

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**приладобудівний факультет
кафедра виробництва приладів**

«На правах рукопису»
УДК 658.511.3

До захисту допущено:

Завідувач кафедри

 Віктор АНТОНЮК

«10» 12 2021 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

**«Комп'ютерно-інтегровані технології виробництва приладів»
зі спеціальності 151 «Автоматизація та комп'ютерно-інтегровані
технології»**

на тему: «Проектування процесів виробництва на основі методології SADT»

Виконав (-ла):

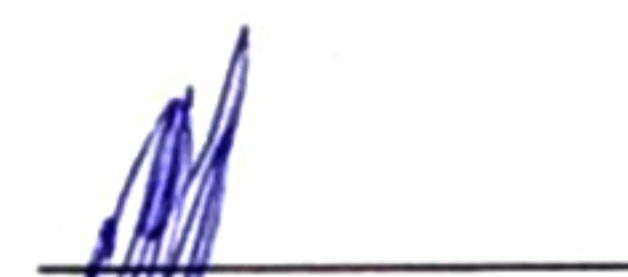
студент (-ка) VI курсу, групи ПБ-01мп

Проскурєнко Денис Миколайович



Науковий керівник:

доцент кафедри виробництва приладів, к.т.н., доцент
Філіппова Марина В'ячеславівна



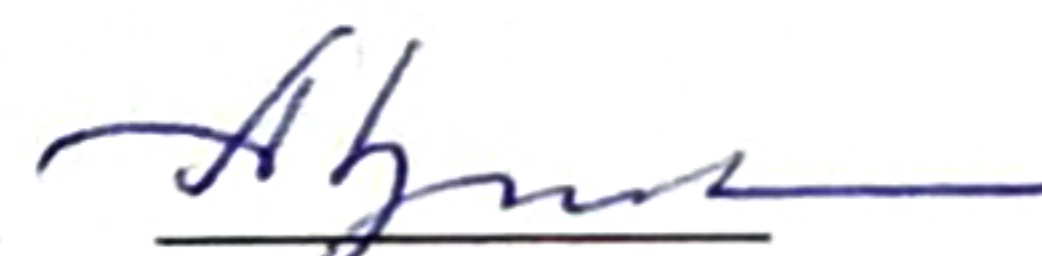
Консультант зі стартап-проекту:

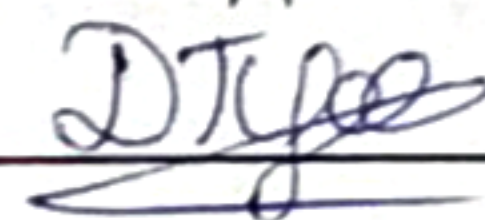
професор кафедри менеджменту підприємств, д.е.н., професор,
Бояринова Катерина Олександрівна



Рецензент:

Професор кафедри комп'ютерно-інтегрованих систем та навігаційних систем, д.т.н., доцент
Аврутов Вадим Вікторович



Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент (-ка) 

Київ – 2021 року

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

Приладобудівний факультет

Кафедра виробництва приладів

Рівень вищої освіти – другий (магістерський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані технології
виробництва приладів»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віктор АНТОНЮК

«10» 12 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Проскуренку Денису Миколайовичу

1. Тема дисертації «Проектування процесів виробництва на основі методології SADT», науковий керівник дисертації Філіппова Марина В'ячеславівна, к.т.н, доцент, затверджені наказом по університету від «03» листопада 2021 р. № 3664-с
2. Термін подання студентом дисертації 10 грудня 2021 року
3. Об'єкт дослідження виробничий процес складання виробів
4. Вихідні дані: автоматизована система проектування процесів складання.
5. Перелік завдань, які потрібно розробити: Сформулювати завдання дисертації та дослідити літературні джерела. Провести системне моделювання процесів автоматизованого технологічного проектування. Розробити математичне та інформаційне забезпечення системи автоматизованого проектування складання. Розробити програмне забезпечення та дослідити експлуатацію системи автоматизованого проектування складання. Розробити стартап-проект «автоматизація проектування складання на основі методології SADT».
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація за результатами дисертаційного дослідження.
7. Орієнтовний перелік публікацій: статті та доповіді на науково-технічних конференціях за результатами дисертаційного дослідження.

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стартап-проект	Бояринова К.О., професор кафедри менеджменту підприємств		

9. Дата видачі завдання 10.09.2021

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вступ та дослідження літературних джерел	24.09.2021	
2	Системне моделювання процесів автоматизованого технологічного проектування	08.10.2021	
3	Математичне та інформаційне забезпечення системи автоматизованого проектування складання	22.10.2021	
4	Програмна реалізація та дослідна експлуатація системи автоматизованого проектування складання	12.11.2021	
5	Розробка стартап-проекту «автоматизація проектування складання на основі методології SADT»	26.11.2021	
6	Оформлення пояснювальної записки та графічної частини проекту	03.12.2021	

Студент



Денис ПРОСКУРЕНКО

Науковий керівник



Марина ФІЛІППОВА

* Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

АНОТАЦІЯ

Магістерська дисертація складається зі: вступу, п'ять розділів, висновків, переліку використаної літератури та додатків у вигляді тексту програми, технологічної документації, переліку наукових праць та акту впровадження.

Основний текст складається з 123 сторінок, включає 26 таблиць та 28 малюнків.

У вступі обґрунтовано актуальність проблеми, сформульовано мету й завдання роботи, наукову новизну та практичну корисність, основні положення захисту.

У першому розділі дисертації представлені: огляд стану проектування технологічних процесів складання, концепція методології SADT, методи проектування технологічних процесів складання та опис систем автоматизованого проектування технологічних процесів, а також формулювання дисертаційного дослідження.

Другий розділ присвячено системному моделюванню процесів автоматизованого технологічного проектування, а саме представлено математична модель виробу і процесу проектування технології, та функціональна модель розроблена на основі методології SADT.

У третьому розділі розглядається математичне та інформаційне забезпечення системи автоматизованого проектування складання.

Четвертий розділ описує практичне використання розробленої системи: детальний опис розробленої програми та приклад її використання.

П'ятий розділ «Управління стартап-проектом» описує проект, технологічний аудит проектної ідеї, аналіз ринкових можливостей для запуску стартап-проекту, а також розробку ринкової стратегії для проекту.

Ключові слова: SADT, ТП, САПР, складання, послідовність складання, генетичний алгоритм, граф зв'язку, STEP, 3Д-модель, прилад, складальна одиниця, Python.

ABSTRACT

The master's dissertation consists of: an introduction, five chapters, conclusions, a list of references and appendices in the form of the text of the program, technological documentation, a list of scientific papers and the act of implementation.

The main text consists of 123 pages, includes 26 tables and 28 figures.

The introduction substantiates the urgency of the problem, formulates the purpose and objectives of the work, scientific novelty and practical usefulness, the main provisions of protection.

The first section of the dissertation presents: a review of the state of design technological processes of assembly, the concept of SADT methodology, methods of designing technological processes of assembly and description of automated design processes, as well as the formulation of dissertation research.

The second section is devoted to system modeling of automated technological design processes, namely the mathematical model of the product and the technology design process, and the functional model developed on the basis of SADT methodology.

The third section considers the mathematical and information support of the automated assembly design system.

The fourth section describes the practical use of the developed system: a detailed description of the developed program and an example of its use.

The fifth section "Startup Project Management" describes the project, technological audit of the project idea, analysis of market opportunities for launching a startup project, as well as developing a market strategy for the project.

Keywords: SADT, TP, CAD, assembly, assembly sequence, genetic algorithm, communication graph, STEP, 3D model, device, assembly unit, Python.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ.....	13
1.1 Опис методології SADT	13
1.2. Методи проектування технологічних процесів складання.....	22
1.2.1. Комбінаторний підхід	29
1.2.2. Геометричний підхід.....	30
1.2.3 Алгоритми оптимізації послідовності складання	32
1.3. Огляд систем автоматизованого проектування.....	35
ВИСНОВКИ ДО РОЗДІЛУ.....	43
РОЗДІЛ 2. СИСТЕМНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ АВТОМАТИЗОВАНОГО ТЕХНОЛОГІЧНОГО ПРОЕКТУВАННЯ	44
2.1 Математичні моделі у задачах аналізу і синтезу процесів складання	44
2.1.1 Математична модель складального виробу.....	44
2.1.2 Математична модель процесу проектування технології.....	47
2.2 Функціональна модель автоматизованого проектування технологічного процесу складання.....	51
ВИСНОВКИ ДО РОЗДІЛУ.....	53
РОЗДІЛ 3. МАТЕМАТИЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ВИРОБНИЧИХ ПРОЦЕСІВ СКЛАДАННЯ	54
3.1 Модель інформації моделі САПР	55
3.1.1 Інформаційна модель складальної одиниці	55
3.1.2 Інформаційна модель складеного приладу	56
3.1.3 Вилучення та дослідження даних складання моделі САПР.....	57
3.2 Математичне забезпечення генерації послідовності складання	66
3.2.1 Генетичний алгоритм для генерації послідовності складання	68
3.2.2 Граф зв'язку	74
ВИСНОВКИ ДО РОЗДІЛУ.....	77
РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ СКЛАДАННЯ.....	78
4.1 Модель роботи інтерфейсу програми.....	79
4.2. Програмна реалізація підсистеми автоматизованого проектування складання.....	81
4.3. Дослідна експлуатація розробленої програми.....	86
ВИСНОВКИ ДО РОЗДІЛУ.....	92
РОЗДІЛ 5. РОЗРОБКА СТАРТАП-ПРОЕКТУ «АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ СКЛАДАННЯ НА ОСНОВІ МЕТОДОЛОГІЇ SADT».....	94
5.1. Опис ідеї проекту та аналіз ринкових можливостей запуску стартап-проекту	94

5.2. Технологічний аудит ідеї проекту	96
5.3. Розроблення ринкової стратегії проекту	107
5.4. Розроблення маркетингової програми стартап-проекту.....	110
ВИСНОВКИ ДО РОЗДІЛУ	116
ЗАГАЛЬНІ ВИСНОВКИ ДО МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ.....	117
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	120
ДОДАТОК А Код програми	129
ДОДАТОК Б Результат роботи програми	141
ДОДАТОК В Публікації.....	148
ДОДАТОК Г Презентація	151

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

САПР – система автоматизованого проектування;

SADT – методологія структурного, структуризованого аналізу і проектування;

PDM – управління даними про продукт;

САПР ТП – система автоматизованого проектування технологічних процесів;

ТП – технологічні процеси;

ТПС – технологічний процес складання;

САП – система комп’ютерного проектування технологічних процесів;

ГА – генетичний алгоритм.

ВСТУП

У сучасній глобальній економіці виробничі компанії працюють в умовах жорсткої конкуренції та ринкового середовища, що динамічно змінюється. Вимоги замовника щодо продукту вищої якості, нижчої вартості та коротшого життєвого циклу продукції тиснуть на виробничий процес та впровадження нових технологій. Витрати та якісні параметри слід враховувати на ранніх етапах процесу проектування. Необхідність розуміння наслідків проектних рішень на виробництво та якість продукції є визначальною при оцінці ефективності та результативності виробничих процесів.

Виробничий процес — це сполучення предметів, знарядь праці та живої праці в просторі і часі, що функціонують для задоволення потреб виробництва. Технологічний процес становить основу будь-якого виробничого процесу, є найважливішою його частиною, яка пов'язана з переробленням сировини, обробкою матеріалів і перетворенням їх у готову продукцію.

Ефективність виробництва визначається здатністю точно оцінити технологічний і виробничий цикли складання продукції, тому складання є дуже важливим виробничим видом діяльності. На процеси складання зазвичай припадає значний відсоток загального часу виробництва та загальної вартості виробництва. У традиційному промисловому виробництві процеси складання займають близько 50% загального часу виробництва і становлять понад 20% загальної вартості виробництва. Таким чином, щоб бути конкурентоспроможними в сучасному ринковому середовищі, виробники повинні постійно прагнути до скорочення часу та витрат на складання.

Актуальність роботи

Традиційно проектування продукту відбувається в два етапи. По-перше, інженер-конструктор спроектує виріб на основі вимог. Тоді інженер-виробник виготовить виріб, якщо зможе. Інакше він передасть обмеження

інженеру-конструктору, щоб він перепроєктував продукт. Цей процес повторюється безперервно до тих пір, поки не буде розроблено задовільний продукт. Це збільшить час виготовлення виробу. Тому необхідно скоротити час розробки нового продукту. Це створює величезне завдання для проєктувальників, щоб враховувати такі фактори, як дизайн для складання/розкладання, дизайн для технологічності, дизайн для закінчення терміну служби тощо, спільно відомі як DFX. Рішенням цієї проблеми є одночасне проєктування продукту. Враховуючи ці фактори на ранній стадії проєктування, можна значно скоротити час і витрати на розробку продукту. Серед цих факторів планування складання є найважливішим фактором одночасного проєктування продукту, і воно займає дуже багато часу. За даними Nof et al. у [1] витрати на планування складання становлять близько 50% виробничих витрат. Отже, хороша система планування складання може значно знизити собівартість виробництва. Автоматизація процесу планування складання допоможе проєктувальникам приймати рішення, які призведуть до кращого планування складання і, отже, скорочення часу виробництва та витрат. Перевагами автоматизованого планування складання є:

- Можна створити декілька можливих послідовностей складання, і інженер може вибрати потрібну послідовність на основі таких критеріїв, як наявність машини, вартість утримання тощо.
- Виробничі обмеження можуть бути включені швидше в процес проєктування, що призведе до скорочення часу проєктування.
- При серійному виробництві час переналадки набагато менше. Автоматичне планування складання допоможе швидшій заміні людей, машин, інструментів тощо.

Традиційні системи САПР не мають вбудованих можливостей для виконання планування послідовності складання. Таким чином, більшість розроблених програм для планувальння складання працюють незалежно від систем САПР і тому вимагають, щоб користувачі вручну переводили

проектну інформацію, що міститься в моделі САПР, в інформацію, сумісну з програмою. Типові інструменти планування складання вимагають, щоб користувачі вручну визначали та інтерактивно вводили зв'язки обмежень між компонентами. Хоча взаємодія людини під час планування послідовності складання надає більшу гнучкість у встановленні критеріїв оцінки якості послідовності складання, ручний переклад і взаємодія займають дуже багато часу та схильні до помилок. Таким чином, для підвищення ефективності та точності планування послідовності складання дуже важливою є ефективна автоматизована інтеграція проектів САПР із планувальниками послідовності складання.

Щоб покращити інтеграцію та зв'язок між проектами САПР і програмою для планування послідовності складання, запропонований метод використовує нейтральний формат файлу САПР для виведення інформації про проект із систем САПР і для введення інформації про проектування в розроблену програму. STEP (Стандарт для обміну даними моделі продукту) — це широко підтримуваний формат файлів САПР, який сприяє ефективному обміну даними та обміну інформацією між різними системами САПР або між системами САПР і САП.

Це дослідження було проведено для ефективної інтеграції систем САПР із програмою планування послідовності складання та для перевірки того, що автоматизоване планування послідовності складання може допомогти скоротити час складання. Робота, представлена в цьому дослідженні, включає автоматичне вилучення проектної інформації з файлів STEP, переведення інформації про проектування в інформацію про геометричні обмеження виробу, використання програмного забезпечення для створення можливих послідовностей складання з найменшою кількістю переорієнтацій складання та перевірку того, що послідовності складання рекомендовані, зменшуючи переорієнтацію складання, допомагає скоротити загальний час складання як для роботи, так і для ручних операцій складання.

Мета та задачі дослідження. Метою роботи є розробка принципів та методів проєктування процесів виробництва, спрямована на удосконалення технологічної підготовки виробництва шляхом створення базису для наскрізної автоматизації виробництва. Для досягнення мети були поставлені наступні задачі, направлені на розробку:

- Функціональних моделей об'єктів та процесів предметної області проєктування виробничих процесів виготовлення виробів;
- моделі представлення формалізованих знань про предметну область і методики формування цієї моделі;
- методів і алгоритмів прийняття рішень у задачах проєктування маршрутів складання виробів;
- концепції автоматизованого проєктування виробничих процесів, що базується на застосуванні апарату обробки формалізованих знань.

Об'єктом дослідження є виробничий процес складання виробів, а **предметом дослідження** – функціональні моделі виробничих процесів складання.

Наукова новизна, заключається у адаптації методології SADT для роботи розробленого програмного забезпечення, яке створене для побудови плану складання та проєктування технологічних процесів складання.

Практична цінність:

- Розроблено генетичний алгоритм для вирішення задачі автоматизації послідовності складання;
- розроблено програмне забезпечення з інтерфейсом користувача для швидкого аналізу 3Д-модулі виробу, автоматизованого створення послідовності складання та генерації виробничого плану.

РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

1.1 Опис методології SADT

SADT («Structured Analysis and Design Technique» - методологія структурного, структуризованого аналізу і проектування) - має довгу історію в галузі програмного забезпечення. Вона була розроблена Россом (1977) в результаті постійної роботи (1969-1973) з вирішення проблем, починаючи з 1950-х років у Softech [2]. SADT є графічною мовою і широко використовується для опису складних систем у комунікативних конструкціях, військовому плануванні та автоматизованому виробництві (Dickover, McGowan, Ross 1977). SADT була успішно застосована при аналізі проблем та функціональних специфікаціях; проте вона була найбільш ефективно використаний на етапі визначення вимог до проектування програмного забезпечення (Росс та Шоман 1977; Росс 1985). SADT була прийнята як Icam DEFinition for Function Modeling (IDEF0) інтегрованим комп'ютерним виробництвом ВПС США (ICAM) у 1980-х роках. Це було частиною програми ICAM для задоволення потреби в потужних методах моделювання для системного аналізу та проектування. Протягом цього періоду було впроваджено кілька інших пакетів IDEF; переважно, IDEF1, IDEF2, IDEF3 та IDEF1х. Оскільки ми зосереджені переважно на кореневих визначеннях та мітках, розроблених у SADT, ми вважаємо за краще використовувати SADT замість IDEF0. Однак IDEF0 згадується, коли звертаються до будь-якої пов'язаної роботи.

Метою SADT є організація, відображення та надання простої техніки для ідентифікації та перехресного посилення на всю інформацію, необхідну для публікації прийнятних вимог та специфікацій проекту. SADT - це інструмент для структуризації інформації та дозволяє інженерам краще зрозуміти проблему, яку потрібно вирішити. Тому SADT в основному використовується для встановлення мислення та структурування фаз аналізу та проектування. SADT не надає повної специфікації.

Представлення компонентів, відносин та правил

Модель SADT - це розуміння; просте уявлення якоїсь реальності, адекватної певної мети. Для досягнення переваг принципів структурування, особливо зверху вниз, рівні деталізації та ієрархії модель має бути графічною. Кожна модель SADT складається з набору пов'язаних діаграм (рис. 1.1), які організовані зверху вниз. Кожна діаграма є або зведеною (батьківською) діаграмою, або детальною (дочірньою) діаграмою батьків.

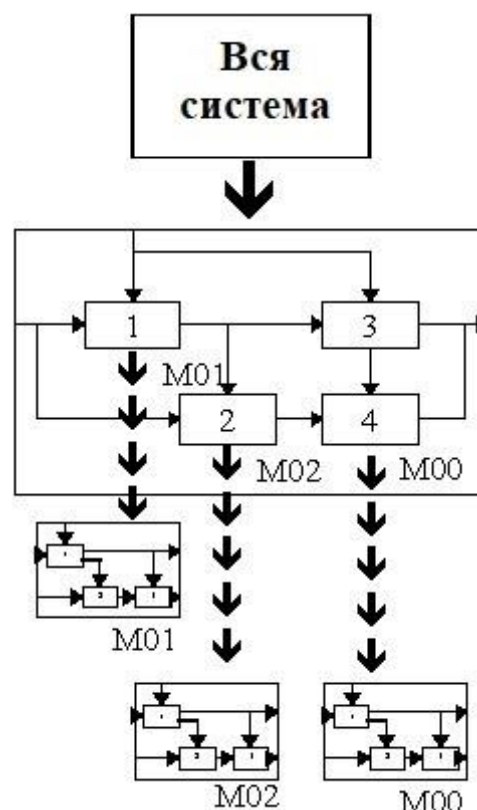


Рис. 1.1. Модель SADT.

Цифрами 1-4 позначено батьківські діаграми, а M00-M02 – це дочірні діаграми, які розкривають детально батьківську діаграму.

Хронологічні числа або "С-числа" використовуються для ідентифікації унікальних версій діаграми, а також вони використовуються для з'єднання діаграм у ієрархії моделі. С-номер діаграми, яка розкладає поле, розміщується під цим вікном на батьківській діаграмі. Тоді батьківська діаграма посилається на дочірню діаграму.

Під час розробки моделі SADT діаграма з її полями та стрілками зазвичай перемальовується кілька разів, створюючи різні версії діаграми. SADT використовує схему керування конфігурацією діаграми, побудовану на основі хронологічних чисел або "С-номерів", щоб відрізняти різні версії однієї і тієї ж діаграми один від одного. Код С-номера складається з ініціалів автора та унікального порядкового номера. Ці коди розміщені у нижньому правому куті форми SADT. Ці С-номери також використовуються для ідентифікації унікальних версій діаграми, вони також використовуються для з'єднання діаграм як вниз, так і вгору в ієрархії моделі.

SADT використовує два типи моделей, а саме моделі діяльності та моделі даних. Модель діяльності SADT описує декомпозицію діяльності. Модель даних SADT описує декомпозицію даних. У лівій частині вікна є вхідні дані або (у випадку моделі даних) діяльність, яка генерує дані. У правій частині вікна є вихідні дані або діяльність, яка використовує дані з вікна. У верхній частині вікна є контрольні дані або діяльність. SADT розрізняє вхідні та контрольні дані. Вхідні дані діаграми фактично трансформуються. Контрольні дані використовуються лише для керівництва діяльністю.

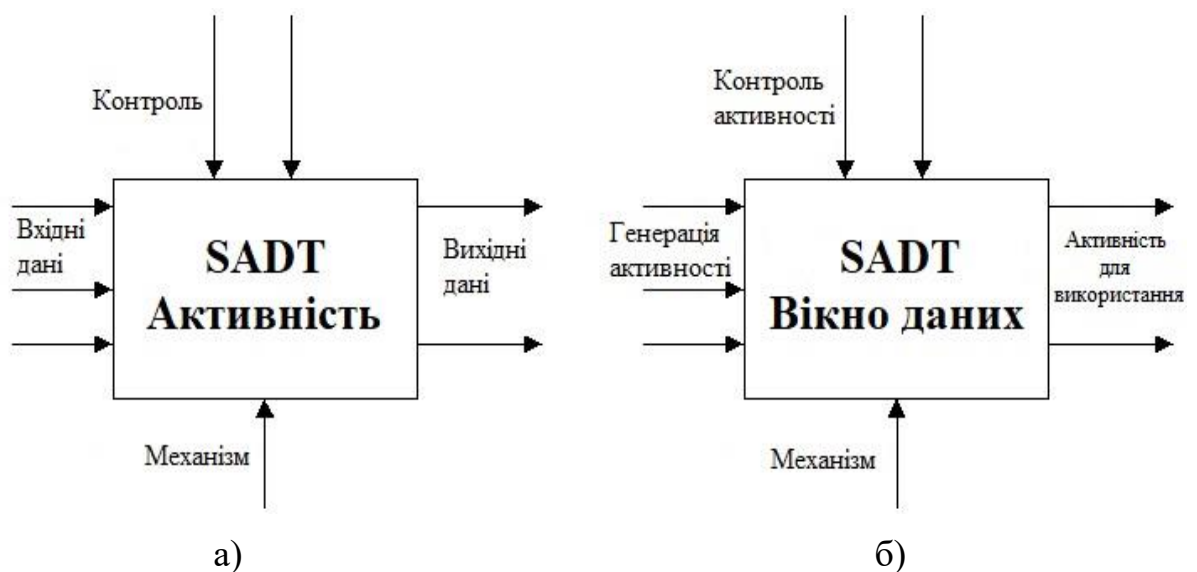


Рис. 1.2. Прості уявлення про два типи моделей:

а) модель даних;

б) модель діяльності.

