Методика автоматизованого проектування технології складання виробів приладобудування. *Вісник НТУУ «КПІ». Серія «Машинобудування*. Київ, 2006. С. 111–117.
21. Створене посилання: Антонюк В. С., Вислоух С. П., Филиппова М. В. Автоматизированное проектирование технологических процессов сборки изделий

приборостроения. Сборка в машиностроение и приборостроении. 2007. Вып. 6. С. 3-5.

References

1. Boothroyd, G., Dewhurst, P., & Knight, W. A. (2010). Product design for manufacture and assembly. Boca Raton, FL: Crc Press.
2. Homem de Mello, L. S., & Sanderson, A. C. (1991). Representations of mechanical assembly sequences. IEEE Transactions on Robotics and Automation, 7(2), 211–227. <https://doi.org/10.1109/70.75904>
3. Santochi M., and Dini G. (1992). Computer-aided planning of assembly operations: the selection of the assembly sequences. Int. J. Robot. Comput. Integr. Manuf., 9(6), 439–446.
4. Ko, H., & Lee, K. (1997). Automatic assembling procedure generation from mating conditions. Computer-Aided Design, 19(1), 3–10. [https://doi.org/10.1016/0010-4485\(87\)90146-1](https://doi.org/10.1016/0010-4485(87)90146-1)
5. BEN-ARIEH, D., & KRAMER, B. (1994). Computer-aided process planning for assembly: generation of assembly operations sequence. International Journal of Production Research, 32(3), 643–656. <https://doi.org/10.1080/00207549408956957>
6. Laperrière, L., & ElMaraghy, H. A. (1996). GAPP: A generative assembly process planner. Journal of Manufacturing Systems, 15(4), 282–293. [https://doi.org/10.1016/0278-6125\(96\)84553-5](https://doi.org/10.1016/0278-6125(96)84553-5)
7. Gottipolu, R. B., & Ghosh, K. (2003). A simplified and efficient representation for evaluation and selection of assembly sequences. Computers in Industry, 50(3), 251–264. [https://doi.org/10.1016/s0166-3615\(03\)00015-0](https://doi.org/10.1016/s0166-3615(03)00015-0)
8. Zhang, Y. Z., Ni, J., Lin, Z. Q., & Lai, X. M. (2002). Automated sequencing and sub-assembly detection in automobile body assembly planning. Journal of Materials Processing Technology, 129(1-3), 490–494. [https://doi.org/10.1016/s0924-0136\(02\)00621-0](https://doi.org/10.1016/s0924-0136(02)00621-0)
9. Halperin, D., Latombe, J.-C. ., & Wilson, R. H. (2000). A General Framework for Assembly Planning: The Motion Space Approach. Algorithmica, 26(3-4), 577–601. <https://doi.org/10.1007/s004539910025>
10. Woo, T. C., & Dutta, D. (1991). Automatic Disassembly and Total Ordering in Three Dimensions. Journal of Engineering for Industry, 113(2), 207–213. <https://doi.org/10.1115/1.2899679>
11. Lee Y. Q., and Kumara S. (1992). Individual and group disassembly sequence generation through freedom and interference spaces. Journal of Design and Manufacturing., 2, 143–154.
12. Wilson, R. H., & Latombe, J.-C. (1994). Geometric reasoning about mechanical assembly. Artificial Intelligence, 71(2), 371–396. [https://doi.org/10.1016/0004-3702\(94\)90048-5](https://doi.org/10.1016/0004-3702(94)90048-5)
13. Romney B., Godard C., Goldwasser M., Ramkumar G. (1995). An efficient system for geometric assembly sequence generation and evaluation. Comput. Eng. 699–712.
14. Su, Q. (2009). A hierarchical approach on assembly sequence planning and optimal sequences analyzing. Robotics and Computer-Integrated Manufacturing, 25(1), 224–234. <https://doi.org/10.1016/j.rcim.2007.11.006>