Позначення SADT складаються з діаграм (блоків) зі стрілками з чотирма стрілками на кожній стороні, що визначаються як: введення, виведення, управління та механізм, і одна дія посередині, як показано на Рис.1.2(а). Їх визначення складаються з наступного:

- Дія(активність): Дія - це будь-яка функція або процес, що служить для перетворення вхідних даних у вихідні
- Вхідні дані: Дані/інформація, необхідні діяльності для початку процесу трансформації
- Вихідні дані: дані/інформація, отримані в результаті діяльності в результаті цієї трансформації
- Контроль: Будь-яке обмеження, яке певним чином впливає на поведінку дії
- Механізм: Особи, ресурси або будь-які засоби, необхідні для здійснення дії

SADT використовує традиційний підхід «зверху вниз» для ієрархічного моделювання функцій. Процес починається з визначення найвищого спрощеного рівня деталізації на діаграмі верхнього рівня, який потім розкладається на додаткові деталі на кожному кроці, поки не буде досягнутий необхідний рівень деталізації. Цей підхід показаний на Рис. 1.3. На першому етапі створюється діаграма верхнього рівня А. Ця діаграма просто стверджує, що вхід І0 перетворюється на вихід О0 за допомогою діяльності А, обмеженої керуванням С0, і з використанням деякого ресурсу R0. Зазвичай С0 і R0 - це групові позначення, які вказують на те, що цей конкретний тип обмеження або ресурсу знадобиться для завершення цього перетворення. На наступному етапі А розкладається на подальші піддіяльності, демонструючи більш детальний погляд на процес нижчого рівня (рис. 1.3). Тепер кожен етап показує окремі вхідні дані, результати, елементи управління та ресурси; проте вони дискретно працюють над тим, щоб створити таке ж перетворення (І0 в О0), як і в оригінальній діаграмі високого рівня.

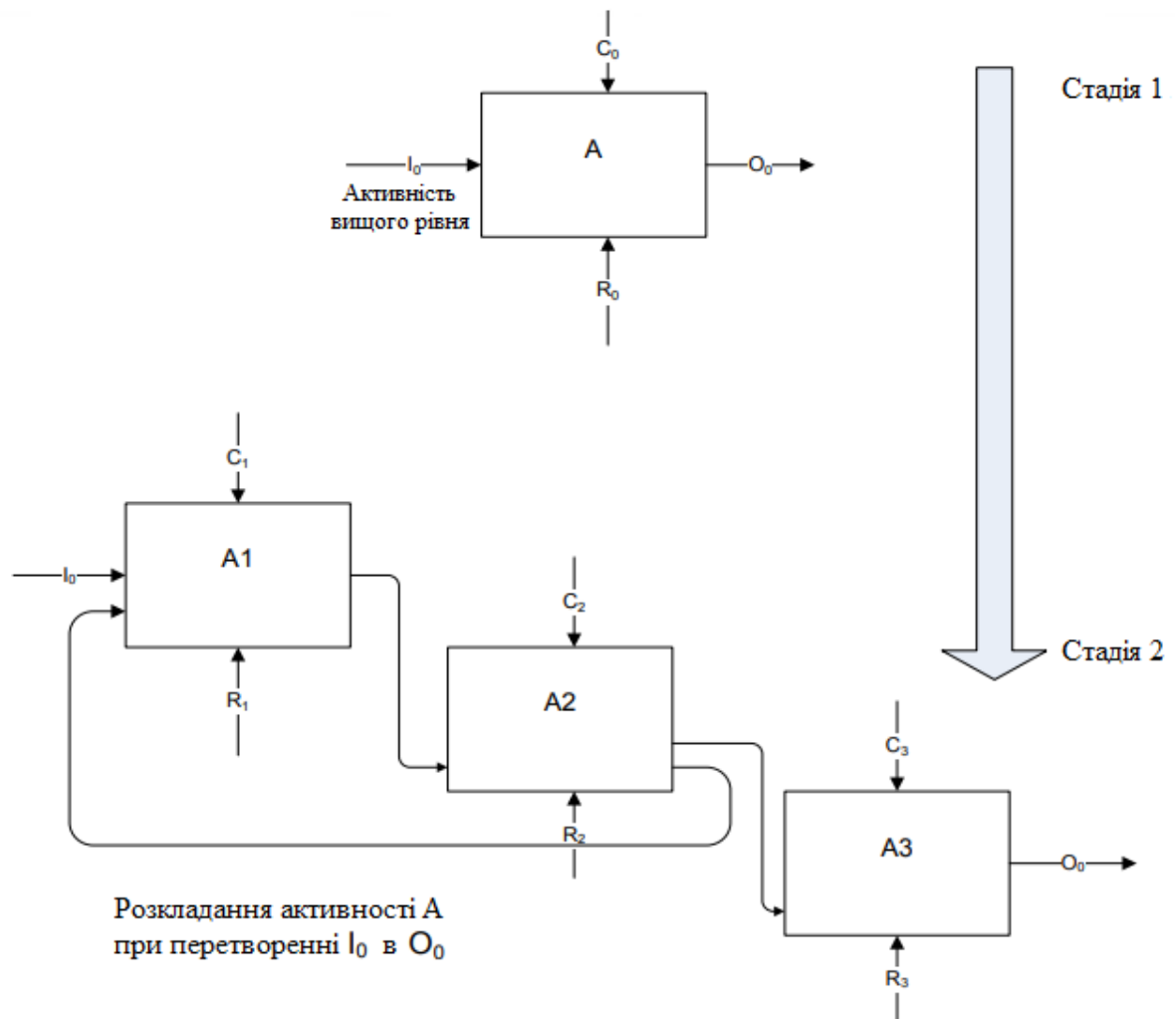


Рис. 1.3. Ієрархічна декомпозиція SADT у стилі «зверху вниз».

Представлення SADT таке:

- Основне поле, де вказується назва процесу або дії
- У лівій частині цього вікна вхідні стрілки: вхідні дані дії.
- У верхній частині вхідні стрілки: дані, необхідні для дії.
- У нижній частині вікна вхідні стрілки: засоби, що використовуються для дії.
- У правій частині вікна, вихідні стрілки: результати дії.

Семантика стрілок для діяльності:

- Вхідні дані вводяться зліва і представляють дані або витратні матеріали, необхідні для діяльності.
- Вихідні дані виходять праворуч і представляють дані або продукти, отримані в результаті діяльності.

- Елементи управління входять зверху і представляють команди або умови, які впливають на виконання діяльності, але не споживаються.
- Механізми визначають засоби, компоненти або інструменти, які використовуються для виконання діяльності. Являє собою розподіл діяльності.

Семантика стрілок для даних:

- Вхідні дані - це дії, які виробляють дані.
- Виходи споживають дані.
- Управління впливає на внутрішній стан даних.

Згідно з Myloropoulos(2004), у процесі розробки програмного забезпечення можна чи потрібно виділити кілька ролей:

- Автор або розробник моделей SADT
- Коментатори, які оглядають творчість автора
- Читачі або користувачі моделей SADT
- Експерти, які можуть порадити авторів
- Детально технічний комітет або рецензенти моделей SADT
- Бібліотекар проекту, який керує проектною документацією
- Керівник проекту, який керує системним аналізом та проектуванням.
- Монітор або головний аналітик для допомоги розробникам і користувачам SADT
- Інструктор для навчання розробників і користувачів SADT

Сильні сторони SADT:

- Точна та зведена схема для представлення графічно складних системних вимог.
- Позначення легко зрозуміти, і вимоги можна обговорювати на будь -якому рівні, придатному для замовника.
- Чітке розкладання зверху вниз для вводу-виводу та управління.

- Уникає захащення шляхом відокремлення даних від діяльності.
- Відрізняє контрольні дані від простих вхідних даних.

Слабкі сторони SADT:

- Коли є багато діаграм, розташованих в ієрархічному порядку, це може бути важко зрозуміти через додаткову контрольну інформацію на діаграмах.

Відповідні сфери застосування

- Великі або складні системи, орієнтовані на процеси.
- Навколишнє середовище, що вимагає безлічі дизайнерів.
- Корисно для зображення дій у процесі розробки програмного забезпечення.
- План військової мобілізації. SADT також відомий у Міністерстві оборони США як IDEF0.

SADT-технології мають дві основні переваги для вирішення науково-технічних класів задач. Вони дозволяють розробляти технології проектування технологічних процесів, а також використовувати отримані результати для проектування технологічних процесів виробництва інших деталей. Метою технології, що забезпечується за допомогою SADT-підходу, є проектування багатofакторних технологічних процесів, оптимізованих для різних параметрів якості.

Метод структурного аналізу є пріоритетним методом для розробки та дослідження технічних систем у складній виробничо-технологічній системі.

Різні рівні аналізу проектованої системи виділяються методом SADT-діаграм. Еталізація процесів забезпечується під час зниження рівнів із збереженням еквівалентності попереднього рівня. Аналізується технологічне середовище разом із оглядом системи та послідовною деталізацією структури системи. Розробка кількох моделей із взаємозв'язками між ними детально описує повний опис системи. Кожна модель має графічний та текстовий опис. Об'єктом аналізу, який

зображений прямокутником на схемі, є або - спроектована система (на верхньому рівні досліджень), або її частина (на нижньому рівні). Навколишнє середовище представлено стрілками (введення, вихід, управління та механізм), спрямованими до прямокутника або з нього.

Метод SADT поряд з іншими універсальними методами лише рекомендує, як організувати та забезпечити процес аналізу. Він не дає жодних рекомендацій щодо того, як розділити/розділити об'єкт на частини. Ця особливість SADT тісно пов'язана з особливостями кожної предметної галузі і вимагає ґрунтового знання не лише методології SADT, а й суті проблеми, що розглядається. Для кожної предметної галузі слід розробити конкретні рекомендації, які зменшують творчу працю, одночасно підвищуючи продуктивність праці при проектуванні чи аналізі системи на етапі технологічної підготовки.

Приклад

Діаграма, наведена на рис. 1.4, описує функції, пов'язані з отриманням персоналу для великого виробника автомобілів. Діаграма описує як автоматизовану, так і ручну діяльність та встановлює вимоги до системи відстеження додатків. У більш широкому контексті ця діаграма може бути однією функцією в загальній системі персоналу.

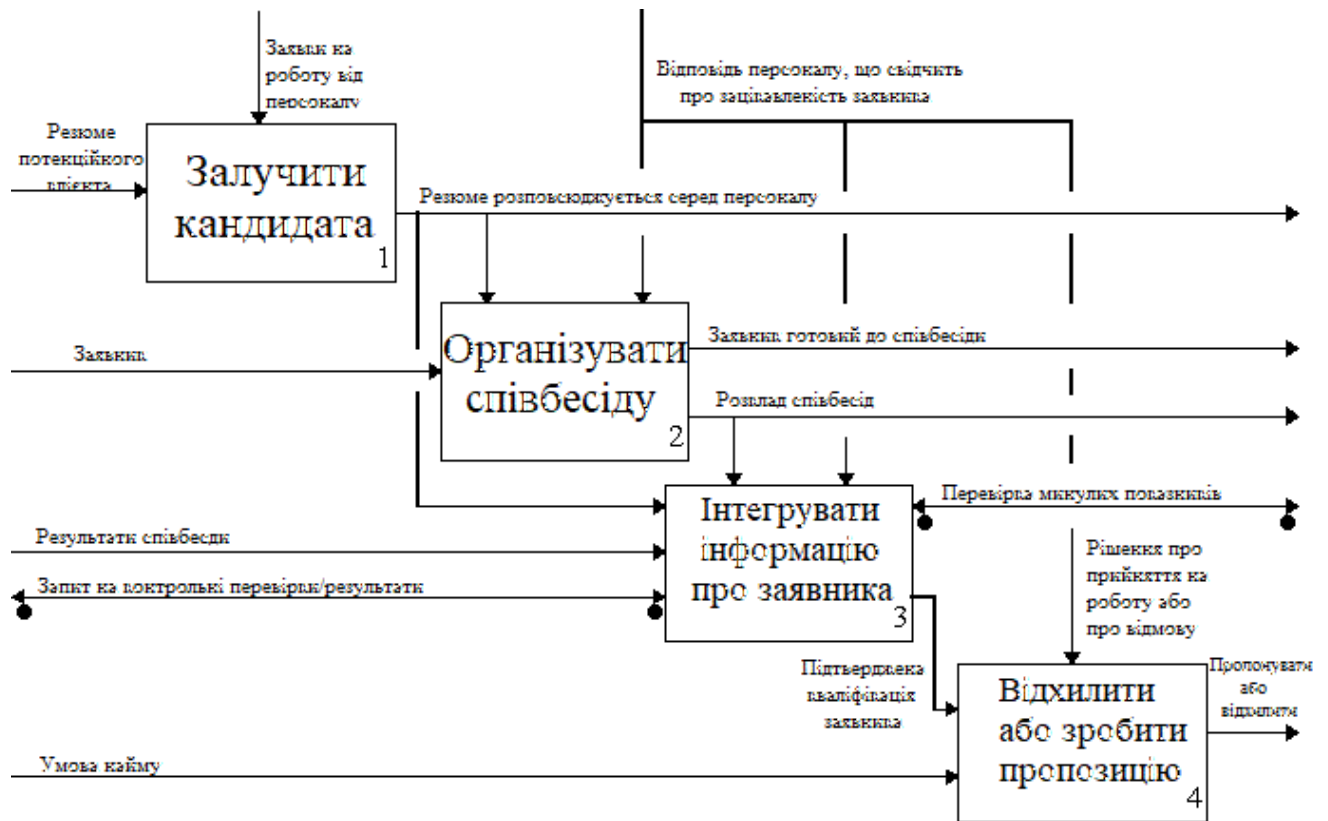


Рис. 1.4. Зразок діаграми діяльності SADT.

Пов'язані інструменти та методології

Методологія SADT включає автоматизовані інструменти для підтримки процедур аналізу та чітко визначений організаційний інструментарій, за допомогою якого інструменти застосовуються. Огляди та етапи визначаються, що дозволяє перевірити зв'язок розробника з клієнтом.

Інженерія автоматичного управління ORCHIS - це графічне та інтерактивне середовище, призначене для реалізації методу IDEF0. Це набір графічних та текстових редакторів, інтегрованих у віконну систему. Доступ до команди здійснюється шляхом випадаючого та спливаючого меню.

Розробка програми відбувається в 3 етапи:

- Вказання системи з автоматичним управлінням синтаксисом у графічному редакторі. Ви формалізуєте поведінку вашої ієрархічної системи за допомогою блоків активності, з'єднаних між собою стрілками. Також у вас є редактор міток, редактор визначення та текстовий редактор.

- Перевірити узгодженість модельованої системи відповідно до стандарту Idef-0. Засіб перевірки сканує вибраний об'єкт Idef-0, вказує на невідповідності та показує зустрічені помилки.
- Написання документації. Він складає оглядовий документ, де ви можете визначити титульну сторінку, обрати у резюме поля діяльності, які потрібно роздрукувати, і вибрати друк графічних та/або текстових частин.

1.2. Методи проектування технологічних процесів складання

Планування складання можна розділити на три основні напрямки, як показано на рисунку 1.5:

1. Планування послідовності складання.
2. Планування прийняття рішень складання.
3. Планування на основі економічних та екологічних факторів.

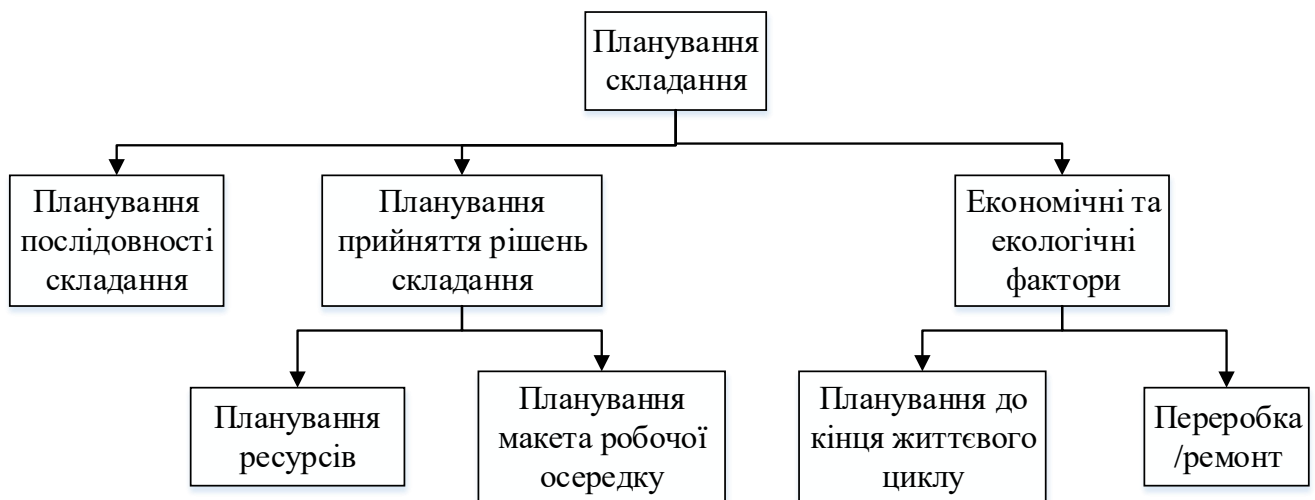


Рис.1.5. Різні аспекти планування складання.

Серед багатьох витрат, пов'язаних із плануванням складання, генерація послідовності складання є основним внеском. Послідовність операцій сполучення, що виконується для складання групи компонентів в остаточну стабільну модель, відома як планування послідовності складання. Це процес, який забирає більшість часу та зусиль на розробку продукту.

Оскільки кількість деталей стає більшою, кількість можливих послідовностей збільшується в геометричній прогресії, і без попереднього

знання найкращих послідовностей складання було б дуже важко вибрати оптимальну послідовність складання.

Планування складання також враховує варіанти закінчення терміну служби продукту. Щоб максимізувати вигоду від ремонту та переробки, необхідно вибрати операції складання та залучити матеріали таким чином, щоб кінцевий продукт був економічним та екологічно ефективним протягом усього його життєвого циклу.

Хоча в останні роки широко досліджувалися різні галузі, дослідження в автоматизації генерації послідовності складання є рідкісними, що вимагає подальших досліджень.

Проблему генерації послідовності складання можна спростити, якщо кількість частин у моделі буде меншою, а компоненти можна згрупувати разом як підскладки для створення меншої кількості послідовностей. Підвузли можна визначити як групу компонентів, які є стабільними і можуть бути складені/розкладені разом в основну модель.

Перевагами групування компонентів у вузли є

1. Час складання буде скорочено, оскільки операції складання можна виконувати паралельно.
2. Час обчислень буде скорочено, оскільки буде менша кількість деталей, з якими потрібно мати справу.

Автоматизовані планувальники складання знайдуть можливу послідовність складання, враховуючи геометричні та фізичні обмеження. Після того, як генерація послідовності складання автоматизована, необхідно знайти оптимальну послідовність складання, щоб заощадити час і витрати. Оптимальна послідовність складання – це послідовність, яка має мінімальну кількість переорієнтацій, мінімальну кількість змін інструменту та захвату тощо. Хоча теоретично, вичерпні методи пошуку можуть гарантувати оптимальне рішення, це неможливо через складність обчислень. Зі збільшенням кількості деталей складність обчислень зростає в геометричній прогресії. Дослідники вивчили різні методи вирішення цієї проблеми,

найефективніші з них, де застосовуються евристичні правила. Дві протилежні цілі, які повинні вирішити ці нові методи, — це зниження складності обчислень і збереження всіх дійсних послідовностей.

Проектування послідовності складання є одним із важливих компонентів у плануванні складання. Складання відноситься до задачі, для якої технологи, на основі своєї евристики при складанні всіх компонентів продукту, організовують конкретну послідовність складання відповідно до опису конструкції продукту [7].

Проектування послідовності складання є складною комбінаторною задачею [8] де простір розв'язків надмірно збільшується при збільшенні кількості компонентів. Розглянемо виріб з шістьма компонентами, які можна зібрати в будь-якій послідовності. У цьому випадку кількість можливих рішень для цього задається як $s = 6!$ що дорівнює 720 рішенням. Коли кількість компонентів зростає до семи, можливі рішення для продукту надмірно збільшилася до $7! = 5040$. Крім того, у справжній задачі складання існують деякі обмеження, які необхідно враховувати при генеруванні послідовностей.

Попередні дослідження показують, що дослідники запропонували та використали багато підходів для представлення проблеми складання [9, 10, 11]. Найпомітнішим способом представлення проблеми складання є використання методу орієнтованого графа. Модель складання можна описати орієнтованим графом $D = (P, C)$. P — скінченна непорожня множина вершин, а C — множина ребер, що їх з'єднують. Кожна вершина представляє компонент, а кожне ребро представляє відношення між двома компонентами. У деяких випадках вершини та ребра приносять додаткову інформацію, таку як орієнтація складання, інструмент, тип складання та час складання.

У представленій моделі орієнтований граф відомий як діаграма пріоритету, оскільки граф являє собою відношення пріоритетності складання [12]. На рисунку 1.6 показана діаграма пріоритету складання з

додатковою інформацією про модель складання. На цій діаграмі вершини представляють компонентами моделі. Тим часом інформація T1, T2, T3 і T4 представляє інструменти складання, а (+x, -x, +y, -y, +z, -z) представляє напрямок складання для певного компонента. Тому, коли встановлена послідовність складання, її можна оцінити на основі інформації в діаграмі пріоритету складання.

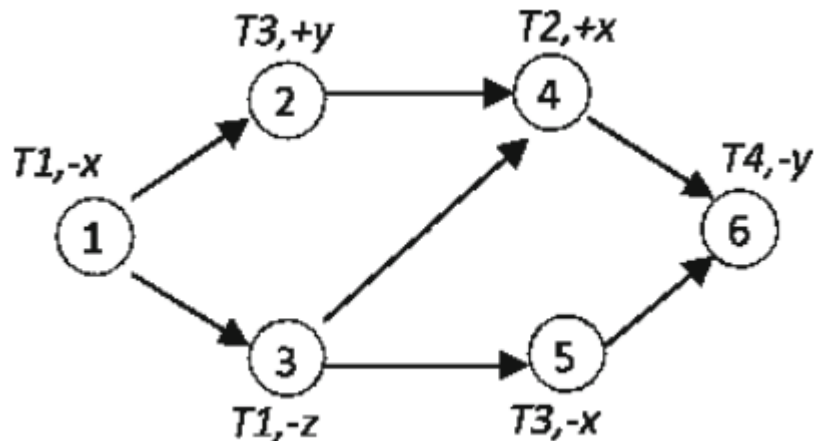


Рис.1.6. Діаграма пріоритетності складання з додатковою інформацією.

Обмеження

Існує два типи обмежень у моделі складання, це «абсолютні обмеження» та «обмеження оптимізації». Абсолютні обмеження відносяться до обмежень, які, якщо їх порушити, призводять до нездійсненної послідовності складання. Тим часом обмеження оптимізації – це обмеження, які при порушенні призводять до зниження якості послідовностей складання.

У контексті проектування послідовності складання абсолютні обмеження, які зазвичай розглядаються, включають пріоритет і геометричні обмеження. Обмеження пріоритету показує співвідношення попередніх і наступних компонентів для процесу складання. Обмеження пріоритету не може бути порушено, інакше буде створена неможлива послідовність складання. Обмеження пріоритету може бути представлено на діаграмі пріоритету (рис.1.6) або у вигляді матриці. У таблиці 1.1. представлена матриця пріоритету для діаграми пріоритету на рис.1.6. У цій матриці

частина i повинна бути зібрана після частини j , $P(i, j) = 1$. В іншому випадку матриця залишиться порожньою [10].

Між тим, геометричне обмеження в моделі стосується складання компонентів без будь-яких зіткнень. При спарюванні двох частин повинна бути принаймні одна траєкторія без зіткнень, що дозволяє привести компоненти в контакт. Усі дійсні послідовності складання повинні відповідати геометричним обмеженням для даної конструкції. Матрицю використовують для опису геометричних обмежень між компонентами в моделі складання [11]. Для кожної пари компонентів (P_i, P_j) матриця записує напрямки, в яких P_i можна зібрати без зіткнення з P_j . Тоді набір дійсних напрямків складання для кожного (P_i, P_j) визначається як рухомий клин P_i щодо P_j , що позначається $MW(P_i, P_j)$. Вони обчислюють рухомі клини для всіх пар компонентів і зберігають усі рухомі клини в матриці MV .

Таблиця 1.1. Матриця пріоритету (P) для Рис. 1.6.

P_{ij}	1	2	3	4	5
1					
2	1				
3	1				
4	1	1	1		
5	1		1		
6	1	1	1	1	1

З іншого боку, обмеження, які класифікуються як обмеження оптимізації, пов'язані з цілями оптимізації задачі. Обмеження, які класифікуються в цій категорії, включають обмеження інструменту складання, обмеження напрямку складання та обмеження стабільності складання.

Матрицю інструментів $TM = [t_{ij}]_{n \times m}$ використовують для представлення обмеження інструменту [12]. Де n – кількість деталей, а m – кількість практичних інструментів для складання відповідної деталі. Після створення оптимальної або близької до оптимальної послідовності

складання відповідні інструменти також підтверджуються, і в той же час можна отримати кількість змін (nt) інструментів складання.

Тим часом обмеження напрямку складання представлено як індекс штрафу. Коли наступний напрямок складання відрізняється від поточного, штраф буде накладено на цю конкретну послідовність складання.

Обмеження стабільності визначається коли складові частини зберігають відносне положення і вони не розривають контакт під час складання. Стабільність, посилену з'єднувальними з'єднувачами, ділять на три рівні відповідно до міцності з'єднання: міцний, слабкий і нестійкий зв'язок. У цьому підході міцному з'єднанню буде присвоєно індекс «0», слабкому з'єднанню – індекс «1», а нестабільному – індекс «2».

Цілі оптимізації

Протягом останніх 10 років для оптимізації проблеми проектування послідовності складання використовувалися різні цілі та методи м'яких обчислень. Найпопулярніша мета проектування послідовності складання - мінімізувати кількість змін напрямку складання. У цій цілі відповідні напрямки складання розташовані вздовж трьох основних осей (+x, -x; +y, -y; +z, -z). Коли напрямок наступної складальної частини відрізняється від поточного напрямку, нараховується штраф відповідно до величини зміни напрямку. Оптимальна послідовність відповідно до цієї мети матиме мінімальне покарання, спричинене зміною напрямку.

Друга за популярністю мета проектування послідовності складання — мінімізувати кількість змін інструментів. При складанні час зміни інструменту вважається непродуктивною діяльністю і може зайняти багато часу, якщо ним не керувати належним чином [13]. У цьому випадку, коли наступний процес складання вимагає іншого інструменту з поточним процесом складання, буде нараховано штраф. Найбільш оптимальною послідовністю, що відповідає цій цілі, є послідовність з мінімальним штрафом за зміну інструменту.

Крім цих двох найбільш частих цілей є ще мета проектування послідовності складання для мінімізації зміни типу моделі складання. Ця мета враховує зміни фізичних характеристик складання, такі як з'єднання, вирівнювання, загвинчування, повернення тощо.

Інші цілі полягають у тому, щоб мінімізувати складність складання, мінімізувати подібність з'єднувачів, максимізувати стабільність складання та мінімізувати шкоду від геометричних обмежень. У проектування послідовності складання деякі дослідження стверджують, що геометричне обмеження є обов'язковим обмеженням. Якщо згенерована послідовність складання не збігається з геометричним обмеженням, послідовність складання не оцінюватиметься. Таким чином, цей атрибут не вважається об'єктивним. Однак у невеликій кількості досліджень геометричне обмеження оголошується як одна з цілей проектування послідовності складання. Якщо геометрія складання нездійсненна, великий індекс штрафу буде надано функції придатності. Таким чином, нездійсненна послідовність не буде виглядати як оптимальна послідовність складання в кінцевих результатах.

Проектування складання включає такі теми, як розподіл ресурсів, компонування робочої комірки, проблеми, пов'язані з допуском, генерація послідовності складання. З усіх додаткових проблем у плануванні складання найважливішою проблемою є послідовність складання. Різні рівні деталізації моделі складання можуть бути складовими, деталями, елементами та межовими представленнями. Усі дослідження, які вже були проведені в області генерації послідовності складання, можна широко класифікувати на дві категорії на основі знання відношень пріоритету всіх частин у моделі. Два підходи є

1. Комбінаторний підхід
2. Геометричний підхід

1.2.1. Комбінаторний підхід

У комбінаторному підході послідовності складання представлені кількома способами. Після того, як послідовності складання представлені, до них можна застосувати алгоритми оптимізації, щоб знайти оптимальну послідовність складання. Представлення моделей складання в основному базуються на частинах і на їх складі. Частини можуть бути представлені мітками, вузлами, векторними елементами тощо. Складальні одиниці можна описати як вузли, списки, векторні графіки тощо. Де Мелло та Андерсон [14] запропонували, що представлення послідовності складання можна розділити на явні та неявні. Ці уявлення вимагають введення графа зв'язку, вузли якого є деталями в моделі, а ребра є контактами між деталями. Мантріпрагада і Вітні [15] запропонували розширену версію графа зв'язку, який фіксує розмірні обмеження між одним або кількома ступенями свободи між частинами. Де Мелло та Сандерсон [16] також запропонували інший варіант діаграми зв'язку, де вони додали інформацію про тип контакту (на основі еометрії) та кріплення (клей, гвинт, фіксація під тиском). Уолтер [17] представив єдину послідовність складання частковим деревом складання, яке є бінарним деревом, кореневий вузол якого є кінцевою моделлю, а листові вузли є окремими частинами. Уолтер [17] також додав додаткову інформацію, таку як упорядкування операцій, інформація про вставку та інформація про закріплення. Кілька послідовностей складання можуть бути представлені орієнтованим графом, вузли якого являють собою набір деталей, а дуги є можливими операціями складання. Де Мелло та Сандерсон, Ромні у [18], Вілсон та Ріт у [19], Томас та ін. У [20], Лі та Сайту у [21] усі використовували графіки AND/OR для представлення асемблерних послідовностей. У графі I/АБО вузли - це стабільні підвузли, а гіпердуги відповідають можливим операціям складання. Листові вузли - це окремі частини, а кореневий вузол - це ціла модель. Явні представлення моделі складання можна використовувати для визначення відносин пріоритету між з'єднаннями, а також для кодування

можливих послідовностей складання. Буржо в [22] систематично використовував відношення пріоритету між конкретними парами зв'язків. Фаціо та Уїтні в [23] використовували пріоритетні відносини між даним зв'язком та набором зв'язків. Ці відношення пріоритету з'ясовуються, запитуючи серію питань до експерта-людини. Де Мелло та Сандерсон у [15] запропонували методи для генерування відношень пріоритету з діаграм зв'язку за допомогою методів вирізання. Bonneville та ін. у [24] закодовані відношення пріоритету як геометричні обмеження, які показують відсутність траєкторій без зіткнень, щоб дозволити злиття двох складальних одиниць. Уолтер у [17] побудував операції сполучення з усіх можливих відношень пріоритету — «строго передувати», «передувати або супроводжувати», «та сама операція», «і різні операції». Naphade та ін. у [25] визначають послідовність складання як проблему виконання обмежень з точки зору умов встановлення. Хуанг і Лі в [26] розглянули вплив обмежень пріоритету не лише на завдання складання, а й на такі ресурси, як прилади та захвати. Goldwasser і Motwani в [27] розглянули деякі оптимальні критерії, такі як кількість напрямків, в яких деталей можна видалити, кількість інструментів і змін інструментів, кількість переорієнтацій, глибина послідовності складання тощо.

Основне питання, яке не розглядається у наведеній вище літературі, полягає в тому, що у всіх випадках відносини пріоритету визначаються користувачем. Цей метод пошуку відношень пріоритету робить практично неможливим створення відносин пріоритету для більших моделей через проблему вибуху. Цю проблему можна вирішити, якщо відносини пріоритету генеруються автоматично без втручання користувача.

Дослідження в комбінаторному підході передбачає знання відносин пріоритету.

1.2.2. Геометричний підхід

Другий підхід — це геометричний підхід, де відносини пріоритету вивчаються з опису моделі складання.

Вілсон та Ріт [18] запропонували підхід генерування та перевірки відносин пріоритету, але цей метод буде недоцільним для більших моделей через властивий комбінаторний вибух.

Залежно від того, як використовується інформація з моделей САПР, дослідники вибрали два різні шляхи. Перший шлях використовує просторові уявлення всіх деталей для визначення блокуючих зв'язків між усіма деталями в моделі складання. Другий шлях досліджує геометричну інформацію для планування рандомізованого шляху. Блокувальні графіки визначають напрямки, в яких усі частини можуть бути відокремлені одна від одної. Розділи простору розділяючих напрямків обчислюються в припущенні, що частини є вільно літаючими об'єктами.

Ромні та ін. [17] обчислювали локальні конуси свободи трансляції на S^2 , а потім використовували блокуючі графіки для визначення можливих послідовностей складання. Швейкард і Шварцер [28] виразили контактні відносини між частинами як лінійні обмеження для визначення можливих напрямів поступання. Але ці локальні конуси трансляційної свободи важко обчислити для частин, які мають складну геометрію, оскільки ці алгоритми будуть ефективно працювати лише проти багатогранних частин, тобто частин, які мають просту площину геометрію.

Вілсон і Латомбе [29] ввели концепцію ненаправлених блокуючих графіків для визначення розділів моделі. Це поділ трансляції на кінцеві комірки, де фіксуються відносини блокування всередині кожної клітинки. Цей підхід буде дуже складним у реалізації, оскільки він чутливий до апроксимацій з плаваючою комою. Крім того, складність зростає в геометричній прогресії з розмірами простору дозволених рухів.

Вілсон [30] розробив діаграму інтерференції для визначення шляху без зіткнень для вузла. С-простір будь-якої частини є множиною всіх можливих значень ступенів свободи. С-перешкоди — це підмножина С-простору, де частина стикається з іншими частинами. Планування шляху означає визначення шляху для кожної частини, яка міститься у С-вільній (С-простір-

С-перешкоди). La Valle в [31], Choset et al. [32] використовували локальні планувальники, щоб знайти сусідні вільні конфігурації для побудови дорожньої карти, а потім застосували пошук у графі на цій дорожній карті, щоб знайти шлях вирішення.

Одним з основних недоліків методів, які обговорювалися раніше, для генерування послідовностей складання з використанням геометричного підходу є складність цих алгоритмів у їх реалізації. Також більшість дослідницьких робіт у цій галузі зосереджено на багатогранних об'єктах. Практично об'єкти є більш складними, а не правильними багатогранниками. Тому ці алгоритми можуть працювати не ефективно для всіх практичних випадків.

1.2.3 Алгоритми оптимізації послідовності складання

Після створення можливих послідовностей складання до них можна застосувати алгоритми оптимізації для створення оптимальної послідовності складання. Нільссон у [33] використовував алгоритм A^* , який гарантує знайти шлях мінімальної вартості, тобто послідовність складання. Мартеллі та Монтанарі [34] використовували алгоритм AO^* для пошуку найкращої послідовності складання. Казеллі та Занічеллі [35], Као та Сандерсон [36] побудували асемблерні мережі Петрі з графіків I/АБО для отримання ефективних обчислень асемблерних послідовностей. Жа та ін. у [37] використані мережі Петрі на основі знань, які можуть автоматично регулювати теоретичні та реальні відхилення параметрів складання, що гарантує найкращий план послідовності складання. Бен-Ар'є та інші [38] використовувався гібрид кольорової мережі Петрі та системи на основі знань для обчислення ступеня складності кожної послідовності складання, що допомагає у виборі оптимальної послідовності. Мотаваллі та Іслам [39] використовували методику моделювання відпалу, де вони визначили енергетичну функцію з точки зору кількості переорієнтацій та загального часу складання. Вони закодували обмеження пріоритету в діаграми пріоритету та перевіряли для кожної послідовності складання. Чен [40]

закодував знання пріоритету за допомогою нейронних мереж і виразив обмеження пріоритету І/АБО між зв'язками в силі зв'язку між нейронами мережі Хопфілда. Цао і Сяо [41] використовували підхід імунної оптимізації, де проблеми послідовності складання представлені як антигени, а послідовності складання представлені як антитіла. Ван та ін. [42] застосував підхід оптимізації колоній мурах, щоб знайти оптимальні послідовності складання. Холланд [43] спочатку використовував генетичні алгоритми для пошуку високоякісних рішень складних інженерних проблем. Він довів, що кращі рішення можна отримати, застосовуючи такі генетичні оператори, як кросовер, мутація, інверсія до відомої сукупності популяції. Він також довів, що безперервно застосовуючи генетичні оператори з плином часу, можна постійно виробляти кращі рішення. Бонневіль [44] є одним із перших, хто використовує генетичні алгоритми для оптимізації створених послідовностей складання. Він використав мутацію та кросовер для обчислення оптимальної послідовності складання. Тут початкова сукупність надається користувачем. Пізніші генетичні оператори застосовуються до найкращих планів з будь-якого покоління, і генеруються нові плани. В результаті чисельність населення буде значно зростати. Алгоритм завершиться після пошуку у всьому просторі пошуку або після створення кількох хороших можливих послідовностей складання. Фудзімото [45] використовував генетичні оператори, такі як мутація, кросовер та розмноження, щоб знайти оптимальну послідовність складання. Тут генерується випадкова початкова сукупність. Послідовності батьківського складання вибираються робабілістично, а потім застосовуються оператори для створення пружини вимкнення. Вони класифікують простір рішення на сім'ї, де кожна сім'я містить єдину дійсну послідовність складання. Коли генетичні оператори застосовуються до асемблерних послідовностей, вибраних випадковим чином, недійсні нащадки замінюються відповідною сімейством дійсної асемблерної послідовності за допомогою функції перетворення. Чен у [46] використав

п'ять операторів (відтворення, кросовер, мутація, вирізати і вставити, розрив з'єднання), щоб знайти оптимальні послідовності складання для двох прикладів продуктів. Тут початкову сукупність може надати користувач або її можна вибрати випадковим чином. Чисельність населення також фіксована. Наступне покоління відбирається з найкращих нащадків і найкращих батьків. Алгоритм завершується, коли загальна придатність не покращується за кілька поколінь. Хонг і Чо [47] використовували кросовер і мутацію для створення послідовностей складання для двох механічних продуктів. Тут початкова сукупність генерується випадковим чином. Функція фітнесу базується на загальній вартості складання, а послідовності вибираються імовірно на основі значень функції фітнесу. Алгоритм завершується через певну кількість поколінь або якщо придатність не покращується протягом певного періоду часу. Сенін [48] використав кросовер і мутацію для створення планів складання для кількох тестів конструкцій. Він показав, що генетичні алгоритми, використовуючи будь-які методи виборів, знайдуть високоякісні плани складання в порівнянні з технікою перерахування. Для складних тестових випадків генетичні алгоритми дають плани низької якості в порівнянні з методами перерахування. Однак час виконання в генетичних алгоритмах набагато менше, ніж у техніці перерахування.

З наведеної вище літератури ми бачимо, що було проведено лише невелику кількість досліджень для створення послідовностей складання за допомогою даних САПР. Більшість досліджень, проведених дотепер, базуються на передумові, що відносини пріоритету відомі заздалегідь, що не стосується складних моделей. Хоча є деякі дослідницькі роботи, проведені з використанням геометричного підходу, ці алгоритми буде важко впровадити на практиці, а також вони працюють для багатограних моделей складання і не будуть ефективними, якщо їх застосовувати для складних моделей.

1.3. Огляд систем автоматизованого проектування

Система автоматизованого проектування (САПР/CAD) використовує математичну та графічну обчислювальну силу комп'ютера, щоб допомогти інженеру-механіку у створенні, модифікації, аналізі та відображенні проектів, процесів, тощо. Багато факторів сприяли тому, що технологія САПР стала необхідним інструментом у світі інженерії, наприклад, швидкість роботи комп'ютера при обробці складних рівнянь та управлінні технічними базами даних. САПР поєднує в собі характеристики інженера та комп'ютера, які найкраще підходять для процесу проектування. Існує також поєднання людської творчості з комп'ютерними технологіями, що забезпечує ефективність проектування, що зробило САПР таким популярним інструментом.

Функції системи САПР виходять далеко за межі її здатності представляти та маніпулювати графікою. Геометричне моделювання, інженерний аналіз, моделювання та передача проектної інформації також можна виконувати за допомогою САПР.

У кожній галузі інженерії до впровадження САПР проектування традиційно виконувалося вручну. Отримане креслення із суттєвими деталями потім піддавали аналізу з використанням складних математичних формул, а потім слали назад з пропозиціями щодо покращення. Була виконана та сама процедура, і через ручний характер креслення та подальший аналіз, вся ця процедура займала багато часу та трудомісткість.

Протягом багатьох десятиліть САПР дозволяла проектувальнику обійти більшість ручного складання та аналізу, які були необхідні раніше, роблячи процес проектування більш плавним та набагато ефективнішим. Корисно розуміти загальний процес розробки продукту як поетапний процес. Однак у сучасному інженерному середовищі окреслені кроки перетворилися на більш спрощений підхід, який називається паралельною інженерією. Такий підхід дозволяє командам працювати одночасно, забезпечуючи спільну основу для взаємопов'язаних завдань із розробки

продуктів. Інформація про продукт може бути легко передана між усіма процесами розвитку: проектуванням, виробництвом, маркетингом, управлінням та мережами постачальників. Паралельна інженерія визнає, що менші зміни призводять до меншої витрати часу та грошей на перехід від концепції дизайну до виробництва та від виробництва до ринку. Відповідні процеси автоматизованої інженерії (CAE), автоматизованого виробництва (CAM), комп'ютерного складання (CAA), автоматизованого тестування (CAT) та інших систем з автоматизованим управлінням стали невід'ємними частинами паралельної інженерії.

При проектуванні методів виробництва та складання використовується міждисциплінарний вхід з різних джерел (інженери-конструктори, інженери-виробники, постачальники матеріалів та обладнання та персонал цеху) для полегшення ефективного проектування виробу, який можна виготовляти, складати та продавати в найкоротші терміни. CAD, CAE, CAM, CAA та CAT - це напрямки всіх типів дизайну виробів, виготовлення форм або штампів, а також лінії виготовлення. Кількість і складність виробів, що виробляються, з кожним роком стає все більшою, але кількість досвідчених дизайнерів виробів, дизайнерів форм/штампів та виробників, як правило, не йде в ногу з цим. Відповіддю на цей дефіцит "людської сили" стало збільшення проектування за допомогою CAD/CAE/CAM/CAA/CAT.

CAD дозволяє швидко переглядати проект та дозволяє легко виконувати оцінку проекту. Точність проектування можна перевірити за допомогою автоматизованих процедур допуску та визначення розмірів, щоб зменшити ймовірність помилки. Шарування - це техніка, яка дозволяє дизайнеру накладати зображення один на одного. Це може бути дуже корисним на етапі оцінки процесу проектування, дозволяючи інженеру візуально перевіряти розміри остаточного проекту щодо розмірів етапів запропонованого виробником, гарантуючи наявність достатніх матеріалів на попередніх етапах для правильного виготовлення. САПР дозволяє

перевіряти потенційні проблеми перешкод. Ця процедура передбачає, щоб дві частини дизайну не займали один і той же простір одночасно.

Можливості автоматизованого проектування в системах САПР полегшують презентацію проекту, що є завершальним етапом процесу проектування. Дані САПР, які зберігаються в пам'яті комп'ютера, можна надсилати на плортер або інший пристрій для друку, щоб швидко та легко створити детальний малюнок. На початку САД ця функція була основним обґрунтуванням для інвестування в таку систему. Складання конвенцій, включаючи, але не обмежуючись ними, визначення розмірів, перехресне штрихування, масштабування дизайну та збільшений вигляд деталей чи інших областей проектування, можна автоматично включати майже у всі системи САПР. Детальні та складальні креслення, кошторисні матеріали та перерізи дизайнерських виробів також автоматизовані та спрощені за допомогою САПР. Крім того, більшість систем здатні автоматично представити шість видів дизайну. Складання стандартів, визначених компанією, можна запрограмувати в систему таким чином, щоб усі остаточні проекти відповідали стандарту.

Документація проекту також спрощується за допомогою САПР. Управління даними про продукт (PDM) стало важливою програмою, пов'язаною з САПР. PDM дозволяє компаніям робити дані САД доступними міжвідомчими в комп'ютерній мережі. Цей підхід має значні переваги перед традиційним управлінням даними. PDM - це не просто база даних, яка містить дані САПР як бібліотеку для зацікавлених користувачів. Системи PDM пропонують підвищену ефективність управління даними за допомогою відносин клієнт-сервер між окремими комп'ютерами та мережевим сервером. Переваги існують при впровадженні системи PDM:

- забезпечує швидший пошук файлів САПР за допомогою пошуку ключових слів та інших функцій пошуку;
- автоматизований розподіл проектів керівництву, інженерам-виробникам та працівникам цехів для огляду проекту;

- функції ведення обліку, які містять історію змін конструкції; та функції захисту даних, що обмежують рівні доступу до файлів дизайну.

PDM сприяє обміну інформацією, характерною для нових робочих місць. Оскільки компанії стикаються з посиленням тиском надавати клієнтам індивідуальні рішення для їх індивідуальних потреб, системи PDM дозволяють підвищити рівень командної роботи між персоналом на всіх рівнях проектування та виробництва продукції, скорочуючи витрати, часто пов'язані з відставанням інформації та переробкою. Незважаючи на те, що САПР зробила процес проектування менш нудним та ефективним, ніж традиційні методи, основний процес проектування в цілому залишається незмінним - він все ще вимагає людського втручання та винахідливості, щоб розпочати та пройти багато ітерацій процесу. Тим не менш, САПР настільки потужний інструмент проектування, що економить час, тому зараз важко функціонувати в конкурентному інженерному світі без такої системи.

Проектування технологічного процесу охоплює заходи та функції для підготовки детального набору планів та інструкцій щодо виготовлення деталі. Проектування починається з інженерних креслень, специфікацій, списків деталей або матеріалів та прогнозу попиту. Результати планування такі:

- Маршрутизація, яка визначає операції, послідовності операцій, робочі центри, стандарти, інструменти та пристосування. Ця маршрутизація стає важливим вкладом у систему планування виробничих ресурсів для визначення операцій з метою контролю виробничої діяльності та визначення необхідних ресурсів для цілей планування вимог до потужностей.
- Плани технологічних процесів, які зазвичай містять більш детальні, покрокові робочі інструкції, включаючи розміри, пов'язані з окремими операціями, параметри обробки, інструкції з налаштування та контрольні пункти забезпечення якості.

- Креслення виготовлення та складання для підтримки виробництва (на відміну від інженерних креслень для визначення частини).

Проектування процесу вручну ґрунтується на досвіді інженера, виробника та знаннях щодо виробничих приміщень, обладнання, їх можливостей, процесів та оснащення. Проектування технологічного процесу займає багато часу, і результати залежать від того, хто планує.

Системи САПР ТП призначені для автоматизації технологічної підготовки виробництва, автоматизованої розробки технологічної документації на деталі або складальні одиниці, а також для автоматизації нормування матеріалів та технічних процесів. САПР ТП забезпечує:

- Налаштування технологічних довідників для завдань підприємства, а при необхідності - підключення нормативної та довідкової інформації з інформаційної системи підприємства;
- налаштування технічного процесу для специфіки виробництва;
- автоматизоване нормування праці та матеріалів технічного процесу за алгоритмами, визначеними користувачами системи;
- автоматизований розрахунок режимів обробки та різання за алгоритмами, визначеними користувачами системи;
- формування вихідних форм технологічної документації.