15. Lee, H.-R., & Gemmill, D. D. (2001). Improved methods of assembly sequence determination for automatic assembly systems. *European Journal of Operational Research*, 131(3), 611–621. [https://doi.org/10.1016/s0377-2217\(00\)00103-x](https://doi.org/10.1016/s0377-2217(00)00103-x)
16. Jin, S., Cai, W., Lai, X., & Lin, Z. (2009). Design automation and optimization of assembly sequences for complex mechanical systems. *The International Journal of Advanced Manufacturing Technology*, 48(9-12), 1045–1059. <https://doi.org/10.1007/s00170-009-2361-8>
17. Homem de Mello, L. S., & Sanderson, A. C. (1990). AND/OR graph representation of assembly plans. *IEEE Transactions on Robotics and Automation*, 6(2), 188–199. <https://doi.org/10.1109/70.54734>
18. Kang, J.-G., Lee, D.-H., Xirouchakis, P., & Persson, J.-G. (2001). Parallel Disassembly Sequencing with Sequence-Dependent Operation Times. *CIRP Annals*, 50(1), 343–346. [https://doi.org/10.1016/s0007-8506\(07\)62136-2](https://doi.org/10.1016/s0007-8506(07)62136-2)
19. Ong, N. S., & Wong, Y. C. (1999). Automatic Subassembly Detection from a Product Model for Disassembly Sequence Generation. *The International Journal of Advanced Manufacturing Technology*, 15(6), 425–431. <https://doi.org/10.1007/s001700050086>
20. Filippova M. V., Vysloukh S. P. (2006). Metodika avtomatizovanoho proektuvannya tekhnolohiyi skladannya virobiv priladobuduvannya [Methods of automated design of technology for assembling instrumentation products]. *Visnik NTUU «KPI»*. Kyiv, 111-117.
21. Antonyuk V. S., Vysloukh S. P., Filippova M. V. (2007). Avtomatizirovannoe proektirovanie tekhnolohicheskikh protsessov sborki izdeliy priborostroeniya [Computer-aided design of technological processes for assembling instrumentation products]. *Sborka v mashinostroenie i priborostroenii*. 6(3). 3-5.

АВТОМАТИЗИРОВАННОЕ ПЛАНИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ СБОРКИ

Д. М. Проскурено, Е. В. Третьяк, М. А. Демченко, М. В. Филиппова

Аннотация. Современное промышленное производство требует усовершенствования процессов сборки, а значит, повышения самого уровня автоматизированного интеллектуального планирования последовательности. Поэтому автоматизация последовательности сборки изделий с целью создания более быстрых методов сборки в промышленных отраслях сегодня актуальна. Одной из проблем является создание графов процесса сборки. Система планирования сборки может сократить вмешательство человека в процесс и, как следствие, снизить вычислительные усилия. Готовая сборочная единица включает в себя много компонентов, которые можно собрать с помощью множества последовательностей. Обзор методов по литературе показал, что они хоть и повышают уровень автоматизации, однако все равно не применимы в реальном производстве, поскольку не принимают во внимание приобретенный опыт и знания, которые могут играть большую роль в планировании и имеют большую ценность. Было проанализировано состояние организации сборки узлов/изделий на современных производствах. Определено, что планирование сборки осуществляется рабочими, обладающими знаниями в предметной области. Поэтому возникла необходимость в исследовании проблемы планирования

составления с целью практической реализации и стандартизации сборочных планов. Оптимальная последовательность сборки – интересный аспект для промышленного инженера, чтобы минимизировать время и стоимость сборки, что дает количество уровней сборки и последовательность операций сборки. Последовательность сборки с большим количеством возможных параллельных стабильных узлов значительно сокращает общее время сборки для крупномасштабных изделий. Проанализированы методы и подходы для оптимизации планирования возможных последовательностей сборки. В работе представлена система автоматического создания альтернативных последовательностей сборки, которая может использоваться проектировщиками продуктов на ранней стадии. Предлагаемая система генерирует последовательность сборки, где одновременно можно собрать несколько деталей. В этой работе не использовались практические ограничения, такие как сила тяжести и необратимые сборочные операции, такие как сварка, постоянное крепление и другие.

Ключевые слова: *блокирующий граф, граф связи, последовательность, сборка*

AUTOMATED PLANNING OF GRADUALITY ASSEMBLY

D. Proskurenko, O. Tretyak, M. Demchenko, M. Filippova

Abstract. Modern industrial production requires the improvement of assembly processes, and thus increase the level of automated intelligent sequence planning. Therefore, researches in the field of automation of the sequence of assembly of products in industries are relevant at this time. In today's world there is a need to develop complex, accurate products. Problems are created in industries due to the reduction of the life cycle of products. There is a need to study the problem of assembly planning to achieve the goal of practical implementation and standardization of assembly plans. Creating graphs of the addition process is one of the problems. The assembly planning system can reduce human intervention in the process and reduce computational effort. The finished assembly contains many components that can be assembled using many sequences. A review of the methods from the literature showed that although these methods increase the automation level, they still cannot be applied to actual production because they do not take into account the experience and knowledge that can play a major role in planning and are of great value. Assembly planning, relationship charts, priority charts. Improving the assembly planning system to create a communication schedule and an assembly priority schedule was proposed. The advanced system will be used to generate possible assembly sequences with subassembly identification. A system has been developed to create alternative possible assembly sequences that can be used by component part / product designers in the early stages. A system capable of generating assembly sequences for simultaneous assembly of multiple parts has been proposed. Conclusions and work results can be applied used and improved for more productive product development by designers in the early stages and faster assembly of products in enterprises. The paper did not consider practical limitations (gravity) and irreversible assembly operations, such as permanent fastening, welding etc.

Key words: *assembly, blocking graph, relation graph, sequence*