САПР ТП підтримує різні методи проектування:

- інтерактивний дизайн;
- проектування на основі аналогового технічного процесу;
- запозичення технологічних рішень з раніше розроблених технологій;
- проектування з використанням бібліотек технологічних рішень;
- проектування типових технічних процесів;
- проектування технічного процесу з урахуванням нових даних у неінтерактивному режимі.

Усі ці інструменти забезпечують технологу максимальну продуктивність праці та значно покращують якість результатів.

При використанні САПР досягаються такі результати:

- підвищення продуктивності роботи технолога;
- зменшення кількості помилок при проектуванні;
- підвищення якості роботи технолога;
- скорочення термінів технологічної підготовки виробництва;
- підвищення ефективності оновленої інформації про розроблені технічні процеси;
- скорочення термінів та витрат на адаптацію молодих спеціалістів до конкретних умов підприємства;
- наповнення та застосування бази знань підприємства для технологічного проектування нової продукції;
- підвищення точності інформації, що використовується для нормування та визначення собівартості.

Впровадження САПР ТП допоможе підняти розвиток технічних процесів на новий якісний рівень, прискорить виробництво циклу технологічної підготовки, звільнить технологів та співпрацівників від рутинної роботи.

САПР ТП дозволяє вирішувати всі завдання технологічного проектування, які виникають у роботі промислового підприємства, завдяки дуже гнучкій конфігурації системи для будь-якого виду виробництва.

Для роботи з технологічними даними системи повністю інтегрують з функціями доступу до композицій продукції, номенклатури та довідкової інформації.

САПР ТП повністю налаштовується для специфіки виробництва та завдань підприємства, що дозволяє оптимізувати алгоритми формування технічного процесу та його стандартизацію для специфіки виробництва.

По-перше, користувач автоматизованої системи має бути висококваліфікованим технологом, оскільки система сама виконує рутинну роботу з пошуку необхідної інформації, стандартних розрахунків для оформлення технологічної документації. Технолог у режимі діалогу повинен вирішувати творчі проблеми, пов'язані із прийняттям складних логічних рішень, які система не може виконати.

По-друге, технолог повинен знати мову спілкування з САПР ТП і вміти керувати нею, включаючи аналіз помилок, що виникають у системі, та шляхи їх усунення.

По-третє, у технолога є нова функція підтримки автоматизованої системи. Ця функція складається з:

- оновлення баз даних (знань) та в поповненні їх новою інформацією;
- пошук та усунення алгоритмічних помилок;
- розробка алгоритмів подальшого вдосконалення САПР ТП.

Цей же механізм дає змогу розробляти такі технології, які одночасно включають операції механічної обробки, штампування, термічної обробки, покриттів тощо. Підключення нового технологічного перерозподілу здійснює сам користувач.

Об'єктивний хід розвитку технології, технології засобів автоматизації робить вирішення поставлених задач автоматизації проектування надзвичайно актуальним, що визначає основні напрямки вдосконалення САПР.

Незважаючи на великий вибір систем САПР ТП на ринку, вони все ще потребують значного перегляду та вдосконалення, перш за все, пов'язаного зі збільшенням рівня автоматизації. Можна виділити наступні сфери вдосконалення:

- 1) Удосконалення процесу формування технологічних рішень.

Проектне технологічне рішення - це опис технологічних об'єктів та їх взаємодій, що забезпечують досягнення заданої сукупності

результатів та їх значень, що відповідають фіксованому набору умов функціонування технологічного об'єкта.

- 2) Автоматизація вибору технологічних основ. Вибір технологічних основ - найважливіший етап проектування ТП. У сучасних СПАР ТП він практично не автоматизований - прийняття необхідного рішення повністю відповідає за користувача.
- 3) Передбачення якості товару. Основною метою проектування ТП є забезпечення досягнення заданої якості продукції під час її впровадження. Жодна з існуючих САПР ТП не має можливості передбачити очікувану якість продукту при впровадженні дизайнерських рішень, сформованих за його допомогою. Передбачення якостей, але не виконане та з неавтоматизованим проектуванням ТП.
- 4) Спрямоване формування властивостей продукту. Автоматизація синтезу структур маршрутних ТП, наприклад, виготовлення деталей машин, повністю реалізує концепцію спрямованого формування властивостей продукції.
- 5) Інтеграція САПР ТП у системи підтримки та управління життєвим циклом продукту. Проблема поєднання автоматизованих систем, які підтримують окремі етапи життєвого циклу, в інтегровану системну підтримку та управління життєвим циклом є надзвичайно актуальною.
- 6) Впровадження нових методологій проектування. Необхідність забезпечення високоякісного проектування та технологічного проектування складних виробів при скороченні часу на проектування призвела до розробки нових методологій проектування.

ВИСНОВКИ ДО РОЗДІЛУ

У розділі розглянуто основи методологія SADT та сучасні методи проектування технологічних процесів складання.

SADT - це інструмент для структуризації інформації та дозволяє інженерам краще зрозуміти проблему, яку потрібно вирішити. Тому SADT в основному використовується для встановлення мислення та структурування фаз аналізу та проектування.

SADT-технології мають дві основні переваги для вирішення науково-технічних класів задач. Вони дозволяють розробляти технології проектування технологічних процесів, а також використовувати отримані результати для проектування технологічних процесів виробництва інших деталей. Тому методологія найкраще підходить для проектування процесів складання.

Проектування послідовності складання є одним із важливих компонентів у проектування процесів складання. Складання відноситься до задачі, для якої технологи, на основі своєї евристики при складанні всіх компонентів продукту, організовують конкретну послідовність складання відповідно до опису конструкції продукту. Тому постає задача в автоматизації проектування процесів складання. Методи для автоматизації було розглянуто в розділі.

У наступному розділі розглядається системне моделювання процесі автоматизованого технологічного проектування.

РОЗДІЛ 2. СИСТЕМНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ АВТОМАТИЗОВАНОГО ТЕХНОЛОГІЧНОГО ПРОЕКТУВАННЯ

Перехід до сучасних форм організації виробництва та необхідність випуску конкурентоспроможної продукції потребує пошуку нових закономірностей виконання технологій, що враховують як конструкторсько-технологічні властивості виробу, і властивості технологічного оснащення. Заключним етапом виробничого процесу в машинобудуванні є складання, тому якість готових виробів багато в чому залежить від її технології.

2.1 Математичні моделі у задачах аналізу і синтезу процесів складання

Зі зменшенням тривалості виробництва зростає продуктивність роботи, зменшується собівартість готових виробів, прискорюється оборот коштів, вкладених у підприємство. Отже, проектування такої послідовності складання, що призводить до скорочення виробничого циклу за часом, є актуальним завданням. Побудова раціональної послідовності складання є конструкторсько-технологічним завданням. Ефективність її вирішення залежить від використання комп'ютерних технологій та вимагає формалізованого математичного опису для розробки відповідного алгоритмічного та програмного забезпечення.

2.1.1 Математична модель складального виробу

При інформаційній підтримці життєвого циклу складального виробу виникають деякі завдання, розв'язати які можна проаналізувавши певні властивості конструкції виробу. Тому є бажання створити єдину та універсальну по формі уявлення математичну модель, яка буде придатна для вирішення більшості завдань. У таких завданнях є декілька сутностей, і ключову роль в них грає поняття зв'язку. Зрозуміло, що в якості таких сутностей виступають властивості реальних об'єктів (наприклад, деталі виробу, технологічне оснащення, обладнання тощо), а зв'язки – асоціація

між окремими властивостями згадуваних об'єктів. У математиці такі зв'язки прийнято називати відносинами і представляти як декартів добуток множин. Наприклад, якщо об'єктами вважати деталі конструкції, то відношеннями між ними (скажімо, властивість наявності контакту) є підмножина впорядкованих пар $R \subset A_1 \times A_2 \times \dots$, де $a_i \in A_1$, $a_j \in A_2$ і так далі – номери деталей виробу.

Щодо ступеня відношення. Якщо n – загальна кількість деталей, то, має виконуватися рівність $A_1 = A_2 = \dots = A_n = A$. Однак розглядати $R \subset A_n$ не доцільно, тому що будь-який процес складання можна зробити детальним (установка по одній деталі на кожній операції). Це, в свою чергу, дозволяє дійти невтішного висновку, що основна властивість такого уявлення – „ a_i слідує за a_j ” – характеризує зв'язок між парою деталей, тобто бінарне відношення $R \subset A^2 = A \times A$.

Власне, для теоретичного розгляду математичної моделі складального виробу достатнім є її опис в термінах теорії множин. Проте на практиці реалізація САПР систем вимагає від розробників математичного забезпечення докладної форми представлення даних.

Логічним є питання щодо змісту чи семантики стосунків. Є дві властивості: контакт і слідування. Однак при автоматизованому проектуванні основною вихідною інформацією є геометрична інформація про топологію поверхонь деталей, яка може бути отримана з САПР, а також властивість проходження, яка не присутня в 3D-моделях у явному вигляді. Тому властивість обмеження переміщення (рухливості), на основі якої робитиметься висновок про черговість установки деталей, має бути формальною. Отже, надалі вираз $(a_i, a_j) \in R$ означатиме, що вислів „деталь a_i обмежує можливе переміщення a_j ” – істинний.

Способи перевірки умов доступу є своєрідним інструментом для формування математичного подання геометричної складової Р моделі складального виробу. Зазвичай це стосується завдань систем комп'ютерної підтримки виробництва.

Математичне уявлення параметрів просторового розміщення деталей досить різноманітне, проте їх загальною особливістю є врахування лише властивості конкретної конструкції.

Пропонується математичну модель подати як поєднання функцій [51]:

$$F_G: R \times T \rightarrow \Theta \quad (2.1)$$

– деталь a_i обмежує можливе переміщення a_j по траєкторії $t \in T$ на відстані $\Theta \in \Theta$ (тут під t слід розуміти довільну криву, як найпростіший і найчастіший випадок – поступальне переміщення вздовж координатного напрямку або поворот навколо осі);

$$F_P: R \times H \rightarrow \Omega \quad (2.2)$$

– з'єднання a_i та a_j належить певному типу сполук H з відомою характеристикою $\omega \in \Omega$;

$$F_E: R \times O \rightarrow Z \quad (2.3)$$

– з'єднання a_i і a_j може бути отримане на устаткуванні $o \in O$ з фізичним(технологічним) принципами функціонування $\zeta \in Z$.

Функція F_G задає відповідність між безліччю деталей та траєкторій їх переміщень. Загалом це n відношень, оскільки траєкторій може бути безліч.

Зазвичай множина T містить до 6 елементів, яким відповідатиме 6 матриць поворотів і переміщень: $M_x, M_y, \dots, R_z$. Важливо зазначити, що ці матриці можна сформувати не тільки вручну, що непродуктивно і матриці можуть містити багато помилок, а й автоматично через аналіз 3-Д моделі, створеної в САПР, із забезпеченням повної інформативності [52].

Для загального аналізу буде достатньо мати значення $\Theta \in \Theta$ функції F_G як логічне значення відсутності чи наявності тієї чи іншої властивості. Для більш детального аналізу, наприклад розмірного аналізу, можна під значенням Θ розуміти розмірну інформацію [53].

Зміст функції F_P повинен відображати, в першу чергу, фізичний принцип функціонування з'єднання та, в другу чергу, підлегло-конструктивне виконання з'єднання. Така постановка пояснюється тим, що фізичний принцип функціонування з'єднання, як правило, є незмінним, тоді як конструктивна реалізація з'єднання в процесі конструювання та

відпрацювання конструкції приладу на технологічність може зазнавати істотних змін.

Зміст функції F_E повинен складатися, насамперед, з фізичних чи технологічних принципів реалізації з'єднання і, у другу чергу, з підпорядковано-технологічного обладнання, яке реалізує цей принцип.

2.1.2 Математична модель процесу проектування технології

Маніпулювання моделями виробів – прерогатива математичних моделей процесу проектування технології. Це найменш формалізована складова всього процесу проектування, оскільки, певною мірою, вона має відтворювати мислення людини. Тут досить добре застосовувати системний підхід. Так, наприклад, процес синтезу технологічного процесу складання (ТПС) може бути уніфікований і зведений до кількох стандартних підпроцесів подальшого опрацювання прийнятих рішень [54]. Інакше висловлюючись, задається процес Π , який відображає проектування маршрутного ТПС, а його складові можна представити як реалізацію послідовності підпроцесів Π_i , в кожному з яких автоматизовано вирішується ряд завдань:

$$\Pi \rightarrow \bigcup_{i=1}^k \Pi_i \rightarrow \begin{cases} \bigcup_{j=1}^f Pr_j^1(Q, M_j) \rightarrow G \\ \bigcup_{t=1}^n Pr_j^2(G, D, W_t) \rightarrow R \end{cases} \quad (2.4)$$

Згідно з цим твердженням(рівняння 2.4), кожен підпроцес Π_i – i -й рівень проектування, на якому виконуються процедури Pr_j^1 синтезу проектних рішень. Результатом виконання підпроцесів $\Pi_1 \dots \Pi_k$ буде множина всіх допустимих рішень. Ця множина містить тільки частину $R \subset G$ домінуючих варіантів, а тому для неприпустимості синтезу $R \cap G$ явно неперспективних варіантів на кожному рівні проектування вводяться процедури Pr_t^2 аналізу та оцінки проектних рішень. Процедури Pr_j^1

виконуються над об'єктом моделювання Q на основі множини правил M_j , а $P_{r_t}^2$ – на основі вихідних даних D та набору обмежень W_t .

Відповідно до вище сказаного, можна виділити такі підпроцеси автоматизованого проектування ТПС (рис. 2.1).

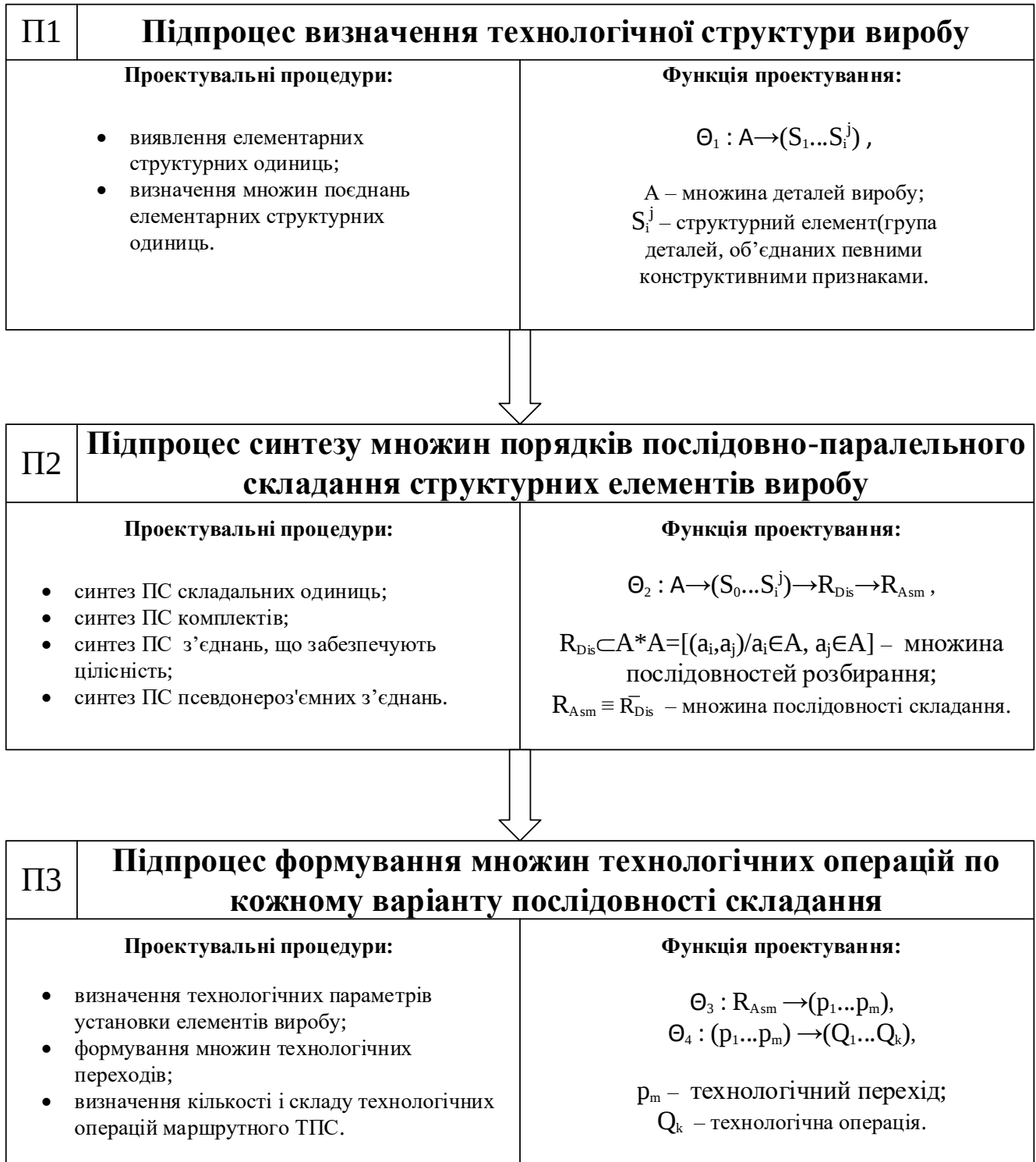


Рис.2.1. Структура процесу синтезу маршрутних ТПС виробів.

Підпроцеси $P_1...P_3$ представляють традиційне проектування, при якому попередньо зі складу виробу виділяються технологічно незалежні частини, для яких проектується окремі ТПС. Такий підхід дозволяє розглядати ТПС виробу, як сукупність процесів складання окремих його елементів, а кількість ітерацій виконання $P_1...P_3$ дорівнює кількості структурних елементів, що входять до складу виробу. Така особливість складання надає можливість обмежитися розробкою алгоритмів проектування лише одного окремого елемента, чим забезпечується універсальність алгоритмічного процесу щодо складної структури виробу.

Для математичного опису підпроцесів є два шляхи:

- імовірнісний – дозволяє з мінімальною трудомісткістю обчислень та за допомогою простих проектних процедур отримати гарні результати, проте це не дає гарантії того, що певні послідовності складання є кращими на багатьох всіх можливих альтернативах;
- логічний – базується на формалізованих методах вирішення технологічних завдань, чим забезпечує передбачуваність результатів (обґрунтованість проектних рішень). Однак через необхідність аналізу повного простору розв'язків задачі, тривалість розрахунків, при сучасному рівні розвитку обчислювальної техніки, навіть для нескладних виробів може бути значною.

З точки зору технології складання, функція F_G несе інформацію про геометричну замкнутість об'єктів, тобто вона показує елементи конструкції, які перешкоджають переміщенню i -тої деталі по необхідній траєкторії.

На основі даної властивості запропоновано метод автоматизованого синтезу послідовностей складання, який отримав назву «зворотного» [55]. Його основною перевагою є лінійна залежність кількості арифметичних операцій від кількості деталей виробу [56]. Це одна з найважливіших особливостей, що дозволяє аналізувати конструкції практично будь-якої складності (порівняно з іншими відомими методами синтезу, які дозволяють

розглядати модель до 15-20 деталей, бо складність обчислювальної задачі для них залежить від n).

Ідея визначення технічно допустимої послідовності складання полягає в тому, що для виробу, який є ієрархічним структурованим об'єктом з вкладеними один в одного складальними одиницями, спочатку моделюється процес їх розкладання, при якому гарантується виконання всіх умов доступу деталей до зони складання, після чого за допомогою інвертування можна отримати послідовність складання.

Виникає необхідність використання функції F_p , адже інформації лише про факт обмеження можливого переміщення деталей недостатньо для виявлення самих складальних одиниць. Крім наявності контакту дана функція ставить у відповідність парі деталей певний тип з'єднання. З точки зору технічної реалізації – це зовсім різні з'єднання (різьбові, пресові, зварні тощо), проте з точки зору їх подання в математичній моделі вони мають одну спільну властивість – це з'єднання, що забезпечують нероз'єдність структурної одиниці. Одне таке з'єднання, і навіть деталі, установка яких передують його створенню, визначають елементарну складальну одиницю.

Комбінуючи між собою складальні одиниці можна створити будь-який з технічно можливих складальних виробів, тобто розділяти всю конструкцію на цілісні частини за технологічними принципами (наприклад, час складання, кількістю деталей тощо).

Розробка технології складання виробу повинна виконуватися автоматизованою системою технологічної підготовки виробництва (АСТПВ). На жаль, зараз у цих системах послідовність виконання операцій зазвичай повністю задається людиною вручну, тобто розривається зв'язок між САПР та АСТПВ. У цей ланцюжок ще втручається технолог, що призводить не тільки до збільшення часу проектних робіт, але й до погіршення якості проектних рішень через неможливість оцінки їхньої повної множини та вибору найкращого варіанту, а також звичайних помилок. Саме тому системи на вирішення слабо формалізованих

технологічних завдань виділено окрему групу: САП – системи комп'ютерного проектування технологічних процесів у технологічній підготовці виробництва.

Кінцевий результат проектування в системах САП досить часто представляється у вигляді графів (або таких, які можуть бути зведені до графів) – аналоги традиційних схем складання. Проте досі не вирішено питання автоматизованого введення та опрацювання інформації технологічного змісту. Наприклад, гостро постають питання автоматизованого вибору технічних засобів; майже відсутні зв'язки між теоретичними моделями технологічних процесів складання та отриманими з дослідів моделями окремих підпроцесів орієнтування чи з'єднання деталей.

2.2 Функціональна модель автоматизованого проектування технологічного процесу складання

SADT — це методологія моделювання, що включає комбіновану графіку та текст для отримання розуміння, підтримки аналізу, забезпечення логіки для налаштування системи, визначення вимог або підтримки проектування та інтеграції системного рівня. Містить як визначення мови графічного моделювання (синтаксис і семантика), так і методологію розробки моделей [2].

SADT можна використовувати для моделювання різноманітних систем, включаючи підприємства, програмне забезпечення, апаратне забезпечення, продукти; можна використовувати для аналізу функцій системи та документування її відносних механізмів; можна використовувати для визначення вимог і специфікації функцій, а потім для проектування реалізації, що відповідає вимогам [57]. Крім графічної та текстової мови, методологія пропонує набір процедур і прийомів для розробки та інтерпретації моделей, наприклад для збору даних, побудови діаграм, циклів перегляду та документації [58].

На основі методології SADT було розроблено функціональну модель автоматизованого проектування технологічного процесу складання (рис.2.2, рис.2.3)



Рис.2.2. Контекстна діаграма.

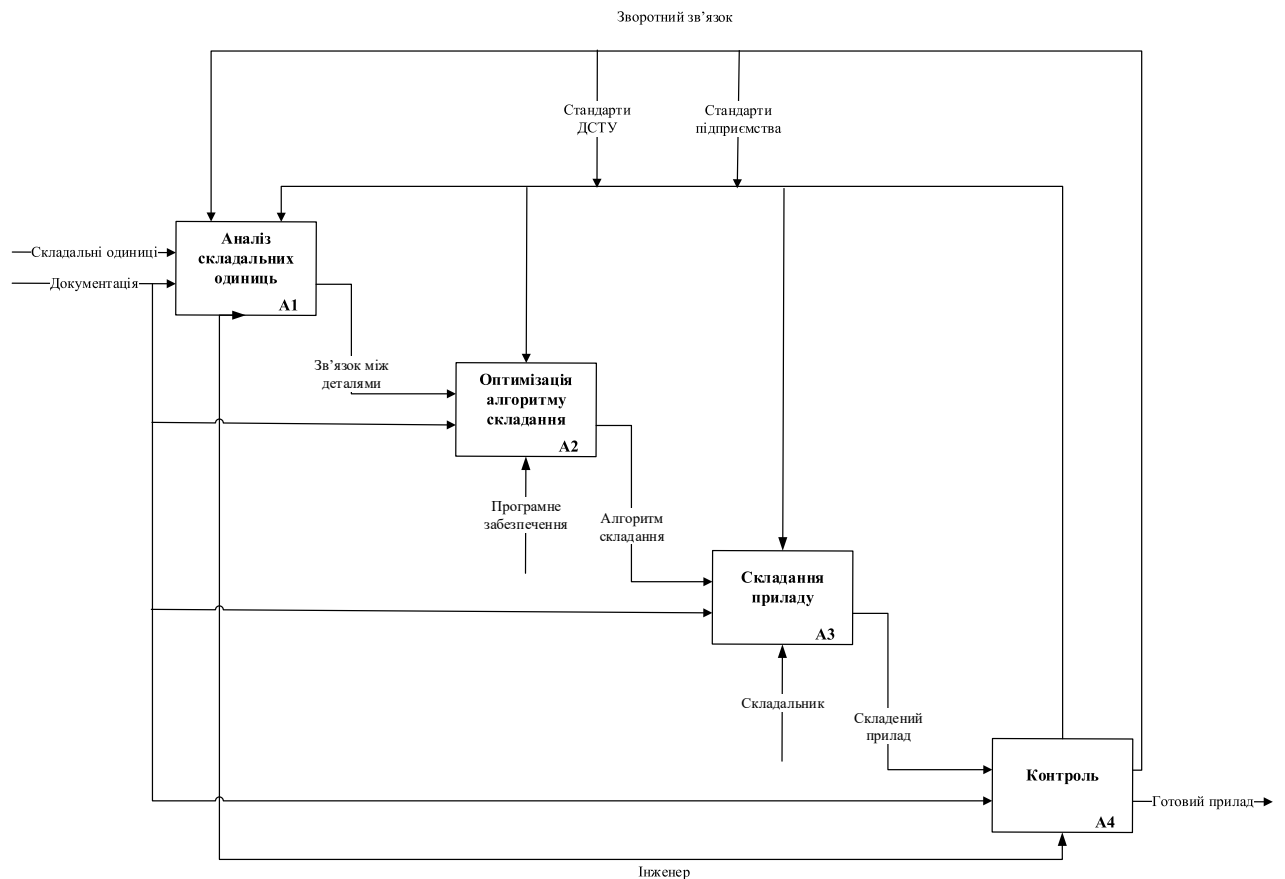


Рис.2.3. Діаграма декомпозиції.

ВИСНОВКИ ДО РОЗДІЛУ

У розділі 2 було розглянуто математичну та функціональну моделі автоматизованого проектування технологічного процесу складання.

Ефективна реалізація комп'ютерно-інтегрованого складального виробництва повинна спиратися на сукупність математичних моделей складального виробу, технології складання та виробництва, кожна з яких повинна уточнювати попередню. Модель виробу, як засіб представлення віртуального об'єкта виробництва, повинна враховувати всі значущі властивості та функції матеріального об'єкта. Модель технології має формально визначати технологічні можливості та спосіб виготовлення матеріального об'єкта. Модель виробництва має формально враховувати технологічні можливості та реальне завантаження технологічного обладнання. Усі математичні моделі повинні мати сумісну, а у кращому разі, єдину математичну основу, що дозволить виключити помилки трансформації та передачі даних, суттєво прискорить процес проектування технології та організації виробництва та гарантувати його коректність.

На основі методології SADT було розроблено функціональну модель автоматизованого проектування складання, модель складається з двох діаграм: контекстної діаграми та діаграми декомпозиції. Контекстна діаграма є представленням функціональної моделі, а діаграма декомпозиції – розширення для контекстної діаграми.

У наступному розділі розглядається модель роботи системи автоматизованого проектування складання.

РОЗДІЛ 3. МАТЕМАТИЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ВИРОБНИЧИХ ПРОЦЕСІВ СКЛАДАННЯ

САПР – це технологія, яка використовується для створення математичної моделі деталей виробу. Технологія включає рівняння геометричних елементів, таких як вершини, ребра, поверхні, взаємне розташування однієї частини з іншою частиною в моделі тощо. Таке програмне забезпечення САПР розробляється багатьма галузями, наприклад CATIA розробляється Dassault Systems, UG NX розробляється Siemens і т. д. Кожен розробник програмного забезпечення САПР постійно впроваджує інновації та створює корисні способи збагачення деталей продукту. Така практика робить дані САПР багатим джерелом інформації про продукт, але не має стандартизованого способу обміну цією багатою інформацією. Наприклад, обмеження складання, специфікація механізмів, результати моделювання тощо не можуть бути спільною між різними програмними засобами САПР. Усе програмне забезпечення САПР дотримується деяких «нейтральних форматів файлів», таких як «.step», «.iges» тощо, щоб поділитися деталями продукту один з одним. Щоб зробити підхід корисним для всіх доступних пакетів САПР, важливо, щоб був обраний узагальнений набір даних, який можна зазвичай витягувати з усіх форматів. Формати даних САПР різні для різних програм. Така інформація, як дані обмежень, специфікація механізмів та інша інформація, орієнтована на застосування, є специфічною для пакету програм САПР. Така інформація про пакет втрачається, якщо хтось використовує невідповідне програмне забезпечення. Отже, у цьому дослідженні розглядаються лише ті характеристики інформації про продукт, якими можна поділитися за допомогою «нейтральних форматів файлів». У цьому підрозділі аналізується література про підходи до планування послідовності складання на основі САПР з метою визначення корисного набору даних, який можна витягти з даних САПР для планування послідовності складання.

Модель складання можна представити так само просто, як просто список компонентів, включених до продукту, наприклад, специфікацію матеріалів, або настільки ж детально, як математичні співвідношення геометрії, наприклад САПР. Кожен тип представлення використовується для певної мети. Планування складання є однією з областей, де складання відіграє життєво важливу роль у допомозі експертам з планування складання. Програмне забезпечення САПР обіцяє геометрично правильне представлення складання, але воно не забезпечує функціональність планування складання. Даний розділ представляє використання даних САПР для створення послідовності складання пристрою.

3.1 Модель інформації моделі САПР

3.1.1 Інформаційна модель складальної одиниці

Основою моделювання приладу є відповідне вираження інформації про його деталі. Інформаційна модель деталі - це розташування рівня деталі, рівня елемента та рівня геометрії, як показано на рис.3.1. Рівень деталі включає повну інформацію про деталь, таку як назва, номер, розмір партії, назва матеріалу. Рівень характеристик включає форму, точність, допуск, шорсткість і термічну обробку матеріалу. Рівень геометрії представляє інформацію про геометрію та топологію точки, лінії та поверхні, які є основними одиницями системи твердотілого моделювання [59].

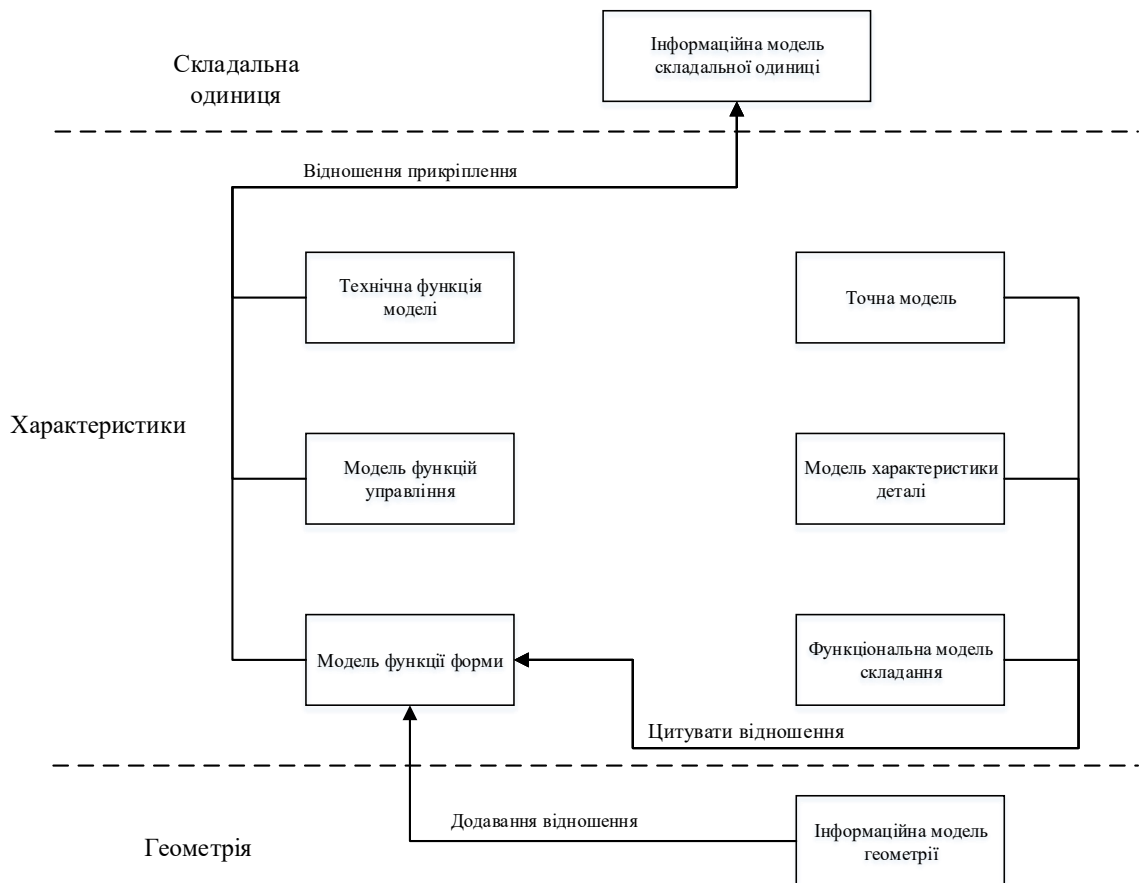


Рис.3.1. Інформаційна модель складальної одиниці.

3.1.2 Інформаційна модель складеного приладу

Вираз семантичної абстракції інформації про прилад інтегрується за допомогою трьох рівнів: концепція/функціональний, структурний та частина/функція.

Як показано на рис.3.2, існує кілька взаємозв'язаних етапів у проектуванні продукту на основі моделювання семантики складання. На ранній стадії схематичного проектування та функціональної декомпозиції анотація, заснована на функції, спочатку створюється проектувальником у своїй свідомості, щоб спробувати виконати всі вимоги, які вимагає цільовий продукт. Результат цього етапу дасть концепцію, а потім призведе до першого проекту прототипу продукту. Зокрема, розкладаючи початкову функцію на дуже концептуальні підфункції крок за кроком, логічно з'явиться структура прототипу продукту. На наступному кроці, виходячи з початкової структури продукту, необхідні характеристики деталей, такі як форма, точність та особливості складання, можуть бути організовані в

ієрархічну структуру. Водночас недоліки можуть бути негайно повернуті проектувальнику для перепроєктування на основі важливих факторів продукту, доки не будуть виконані всі функціональні вимоги продукту [60].

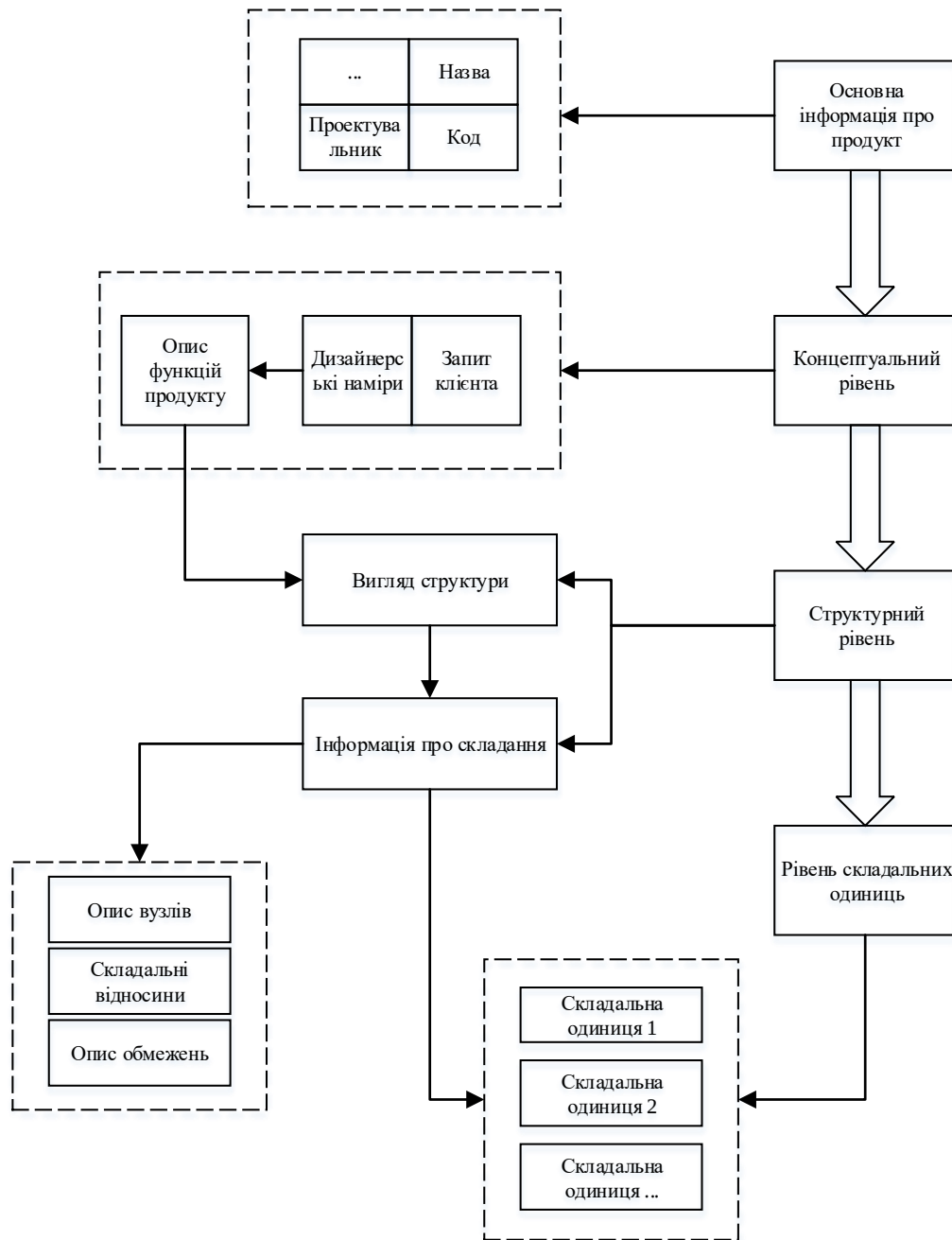


Рис.3.2. Структура складальної моделі.

3.1.3 Вилучення та дослідження даних складання моделі САПР

У наш час системи САПР інтенсивно розвиваються. Модель складання, створена системами САПР, містить багато корисної інформації, яку можна витягти за допомогою програмного забезпечення і використати для генерування необхідної інформації про складання 3Д-моделі.

Якщо модель містить гвинт, гвинтові компоненти можна автоматично упорядкувати відповідно до компонента, який закріплений гвинтом. Такі кріпильні елементи, як болт, гвинт, штифт тощо, завжди слідує за відповідними деталями кріплення. Нещодавно повідомлені підходи, засновані на САПР, використовують «ім'я компонента» для визначення типу компонента, який, швидше за все, виявить збій, якщо галузь використовує іншу мовну систему для визначення назв компонентів. Сьогодні САПР широко використовується для створення визначення продукту. САПР обіцяє геометрично правильне представлення складання, але вони не забезпечують функціональність планування складання.

Крім того, модель складання САПР містить конкретні відповідні дані, які є корисними для створення плану складання. Незважаючи на корисність цих даних, зв'язки між компонентами явно не надаються системою САПР. Для досягнення цієї мети був приведений алгоритм. На рис.3.3. представлений огляд розробленого підходу.

Після завантаження моделі складання алгоритм починається з вилучення компонентів і даних обмежень, а потім встановлюються зв'язки.

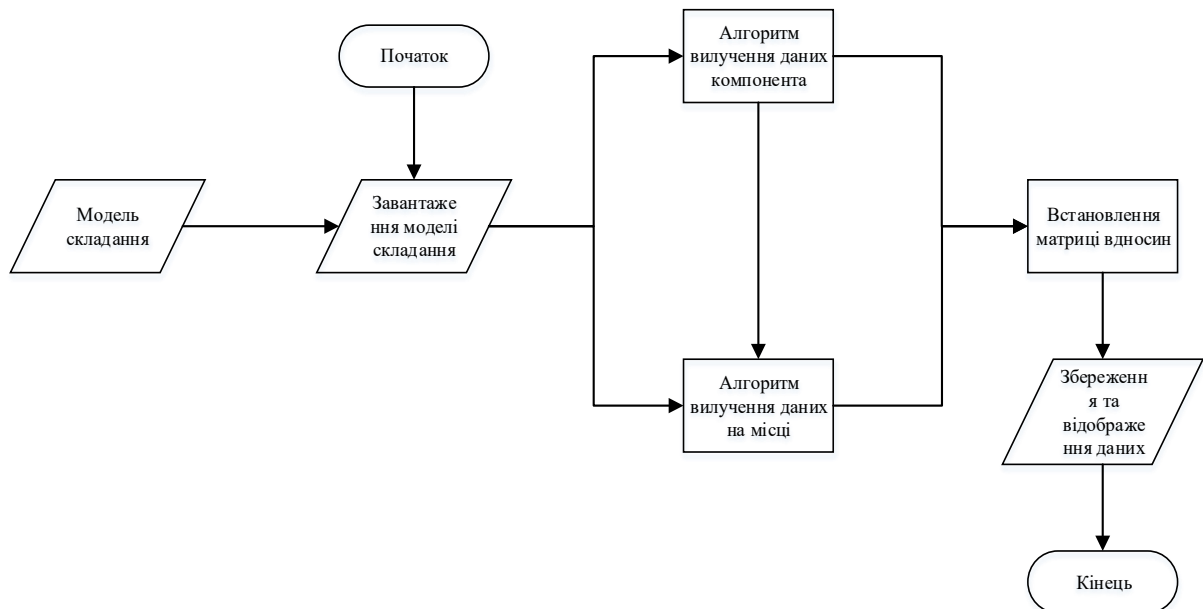


Рис.3.3. Алгоритм вилучення даних складання САПР.

Вилучення даних компонента

Алгоритм використовує властивості тіла кожного компонента, щоб отримати інформацію про масу, об'єм і площу. Крім того, він збирає ідентифікатор компонента (ID) і початкову позицію. Отримані дані згодом послужать для інших процесів, таких як ідентифікація взаємозв'язків та дослідження колізій.

Вилучення даних обмеження

Механічна модель складання – це набір компонентів, пов'язаних між собою за допомогою споріднених/складальних обмежень. Ці обмеження, що підтримуються системою САПР, є об'єктами, що містять кілька даних і параметрів наприклад: тип обмежень і пов'язані елементи.

Насправді компоненти можна додати в модель складання САПР двома методами. Перший – це висхідний метод, коли компоненти або частини моделі створюються окремо, а потім вони послідовно складаються у файлі складання. Метод зіставлення на місці — це низхідний метод, коли деталі створюються безпосередньо у файлі складання. Частина, створена за допомогою цього методу, автоматично фіксується обмеженням «На місці» (збіг).

Вилучення даних на місці

Алгоритм, показаний на рис.3.4, досліджує обмеження складання, присутні в дереві проектування менеджера функцій. Потім він вивчає їх природу і збирає з кожного з них: назву, тип і параметри.

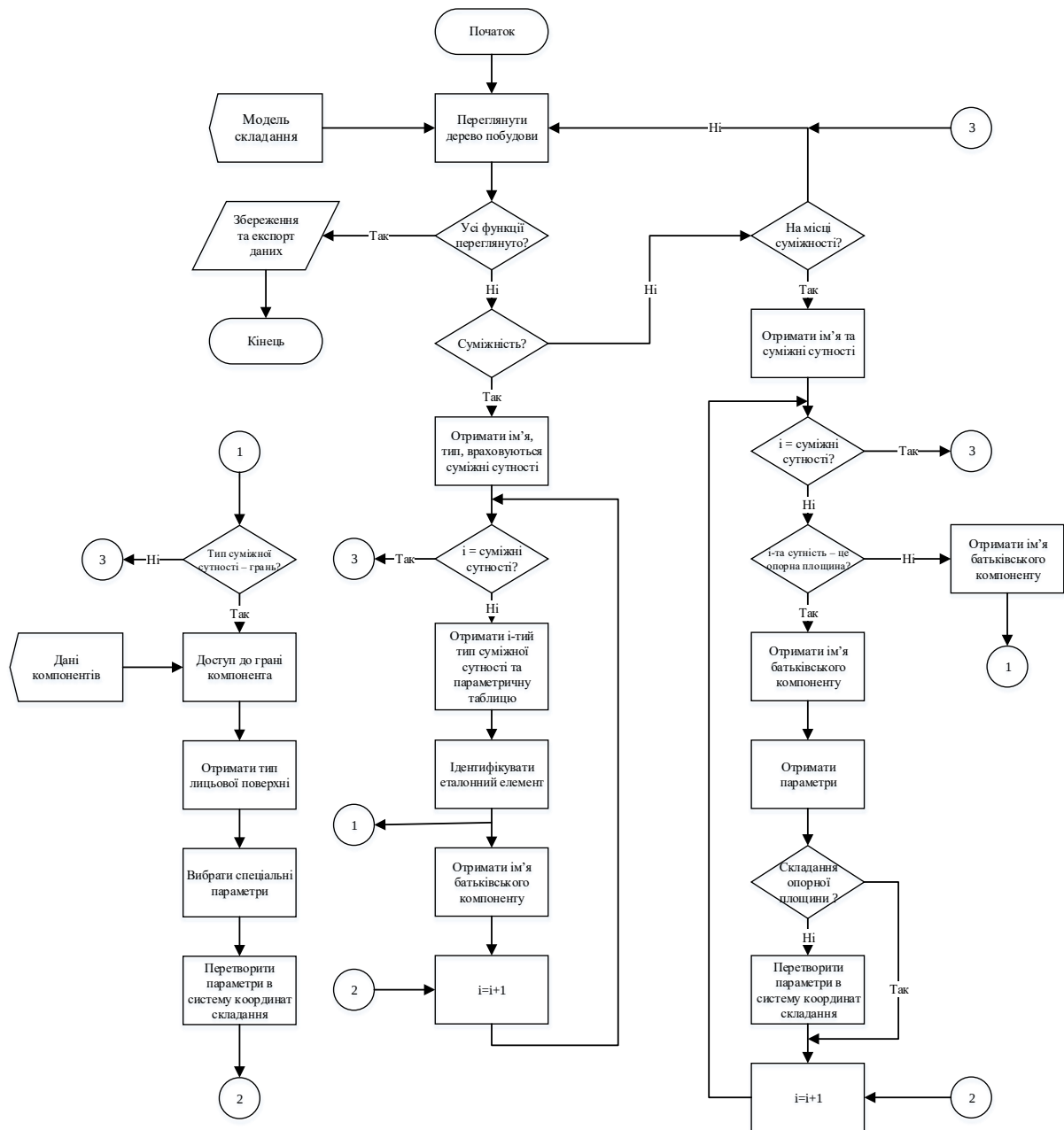


Рис.3.4. Алгоритм видалення даних складання САПР.

Основною проблемою в плануванні послідовності складання є те, як представити знання складання та зберегти їх у комп'ютері. Відмінні методи можуть спростити комп'ютерну обробку. Було узагальнено особливості складання та побудовано представлення знань.

Представлення знань є таким:

- параметри деталі ::= основні параметри , параметри геометрії , спеціальні параметри

- Основні параметри ::= ім'я частини , частина типу , номер креслення , вага , матеріал , колір
- геометричні параметри ::= загальний характер , ключ поверхню уявлення , торцева поверхня набір
- спеціальні параметри ::= спеціальні параметри з'єднувальної частини | спеціальні параметри загальної частини
- сполучна деталь спеціальних параметри ::= тип з'єднання , спільне напрямком , максимальна довжина суглоба
- назва частини ::= рядок символів
- тип деталі ::= тип деталі з'єднання | тип загальної деталі
- номер малюнка ::= рядок символів
- вага ::= додатне дійсне число
- матеріал ::= (сталь) | (алюміній) | (залізо) | (інші)
- колір ::= (синій) | (зелений) | (червоний) | (інші)
- загальна характеристика ::= зовнішній , внутрішній
- подання ключової поверхні ::= набір характеристик
- Тип з'єднувальної частини ::= болт/гвинт/заклепка/штифт/гайка/підшипник/шестерня/інше/
- метод з'єднання ::= 0 (з'єднання з напрямком) | 1 (з'єднання без напрямку)
- спільний напрям ::= вектор
- максимальна довжина з'єднання ::= додатне дійсне число.

Типи партнерів. Розпізнавання типу обмеження складання дуже важливо. Фактично це дає уявлення про характер контакту між компонентами. Ця інформація корисна для створення послідовності складання для визначення можливих напрямків складання. Крім того, кожна сучасна САПР-система включає в себе пакет обмежень для складання та з'єднань між двома компонентами.

Сутності. У САПР-моделюванні компонент характеризується набором сутностей (геометричні об'єкти: поверхня, технологічні об'єкти: матеріал тощо). Два компоненти складаються разом за допомогою геометричних об'єктів. Як наслідок, збір даних від об'єктів дуже корисний для:

- отримання геометричного характеру контакту;
- отримання конкретних геометричних параметрів сутності та позиції партнера;
- знати батьківський компонент.

Вилучені параметри. Параметри обмежень складання є одними з найважливіших зібраних даних. У разі обмеження складання параметри поєднаних об'єктів явно задаються параметричною таблицею, якої недостатньо для визначення об'єкта грані, особливо у випадку циліндричної, конічної, сферичної або торичної поверхні. Отже, розроблений алгоритм на першому кроці має доступ до батьківського компонента сутності та витягує конкретні параметри. Потім, на другому кроці, розроблений алгоритм перетворює витягнуті параметри з координати компонента в систему координат складання.

Дослідження стосунків

Важливим кроком для генерації послідовності складання є виявлення зв'язків у моделі складання. З витягнутих даних зв'язки між компонентами представлені матрицею Re . Ця матриця квадратна і симетрична, а її розмір дорівнює $(N \times N)$, де N представляє загальну кількість компонентів. Елемент re_{ij} з Re , який представляє кількість відносин між двома компонентами i та j , може мати такий атрибут:

- re_{ij} = кількість зв'язків між i та j ;
- $re_{ij} = 0$; якщо немає зв'язку між i та j ;
- $re_{ij} = -$; якщо $i = j$.

Аналіз зіткнень

Аналіз зіткнень моделі складання САПР є процесом, який дозволяє визначити шляхи, за якими компонент можна зібрати або розібрати без перешкод. Основною метою цього аналізу є переміщення компонента в

чітко визначених напрямках. Це переміщення, яке може бути переміщенням або обертанням, виконується матрицями перетворення і здійснюється через прирости переміщення. На малюнку 5 показаний глобальний алгоритм дослідження колізій. Розроблений аналіз зіткнень базується на трьох основних концепціях: напрямки руху, параметр приросту та генерація матриць перетворення та наступних положень, які представляють результат цього аналізу.

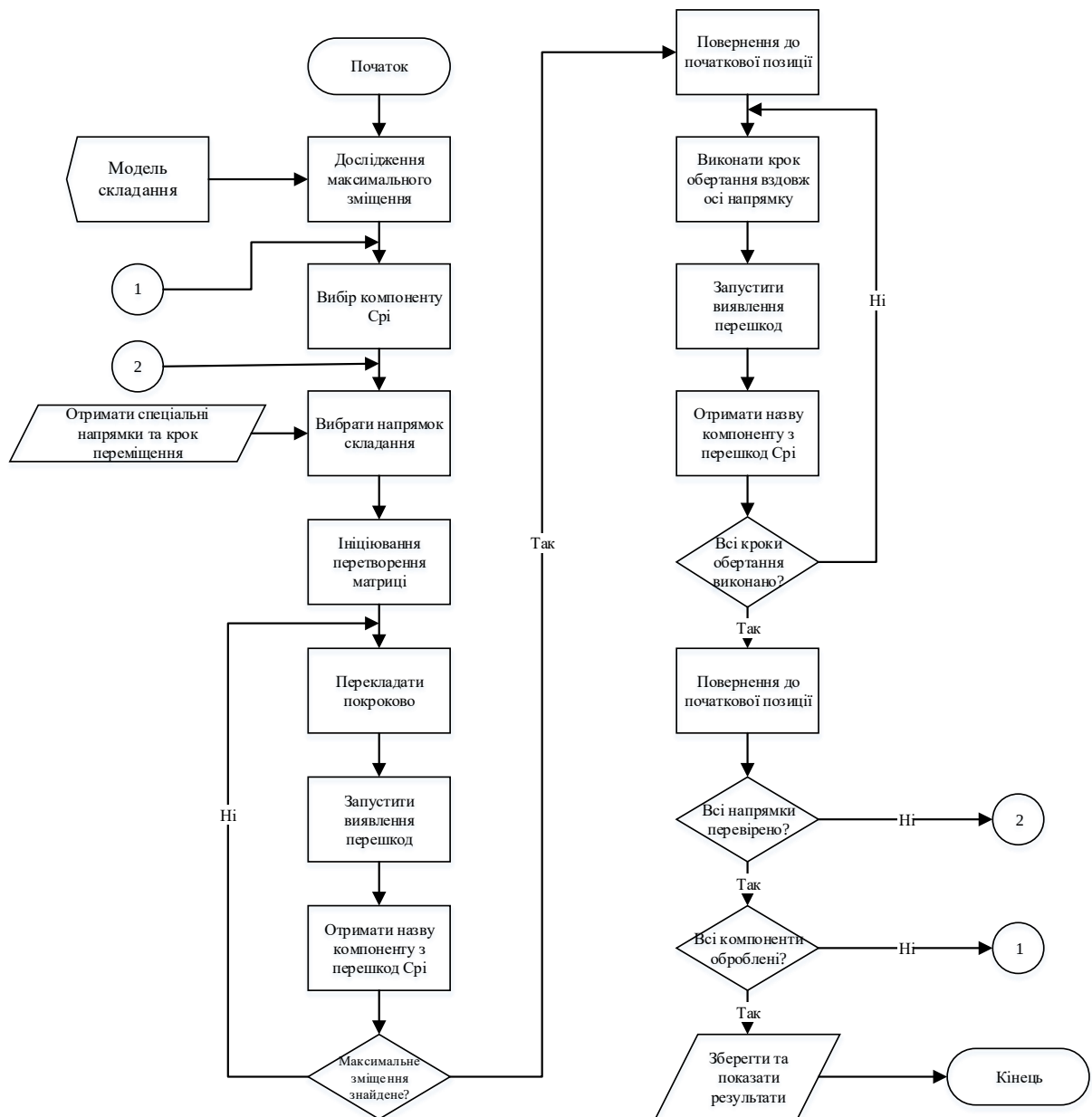


Рис.3.5. Глобальний алгоритм дослідження колізій.

Напрями руху

Визначення напрямків руху для виконання тестів на зіткнення є дуже важливим кроком. У літературі підходи [61], що використовують дослідження інтерференції, в якому компоненти переміщуються вздовж осей $\pm X$, $\pm Y$ і $\pm Z$ системи координат моделі складання. Це може призвести до неточних результатів, особливо для компонентів, які мають певну форму. Щоб уникнути будь-якого конкретного випадку, напрямки руху виводяться з осей обмежень моделі. Щоб мінімізувати непотрібне переміщення, алгоритм визначає ці напрямки з даних зіставлення, які вже вилучені, у такий спосіб:

1. Якщо споріднена сутність є точкою, лінією або площиною, компонент можна скласти відповідно до кількох напрямків. Алгоритм приймає директор або вектор нормалі як перший можливий напрямок складання. Потім з цього вектора будується ортогональний базис.
2. Якщо споріднена сутність є поверхнею обертання, лише вісь обертання представляє шлях складання.

Матриці перетворення та наступної позиції

У системах САПР компонент можна переміщати автоматично за допомогою інтерфейсів прикладного програмування. В основному цей метод здійснюється за допомогою матриці 3D-перетворення. Ці матриці забезпечують трансляцію та/або обертання компонента за будь-якою траєкторією. Як правило, тривимірна матриця перетворення $[Tr]$, задана рівнянням (3.1), є квадратною матрицею (4*4).

$$[Tr] = \begin{bmatrix} tr_0 & tr_1 & tr_2 & tr_1 \\ tr_3 & tr_4 & tr_5 & tr_{10} \\ tr_6 & tr_7 & tr_8 & tr_{11} \\ tr_{13} & tr_{14} & tr_{15} & tr_{12} \end{bmatrix} \quad (3.1)$$

У розробленому підході кожен компонент переміщується переміщенням уздовж його напрямків складання. Для переміщення кожного компонента з його початкової позиції складання алгоритм починається з

визначення початкової матриці перетворення $[M_0]$ компонента, який мав ту саму форму, що й $[Tr]$.

$$[M_0] = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{09} \\ m_{03} & m_{04} & m_{05} & m_{010} \\ m_{06} & m_{07} & m_{08} & m_{011} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Потім наступна матриця позиції $[M_1]$ отримується шляхом множення матриці перетворення $[Tr]$ на $[M_0]$.

$$[M_1] = [Tr] * [M_0] \quad (3.3)$$

У разі компонентів складальної моделі, які вставляють шляхом переміщення їх уздовж осі складання, матриця перетворення стає

$$[Tr] = [T] = \begin{bmatrix} 1 & 0 & 0 & i.d \\ 0 & 1 & 0 & j.d \\ 0 & 0 & 1 & k.d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

де

$[i, j, k]$: вектори напрямку можливого напрямку складання;

d : приріст переміщення.

І наступна матриця положень компонентів виражається як $[M_1]$

$$[M_1] = [T] * [M_0] = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{09} + i.d \\ m_{03} & m_{04} & m_{05} & m_{010} + j.d \\ m_{06} & m_{07} & m_{08} & m_{011} + k.d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

На основі попередніх кроків можна розпочати процес складання. Розроблений процес починається з дослідження базового компонента, що є важливим кроком для визначення порядку складання.

Дослідження базових компонентів

Базовий компонент є основним компонентом, на якому складаються більшість компонентів. Визначення базових компонентів є важливим кроком для створення порядку складання. У деяких роботах [62] базовий компонент вибирається відповідно до найбільшого об'єму матеріалу та/або за максимальною кількістю зв'язків з іншими компонентами. Ці критерії можуть не підходити для всіх моделей. Фактично, вільна площа компонента

може діяти як корисний критерій для ідентифікації базового компонента. У цій роботі та для уникнення конкретних випадків базовий компонент ідентифікується відповідно до оціночної функції S_i , розрахованої за допомогою

$$S_i = \alpha \frac{V_i}{V_a} + \beta \frac{A_i}{A_a} + \gamma * Nb \quad (3.6)$$

де

- V_i і A_i – відповідно об'єм і площа компонента C_{pi} ;
- V_a і A_a – відповідно середній об'єм і площа компонентів, наявних у складанні;
- Nb – кількість зв'язків компонента C_{pi} ;
- α , β , γ – представляють вагові коефіцієнти, їх сума дорівнює 1, $\alpha=\beta=0,35$ і $\gamma=0,3$.

Зауважте, що отримана оцінка кожного компонента C_{pi} пізніше буде служити критерієм упорядкування компонентів.

Порядок складання

Після того, як ідентифікація базового компонента виконана, алгоритм перевіряє його зв'язки з іншими компонентами, починаючи з компонента, що має вищий бал. Потім алгоритм перевіряє результати зіткнень, щоб здійснити можливий спосіб складання. Якщо випробуваний напрямок складання не містить складеного компонента, то він розглядається як напрям складання для цього компонента. Цю процедуру повторюють для кожного компонента, зібраного з базовим. У цьому випадку цей етап являє собою досягнення першого шару. Алгоритм виконує інший рівень, поки не будуть завершені всі компоненти. Результатом виконання алгоритму є отримання матриць всіх взаємозв'язків деталей у приладі.

3.2 Математичне забезпечення генерації послідовності складання

Основна мета фахівця зі складання — знайти оптимальну можливу послідовність складання, щоб виріб можна було зібрати за менший час [63]. Вироби з великою кількістю деталей призводять до кількох можливих

послідовностей складання; отже, дуже мало можливих послідовностей вважаються вибраними як найкращі на основі змін інструментів, зміни напрямку та інших доступних засобів, і цей процес займає багато часу. Крім того, експерт може не придумати найкращу послідовність складання з усього набору доступних можливих послідовностей [64]. Мета послідовності складання залежить від типу продукту, наявності техніки та ринкового попиту [65]. Промислова потреба спрямована на послідовність складання, яка може запропонувати численні переваги, як-от скорочення часу складання, вартість складання та багато іншого. Одна можлива послідовність складання може запропонувати менше часу на складання, але більшу вартість складання, інша може бути навпаки. Потреба спонукала дослідників придумати оптимальну послідовність складання. Методи оптимізації з багатоцільовими функціями придатності ефективні для пошуку оптимальних послідовностей складання. Однак література показує, що на початковому етапі, а потім і в подальшому, для досягнення оптимальної послідовності складання, де використовуються гібридні методи, застосовуються однооб'єктивні методики. Основна схема алгоритму оптимізації представлена на рис. 3.6.

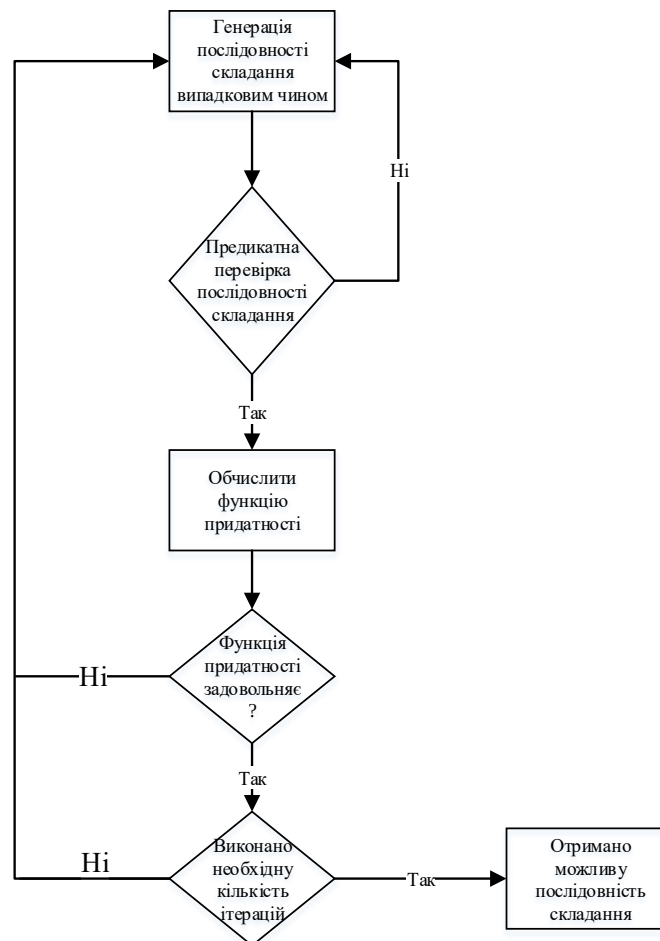


Рис.3.6. Принципова схема алгоритму оптимізації.

3.2.1 Генетичний алгоритм для генерації послідовності складання

Генетичний алгоритм(ГА) — це особливий клас еволюційного алгоритму, який використовується для обчислення оптимального або близького до оптимального рішення для задач пошуку. ГА заснован на еволюційних ідеях природного відбору та генетики [66]. Блок-схема ГА представлена на рис.3.7.

У традиційному ГА випадкова сукупність генерується з набору можливих рішень. Для згенерованого рядка функція придатності буде оцінена шляхом застосування операторів перехрещення та мутації. Під час оцінки фізичної підготовки особа зі слабкою фізичною підготовкою буде виключена. Цей процес буде продовжуватися до тих пір, поки він не досягне максимальної кількості ітерацій для отримання оптимального можливого рішення.

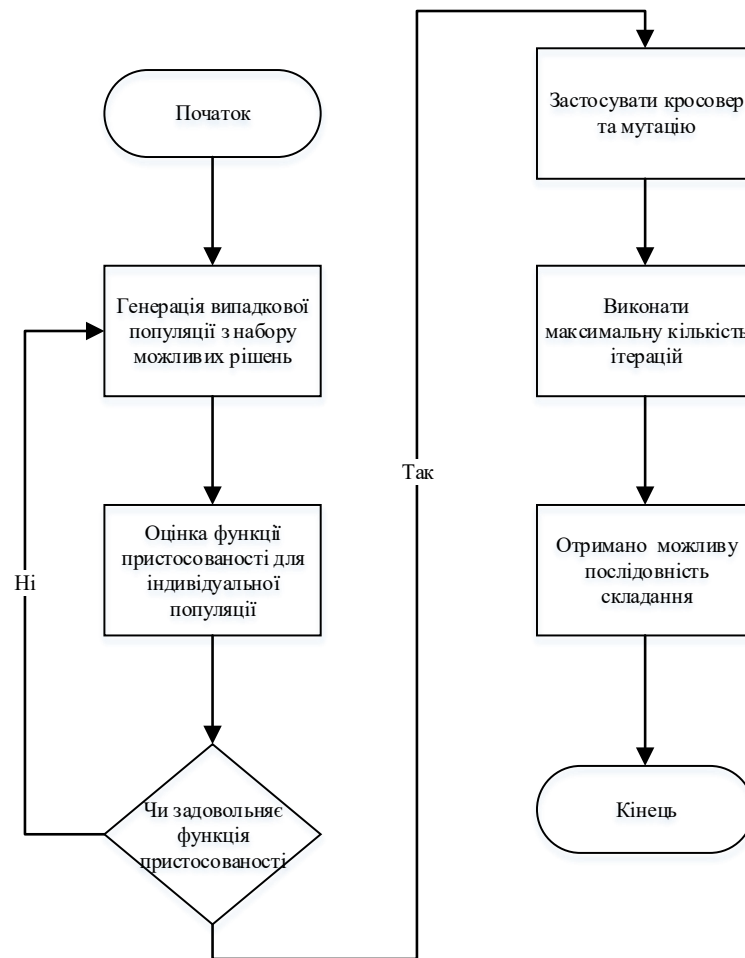


Рис.3.7. Основна принципова схема генетичного алгоритму.

Підхід оптимізації на основі ГА був використаний для визначення можливих і оптимальних послідовностей складання. Найпершим кроком у ГА є генерація початкової сукупності, яка генерується випадковим чином, але оскільки можуть бути як здійсненні, так і нездійсненні рішення, неможливі рішення уникають за допомогою матриці пріоритету (кількість рядків і стовпців дорівнює кількості частин), яка складається зі значень 0 і 1, щоб вказати, чи має частина пріоритет перед іншими частинами. Допустима початкова сукупність оцінюється за допомогою функції пристосованості, яка складається з кількості змін орієнтації напрямку та змін інструменту/захвата та доцільності послідовності. На рис.3.8 показана блок-схема реалізованої ГА.

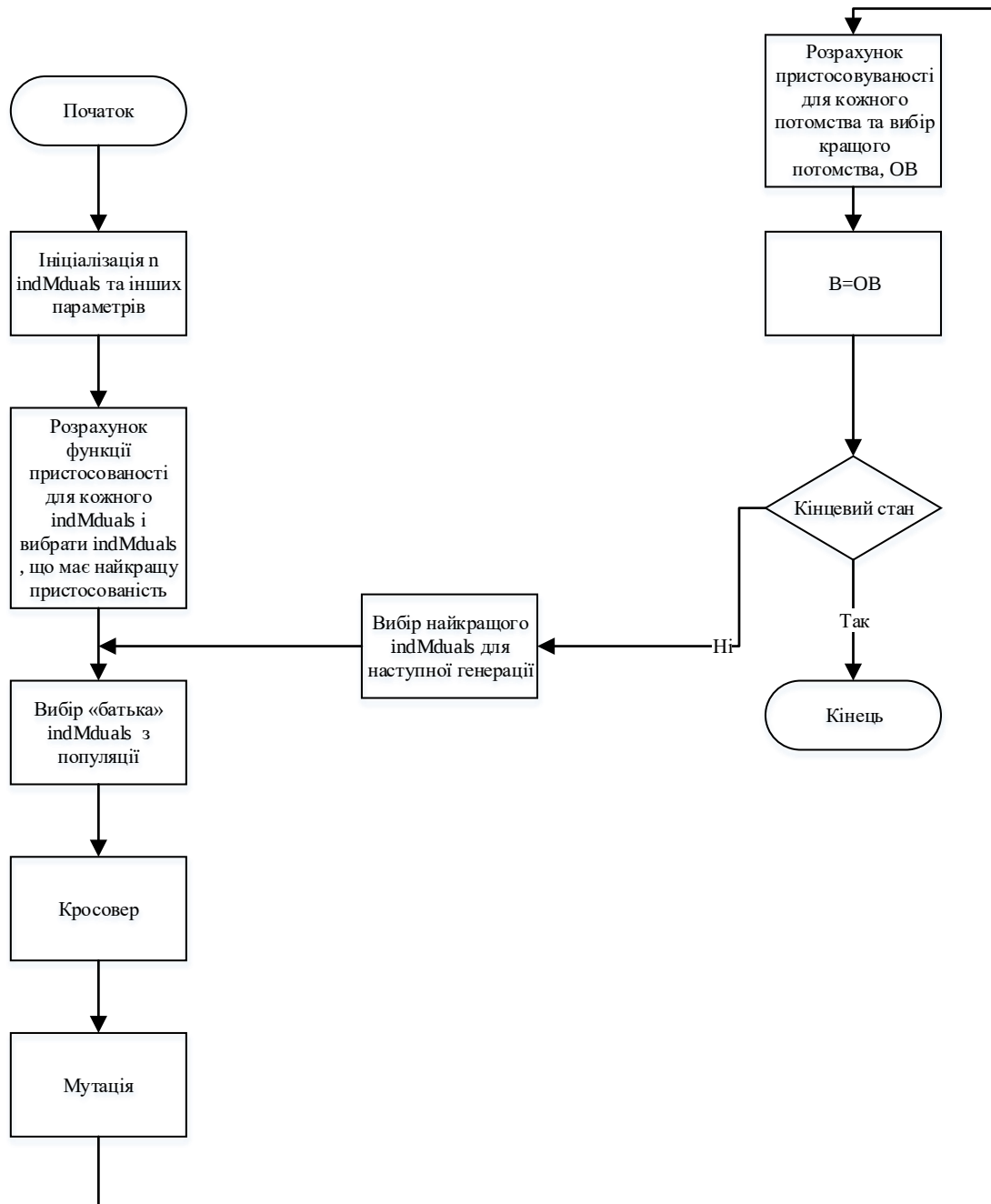


Рис. 3.8. Блок-схема реалізованого генетичного алгоритму.

Першим кроком у розробці генетичного алгоритму для планування складання є відображення рішень проблеми (послідовності складання) на хромосоми [67]. Хромосома повинна бути закодована таким чином, щоб вона містила інформацію про рішення, яке вона представляє. Початкова сукупність генерується випадковим чином з групи матриці пріоритету, а розмір сукупності встановлюється на «n». Тому рядки «n» будуть створюватися випадковим чином. Фенотипове кодування використовується

для представлення хромосоми, як показано на рис.3.9. Кожен вузол відповідає гену, який представляє номер завдань на певному рівні, необхідному для побудови моделі разом з іншими завданнями на цьому конкретному рівні плану.

Хромосоми, які представляють плани складання, перевіряються на доцільність з матриці пріоритету перед проведенням генетичної операції.

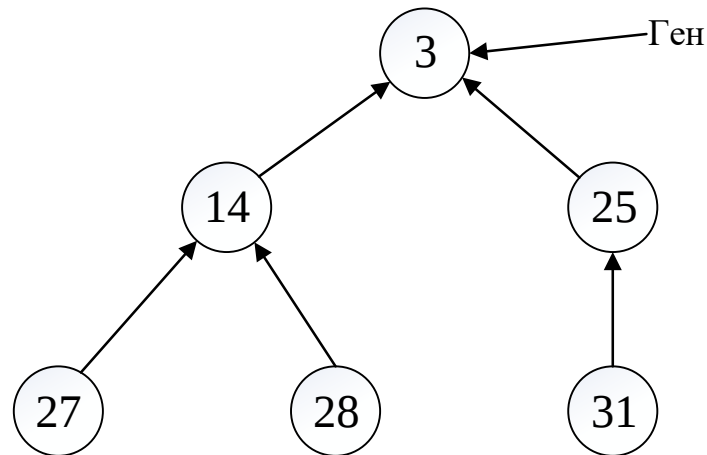


Рис. 3.9. Типова кодована хромосома.

Цільова функція використовується для вимірювання того, як окремі частини показують себе в проблемній області. Цільовою функцією задачі є отримання оптимальних або близьких до оптимальних планів складання на основі мінімізації розмаху для даного виробу. Цільова функція визначається так:

$$m = P - 1 \quad (3.7)$$

$$P_t(n) = OC(n) + DoF(n)$$

$$C_t(AP) = \sum_i^m P_t(i) \quad (3.8)$$

$$MS = Min[C_t]_j \quad (3.9)$$

де m - кількість завдань у кожному дереві складання; P - кількість деталей; $P_t(n)$ - час обробки завдання « n », оскільки i означає кількість завдань, необхідних для завершення кінцевого продукту; n - кількість завдань представляє пару деталей, отриману з відношень пріоритету графа зв'язків;

$OC(n)$ - операційна складність; $DoF(n)$ - ступені свободи підгрупи; $[C_t]_j$ - час завершення кожного плану складання, оскільки «j» означає чисельність популяції; MS- різниця у часі між початком та закінченням послідовності робіт для кожної кількості планів складання.

Представлення хромосом

Для розв'язування досліджуваних проблем за допомогою методики ГА потрібна рядкова схема для кодування варіантів рішень у вигляді символічних рядків, які називаються хромосомами. Цілі числа використовуються для представлення операції призначення завдання робочій станції з послідовним числом 1, 2, ..., N, відповідно. В таблиці 2.1. представлений приклад закодованих рядків, який включає три завдання і три робочі станції [68].

Таблиця 1.2. Приклад закодованих рядків.

Завдання Робочі станції			
	a	b	c
W1	1	2	3
W2	4	5	6
W3	7	8	9

Кодування: рішення-кандидат може бути закодовано як рядок за допомогою перестановки $\{1, 2, \dots, N\}$ для утворення хромосоми. Наприклад, «2-1-4-3-5-6-8-7-9» є однією з можливих хромосом у прикладі, показаному в таблиці 1.2.

Декодування: можливе рішення може бути розшифровано з даної хромосоми. Наприклад, рядок "1-4-5-2-9-6-8-7-3" можна декодувати як, Завдання а призначається робочій станції W1 (рядок 1), b – W2 (рядок 5), і с – W3 (рядок 9), відповідно до закодованого рядка в таблиці 2.1. Рядки 4 (завдання а), 2 (завдання b), 6 (завдання с), 8 (завдання b), 7 (завдання а), і 3 (Завдання с) є манекенами.

Самонастроювання: інколи послідовність генів у даній хромосомі може порушувати відносини пріоритету. Тут використовується механізм самонастроювання, щоб змінити нездійсненну хромосому.

Відбір

Для вибору рішень, які входять у пул сполучення, був застосований оператор генетичного алгоритму, відомий як - пропорційний відбір за придатністю. Його принцип полягає в тому, що спочатку за допомогою функції пристосованості оцінюються всі рішення, а потім відповідно до їх значень пристосованості створюються діапазони. Генеруються випадкові числа (кількість випадкових чисел дорівнює розміру популяції), а потім виявляється діапазон, в який вони потрапляють. Ці значення встановлюються у пул сполучень, на якому виконується операція кросовера.

Кросовер

Було використано частково узгоджений кросовер (PMX). Кросовер PMX створює потомство, вибираючи підпоследовність рішення від одного з батьків і зберігаючи порядок і положення якомога більшої кількості генів від іншого батька. Підпоследовність вибирається шляхом вибору двох випадкових точок, які служать межами для операцій заміни. Але в задачі автоматизованого планування складання жодна частина последовності складання не може бути повторена, про що дбає вищезгаданий оператор кросовера [68].

Мутація

Була використана мутація підкачки(заміни). Процедура заміни випадковим чином вибирає два гени, щоб визначити два гени зміни вибраного батька. Операція мутації обробляється незалежно від того, чи визначається вона значенням, згенерованим випадковим чином. Якщо значення перевищує швидкість мутації, встановлену у файлі конфігурації параметрів ГА, мутація не виконується, а вибраний батьківський елемент передається безпосередньо в нове покоління.

Функція пристосованості

Була створена функція пристосованості, яка в основному включає кількість змін орієнтації та зміни інструменту/захвата. Рішення з меншою придатністю мають високі шанси на виживання, оскільки метою цієї роботи

є мінімізувати час і вартість складання, на які впливає кількість змін напрямку та кількість змін інструментів/захватів. Отже, функція вибирається так, щоб перетворити її в задачу максимізації. Початкова сукупність (доступна сукупність) генерується випадковим чином. Щоб уникнути створення нездійсненої популяції, після схрещування та мутації функція придатності також оснащена індексом доцільності [68]. Під час оцінки рішень, якщо вони виявляються нездійсненими, з функції придатності віднімається певне значення (тобто штраф). Оскільки для наступного покоління підбираються лише кращі рішення, алгоритм поступово починає давати лише можливі рішення. Рівняння (3.10) показує функцію загальної придатності (FF), яка була використана в цій роботі. Символи w_x і w_y в рівнянні (2.4) – це ваги, які має встановити користувач, n_n – кількість змін напрямку, $n_{ш}$ – кількість штрафів.

$$FF = \frac{1}{w_x * n_n + w_y * n_{ш}} \quad (3.10)$$

Отже, ГА є методом оптимізації для вирішення проблем оптимізації послідовності складання завдяки його здатності запропонувати гнучкий спосіб визначення обмежень. ГА був обран для вирішення поставленої проблеми тому, що:

- ГА може просто та швидко працювати зі значними просторами пошуку;
- ГА гнучкий у визначенні обмежень і встановленні їх у функції пристосованості. Це корисно для послідовності складання, де функцію якості важко визначити;
- ГА придатний для переходу від поточного рішення в просторі пошуку до будь-якого подальшого рішення.

3.2.2 Граф зв'язку

Граф зв'язку — це концепція представлення зв'язків між парами частин для опису значущих взаємозв'язків між частинами приладу [69]. Зв'язок — це контакт/зв'язок, встановлений між компонентами [70]. Граф зв'язку

визначається як $LG=G(V,E)$ граф вершин і ребер (з'єднує гіпердуги). Кожна вершина вказує на складену частину, а кожне ребро означає зв'язок між двома компонентами. Граф зв'язку представлений на рис.3.10 через з'єднання (C_i) між деталями.

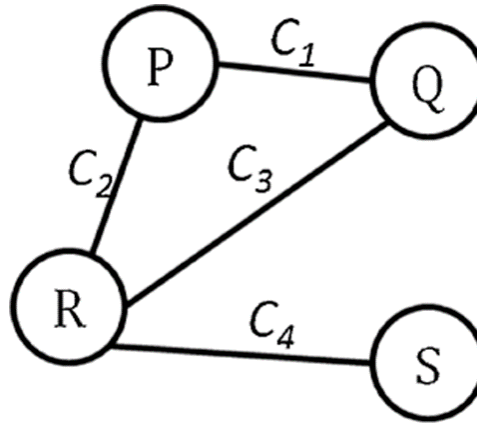


Рис.3.10. Граф зв'язку приладу з чотирьох частин.

Матричне представлення зв'язків з використанням двійкових кодів 1, 0 [61]. Матриця 'nхn' потрібна для представлення зв'язків продукту, зібраного за допомогою 'n' компонентів. Елементи діагоналі цієї матриці складаються з нульових значень, а рядок матриці представляє зв'язки між одним компонентом з іншими компонентами в моделі. Стовпець матриці представляє компоненти, пов'язані зв'язковими відношеннями. Підматриці матриці 'nхn' представляють локальні зв'язки в підсистемах. Матриця зв'язку для моделі складання з чотирьох частин виглядає наступним чином:

$$\begin{array}{c}
 \begin{array}{c} P \\ Q \\ R \\ S \end{array}
 \begin{array}{cc|cc}
 P & Q & R & S \\
 \hline
 \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
 \end{array}
 \end{array}
 \quad (3.11)$$

Комп'ютерні методи та алгоритми були розроблені для вилучення матриці зв'язку через інтерфейс прикладних програм з моделей 3Д САПР для виконання генерації послідовності складання [71,72]. На рис.3.11 представлена блок-схема для вилучення матриці зв'язку з 3D-моделі САПР шляхом виконання аналізу контактів моделі.

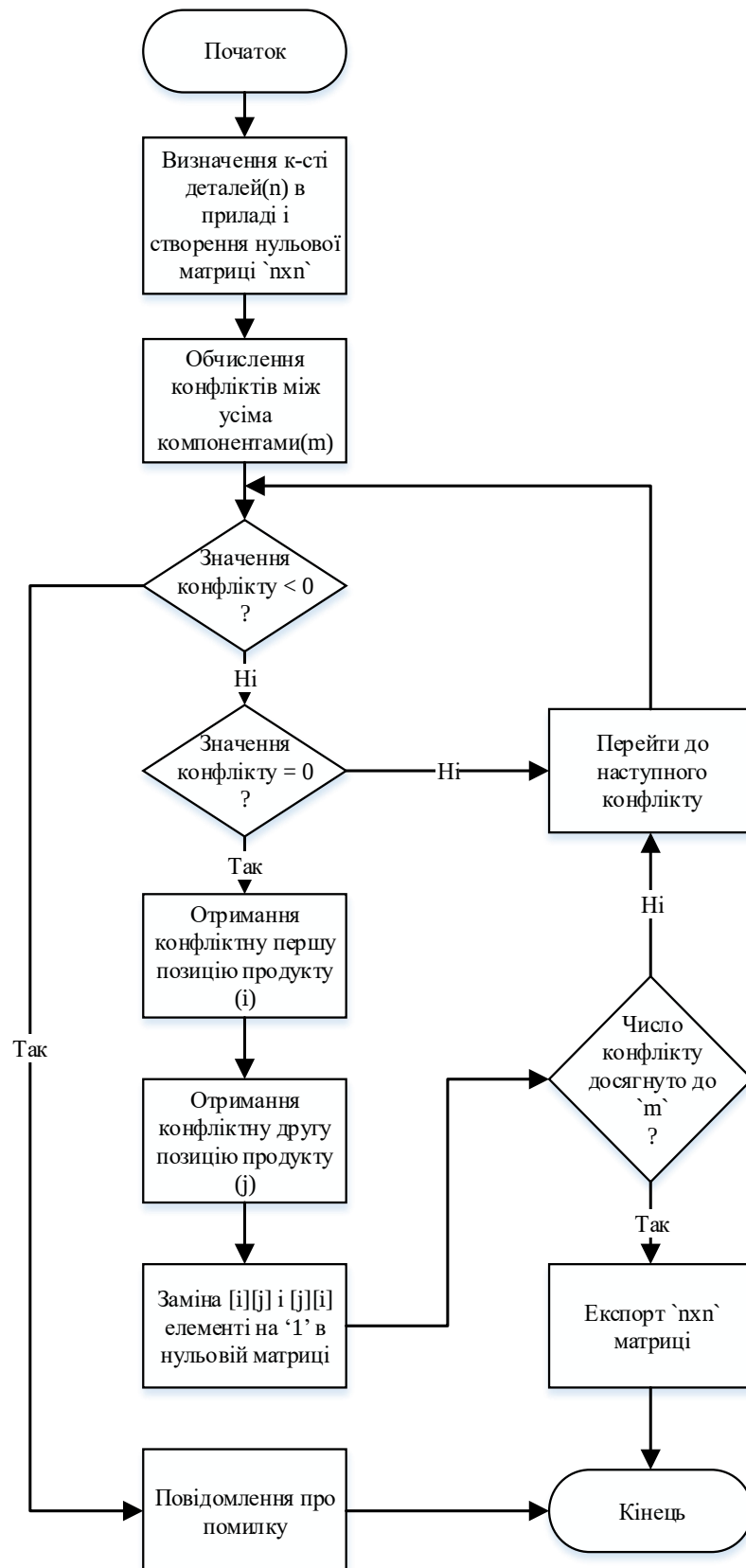


Рис.3.11. Блок-схема механізму вилучення матриці зв'язку.

Запропонований метод(рис.3.11) [73] складання розрізів генерує можливі декомпозиції зібраного продукту, які називаються наборами складання. Складальні розрізи можуть бути роз'єднаними або утвореними

шляхом розриву зв'язку. Подальше вивчення відносин пріоритету на наборах складання призводить до можливих послідовностей складання.

Предикат здійсненності

Предикат здійсненності справедливий для послідовності складання, коли він задовольняє геометричній здійсненності та механічній здійсненності. Вважається, що послідовність складання є геометрично здійсненою, коли всі частини можуть з'єднатися у визначеній послідовності без будь-яких зіткнень. Геометрична здійсненність є функцією геометрії деталей. Механічна доцільність справедлива для послідовності складання, коли інструменти складання можуть виконувати визначену операцію складання без будь-яких зіткнень; отже, це залежить від інструментів і методів, які використовуються для виконання операцій складання. Відношення пріоритету використовуються для перевірки можливості складання деталі. Відношення пріоритету дають інформацію про попередні та наступні операції складання для досягнення можливої послідовності складання.

ВИСНОВКИ ДО РОЗДІЛУ

Інтерпретація даних САПР є складним завданням. Незважаючи на те, що САПР є багатим джерелом даних про продукт, бракує методів інтерпретації даних САПР для планування послідовності складання.

В даному розділі подана інформація орієнтована на розробку системи для вилучення даних про продукт і встановлення, завдяки вилученим даним, послідовності складання цього продукту.

Запропоновані алгоритми покращують можливості планування складання на основі генетичного алгоритму шляхом прямого вилучення інформації з файлів STEP, аналізу геометричної інформації для визначення геометричних обмежень складання та автоматичного генерування матриць, які представляють взаємозв'язок між деталями 3Д-моделі.

РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ СКЛАДАННЯ

Для реалізації програмного забезпечення для поставленої задачі було використано мову програмування Python [74]. Мова Python неймовірно проста у використанні та вивченні для новачків. Мова Python є однією з найдоступніших доступних мов програмування, оскільки вона має спрощений синтаксис і не є складною, що робить більший акцент на природній мові. Завдяки простоті навчання та використання коду на Python можна легко писати та виконувати набагато швидше, ніж інші мови програмування.

Завдяки корпоративному спонсорству та великій спільноті, що підтримує, python має чудові бібліотеки, які ви можете використовувати, щоб вибрати та заощадити свій час і зусилля на початковому циклі розробки.

Python можна використовувати майже в будь-якому середовищі, і ви не зіткнетеся з проблемою втрати продуктивності, незалежно від платформи, на якій він працює.

Ще одна найкраща особливість універсальності мови Python полягає в тому, що її можна використовувати в багатьох різноманітних середовищах, таких як мобільні додатки, настільні програми, веб-розробка, апаратне програмування та багато іншого. Універсальність python робить його більш привабливим у використанні завдяки великій кількості додатків.

Мова Python може дуже допомогти в автоматизації завдань, оскільки є багато доступних інструментів і модулів, що робить роботу набагато зручнішою. Неймовірно знати, що можна легко досягти просунутого рівня автоматизації, просто використовуючи необхідні коди на Python.

Отже, використання мови програмування Python є найкращим рішенням для поставленої задачі та для створення інтерфейсу користувача.

Код програми можна побачити у додатку А.

4.1 Модель роботи інтерфейсу програми

Розроблена програма може працювати та виконувати поставлену перед нею задачу без використання візуального інтерфейсу, але в такому випадку майбутньому користувачу знадобиться багато часу для розуміння програми. Для швидкого зчитування та обробки вхідних даних, а також зрозумілого виводу результатів виконання та полегшення користування було розроблено інтерфейс.

Інтерфейс забезпечує швидкий зв'язок та передачу результатів між розробленими модулями. Алгоритм роботи інтерфейсу програми зображено на рис.4.1.

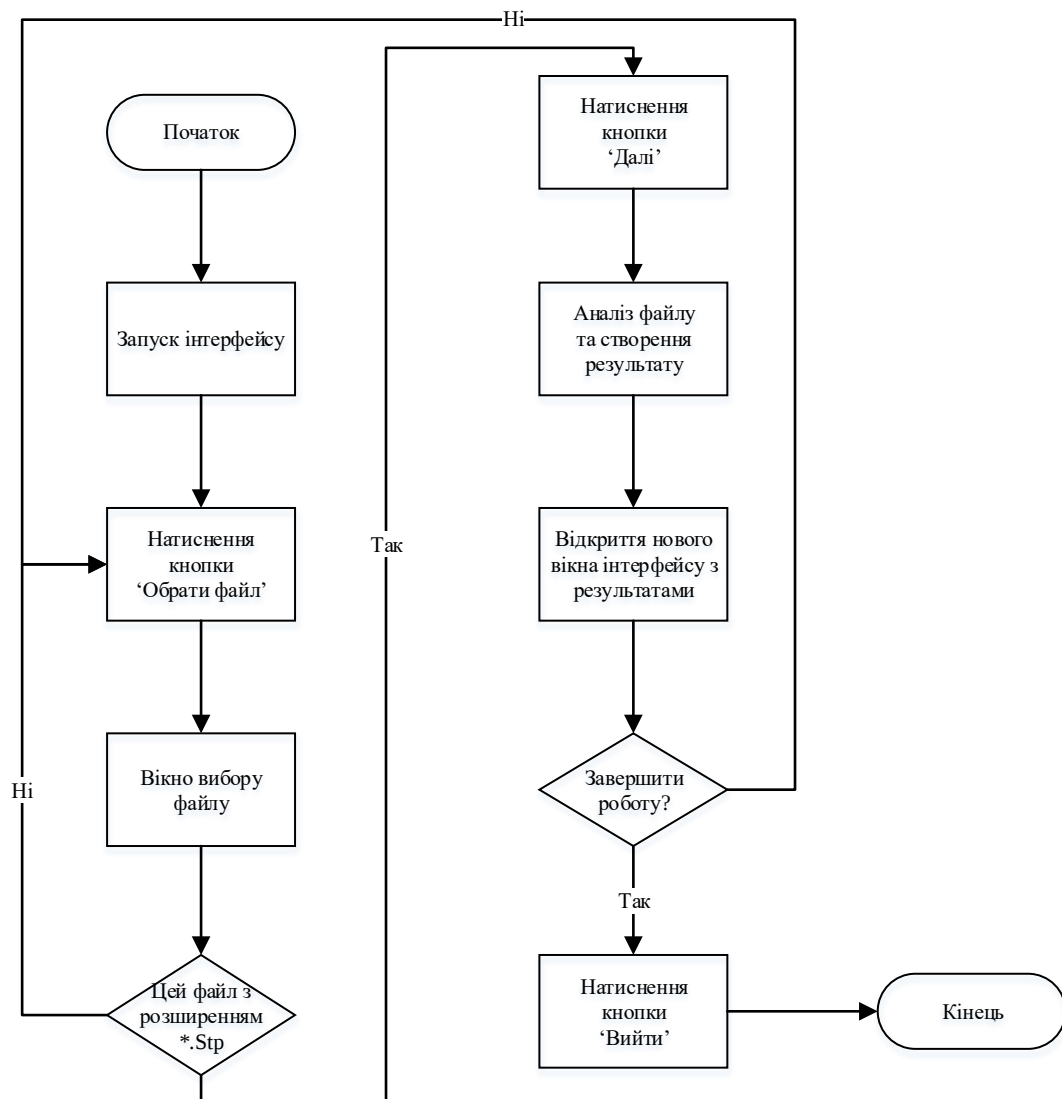


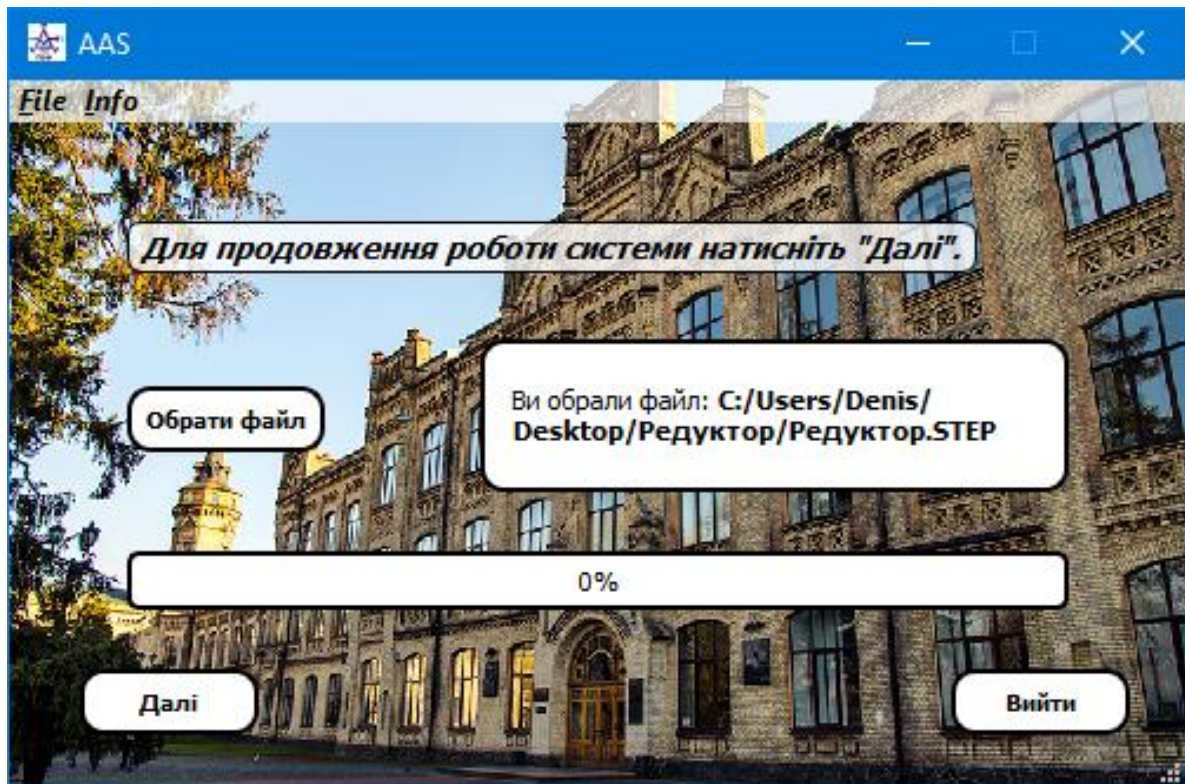
Рис.4.1. Алгоритмічна модель роботи інтерфейсу.

Як видно з алгоритму(рис.4.1) інтерфейс програми дуже простий, адже аналіз та всі розрахунки відбуваються саме в кодї, також програма складається з двох вікон інтерфейсу: перше вікно – введення даних, друге вікно – показ результатів. За інтерфейс першого вікна відповідає клас *main_menu*, другого вікна – *answer_menu*.

У програмі інтерфейсу реалізовано перевірку на правильність обраного файлу, тобто поки користувач не обере файл із правильним розширенням *.stp, саме в такому форматі 3Д-моделі зберігається інформація про взаємозв'язок між деталями приладу яку можливо проаналізувати, не буде доступною нажаття кнопки 'Далі'. Приклад цього можна бачити на рис.4.2. Більше перевірок для роботи програми непотрібно, вона виконає все сама.



а)



б)

Рис.4.2. Інтерфейс програми:

а) обрано невірний формат файлу;

б) обрано вірний формат файлу.

Отже, інтерфейс програми допоможе проектувальнику без розуміння коду швидко виконати пошук найкращого плану складання обраного приладу.

4.2. Програмна реалізація підсистеми автоматизованого проектування складання

Як вже було написано, Python має велику базу різноманітних бібліотек, які допомагають реалізувати будь-які поставлені задачі. Для реалізації інтерфейсу користувача було використано бібліотеку Python PyQt5 [75]. Дана бібліотека надає велику кількість методів та класів для побудови інтерфейсу.

PythonOCC — це проект, який спрямований на надання всього спектру функцій технології OpenCASCADE через ядро PythonOCC-Core [76]. Саме ця бібліотека надає методи для аналізу 3Д-моделі приладу. На основі

PythonOCC було реалізовано алгоритм вилучення даних складання приладу, які є основою для подальших досліджень.

Для реалізації генетичного алгоритму, який виконує пошук найкращого маршруту складання приладу, було використано бібліотеку Pyevolve — це фреймворк з відкритим вихідним кодом для генетичних алгоритмів [77].

NetworkX — це пакет Python для створення, маніпуляції та вивчення структури, динаміки та функцій складних мереж [78]. Завдяки цій бібліотеці було реалізовано алгоритм створення графу зв'язку, в якому зображено всі зв'язки між деталями приладу та позначено отриману генетичним алгоритмом послідовність складання. Також було використано бібліотеку matplotlib(створення статичних, анімованих та інтерактивних візуалізацій), яка необхідна, в даній програмі, лише для створення файлу зображення графу у форматі *.png.

Створена програма складається із 5 файлів:

- main.py – алгоритм запуску програми;
- GUI.py – алгоритм інтерфейсу;
- stepanalyzer.py – алгоритм аналізу 3Д-моделі;
- GA.py – алгоритм генетичного алгоритму;
- graph_.py – алгоритм створення графу зв'язку.

Структуру даних файлів зображено в рис.4.3.

```
from GUI import *
import sys
def start():...
if __name__ == '__main__':
    start()
```

a)

```
import csv
import networkx as nx
import matplotlib.pyplot as plt

def create_graph(mas, NUM_PARTS, csv_file):
```

б)

```

# -*- coding: utf-8 -*-
import time
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import traceback
from GA import *
from graph_ import create_graph
style = (''...'')
class Thread(QThread):...
class main_menu(QMainWindow):
    def __init__(self):...
    def initUI(self):...
    def center(self):...
    @pyqtSlot()
    def getFile(self):...
    @pyqtSlot()
    def help(self):...
    @pyqtSlot()
    def info(self):...
    @pyqtSlot()
    def on_click(self):...
    def signal_accept(self, msg):...
    def closeEvent(self, event):...

class answer_menu(QMainWindow):
    def __init__(self, name, parent=None):...
    def initUI(self):...
    @pyqtSlot()
    def save(self):...
    @pyqtSlot()
    def help(self):...
    @pyqtSlot()
    def info(self):...

```

B)

```

import csv
from random import shuffle
from pyevolve import G1DList, GSimpleGA, Consts, Initializers, Selectors, Mutators, Crossovers
precedence_list = []
orientations = []
tools_grippers = []
def check_precedence_criteria(sequence):...
def check_precedence_swaps(sequence):...
def fitness_func(chromosome):...
def calc_orientation_changes(sequence):...
def calc_tool_changes(sequence):...
def init_pop(genome, **args):...
def GA_algorithm(NUM_PARTS,ors,tos,matrix):...

```

г)

```

#!/usr/bin/env python
from OCC.Core.IFSelect import IFSelect_RetDone
from OCC.Core.STEPCAFControl import STEPCAFControl_Reader
from OCC.Core.TDF import TDF_Label, TDF_LabelSequence
from OCC.Core.TCollection import TCollection_ExtendedString
from OCC.Core.TDocStd import TDocStd_Document
from OCC.Core.XCAFApp import XCAFApp_Application_GetApplication
from OCC.Core.XCAFDoc import XCAFDoc_DocumentTool_ShapeTool
class StepAnalyzer():
    def __init__(self, document=None, filename=None):...
    def read_file(self, fname):...
    def dump(self):...
    def find_components(self, comps):...

```

д)

Рис.4.3. Структура файлів програми:

а) main.py; б) graph_.py; в) GUI.py; г) GA.py; д) stepanalyzer.py.

Отже, алгоритм роботи створеної програми зображено на рис.4.4.

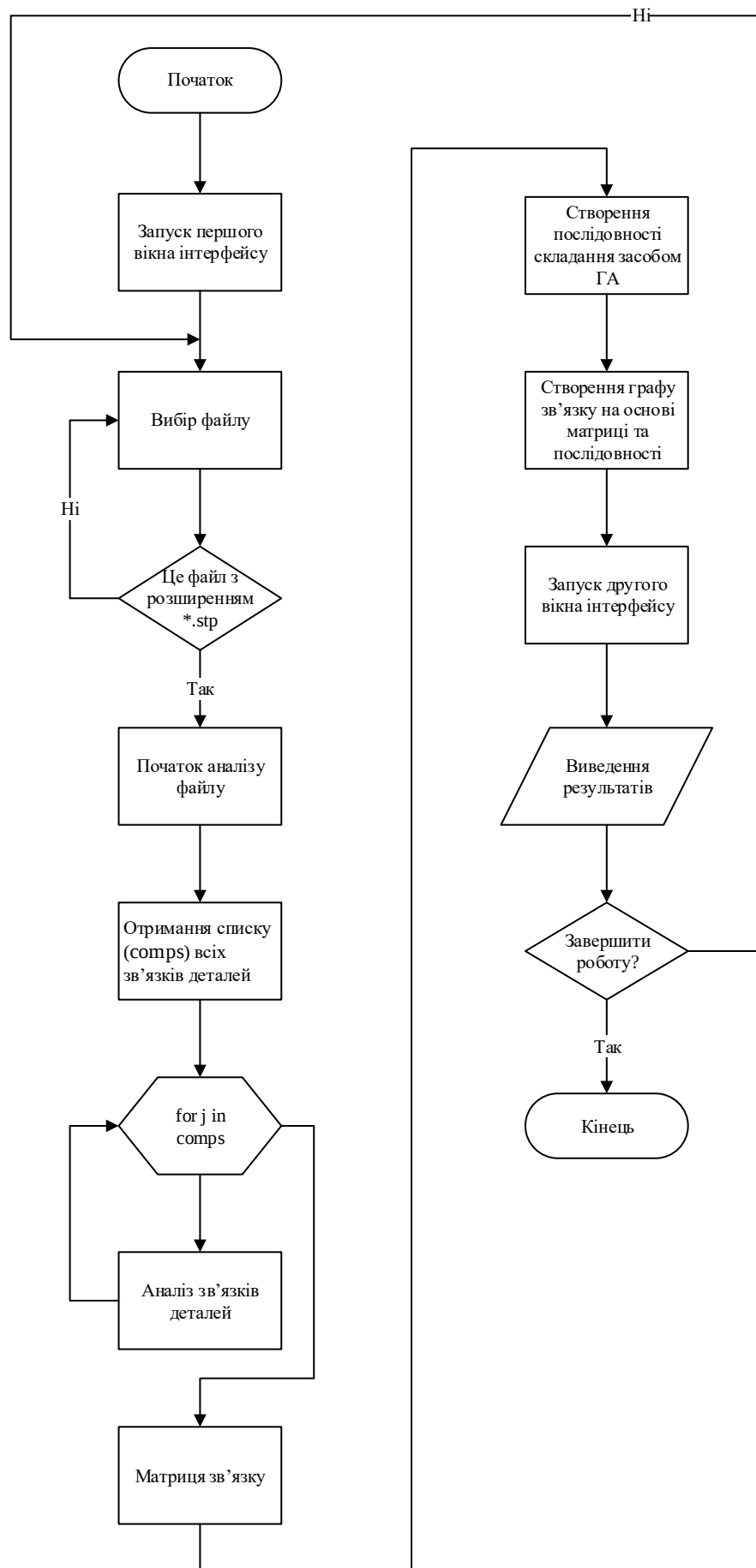


Рис.4.4. Блок-схема програмної реалізації системи.

4.3. Дослідна експлуатація розробленої програми

Для прикладу роботи розробленого програмного забезпечення було обрано 3Д-модель редуктора (рис.4.5). Хоча модель є не досить складною в плануванні її складання, але кількість деталей створює велику кількість можливих варіантів складання, що в свою чергу робить дану модель гарним прикладом для використання в розробленому програмному забезпеченні.

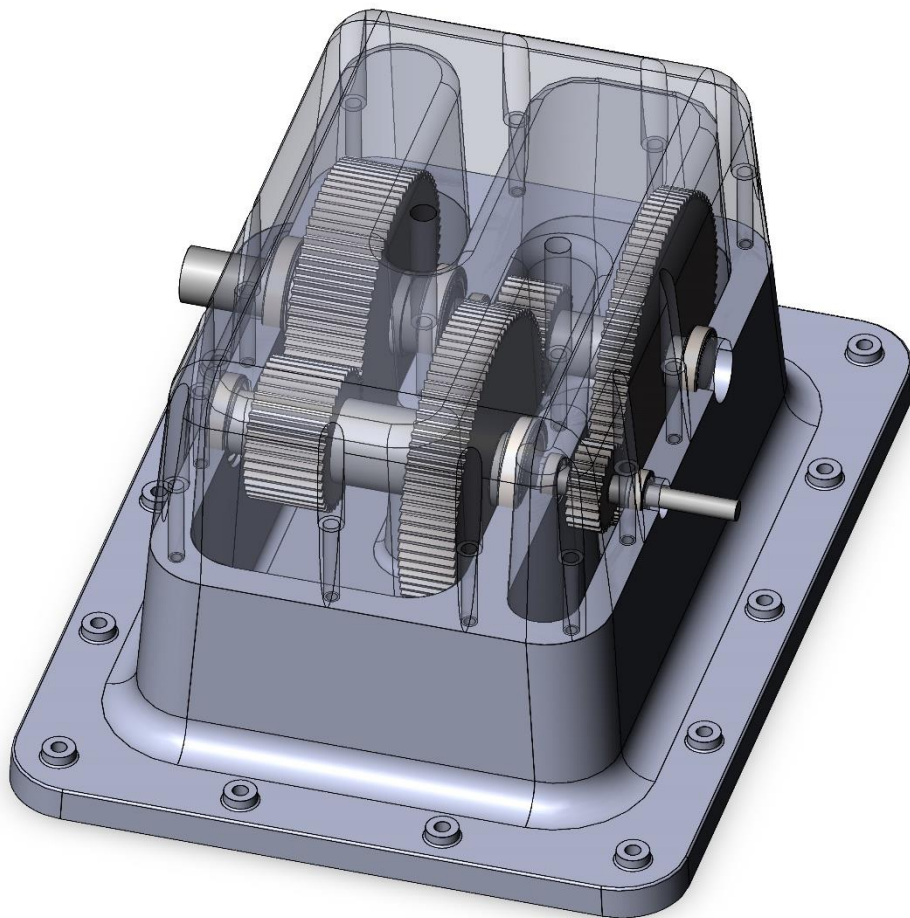
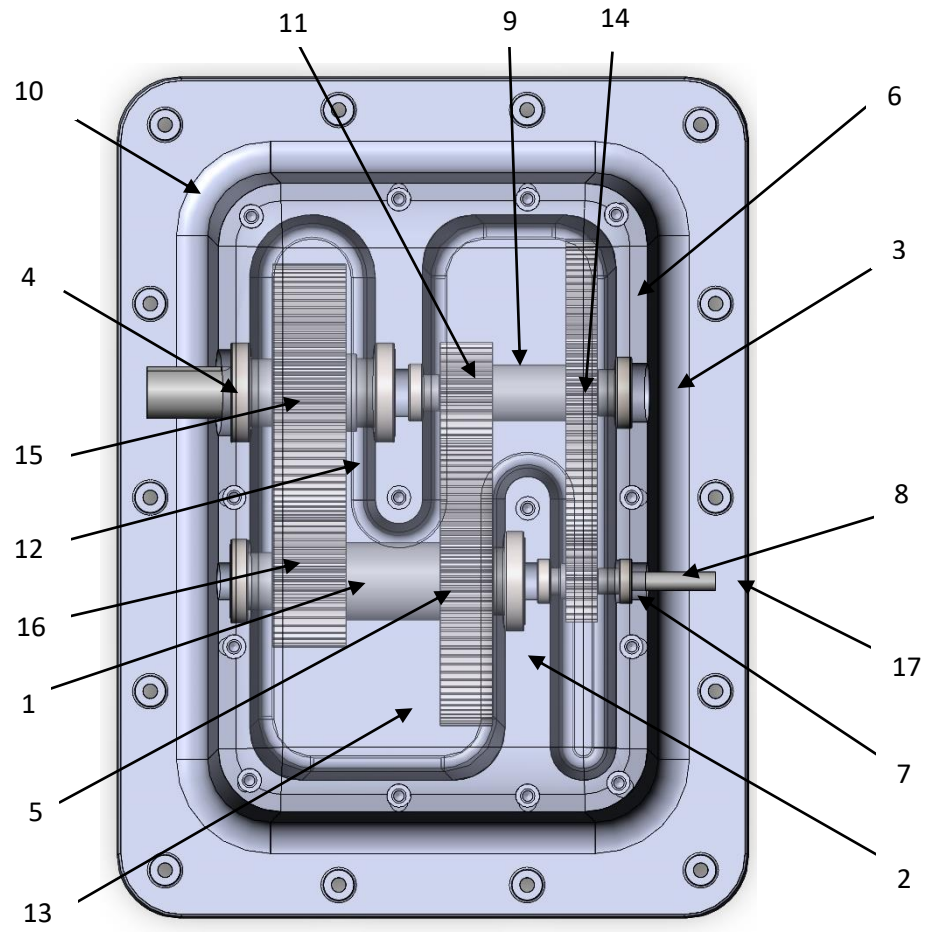
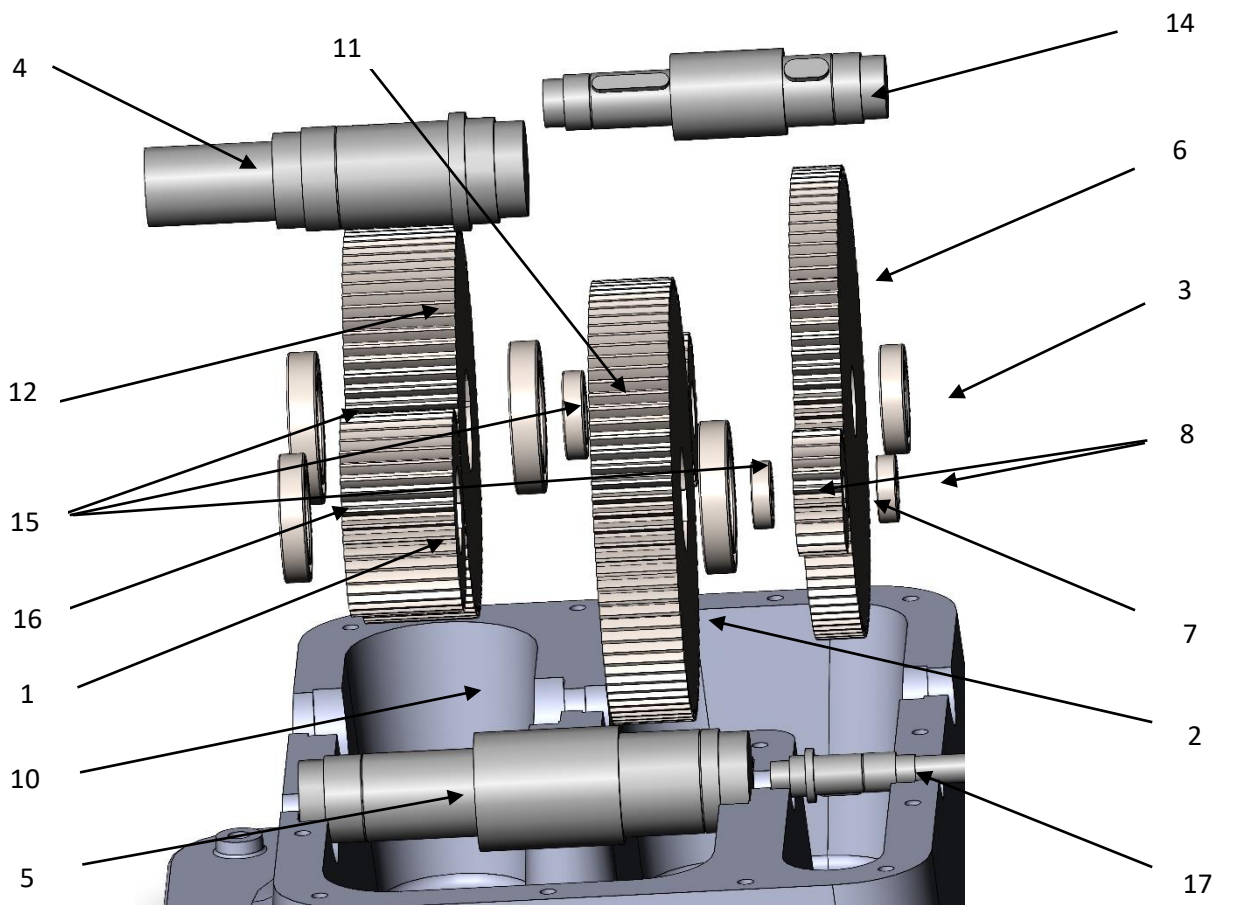


Рис.4.5. 3Д-модель редуктора.

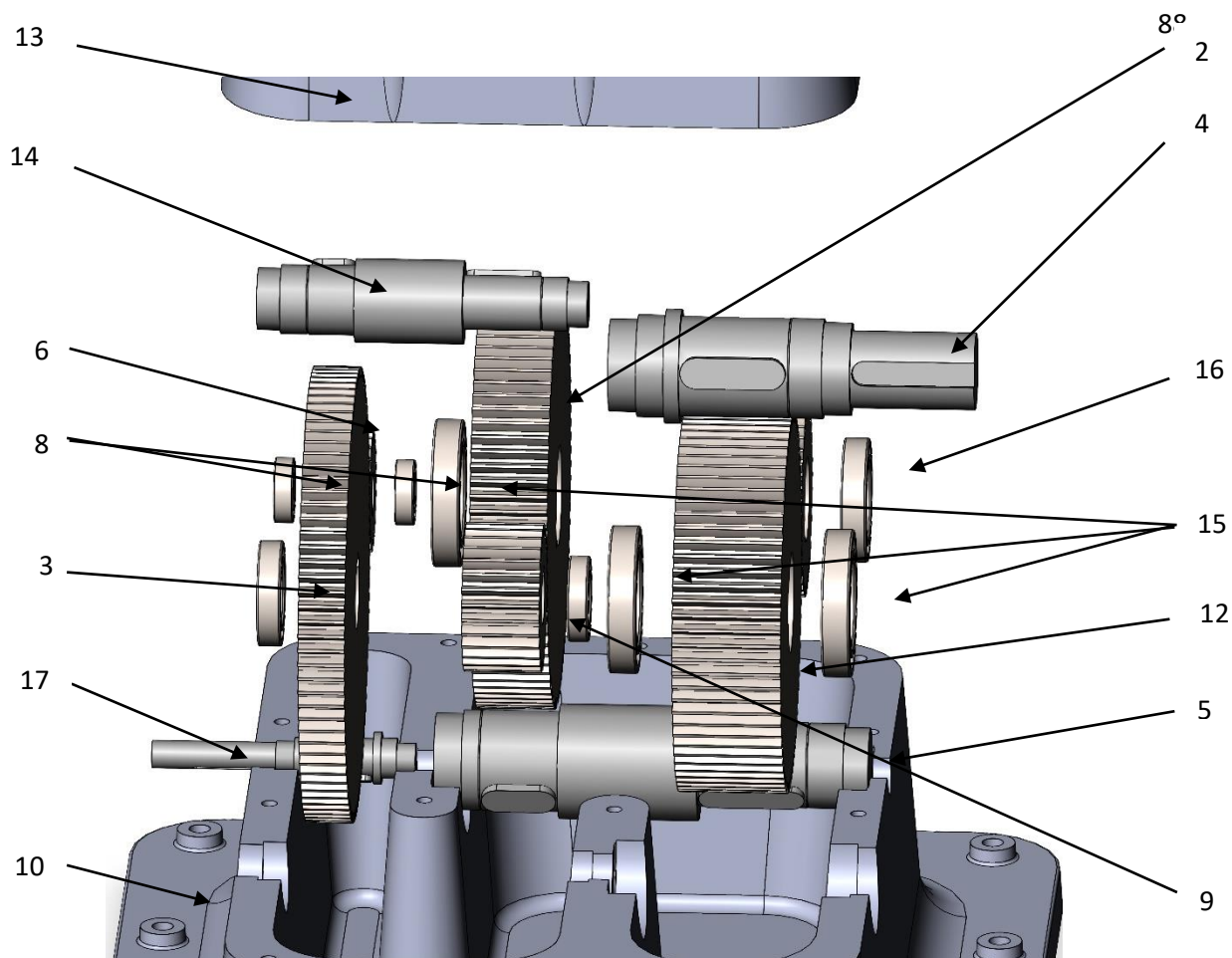
На рис.4.6 показано нумерацію деталей редуктора.



a)



b)



в)

Рис.4.6. 3Д-модель редуктора з нумерацією деталей:

а) вид зверху;

б) вид з рознесеними частинами спереду;

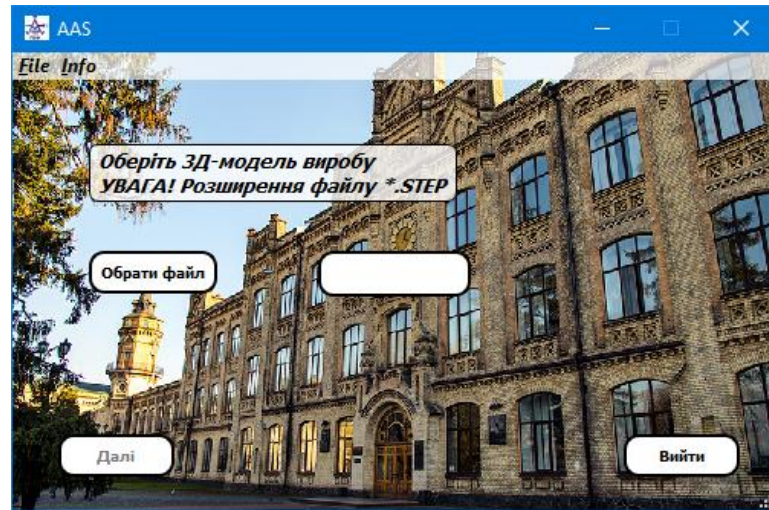
в) вид з рознесеними частинами ззаду;

де, 1 – Зубчате колесо Z40; 2 – Зубчате колесо Z90; 3 – Підшипник B6008;
4 – Вал s4; 5 – Вал s3; 6 – Зубчате колесо Z96; 7 – Зубчате колесо Z24; 8 –
Підшипник B6004; 9 – Зубчате колесо Z30; 10 – Корпус; 11 – Підшипник
B6006; 12 – Зубчате колесо Z80; 13 – Кришка; 14 – Вал s2; 15 – Підшипник
B6012; 16 – Підшипник B6010; 17 – Вал s1.

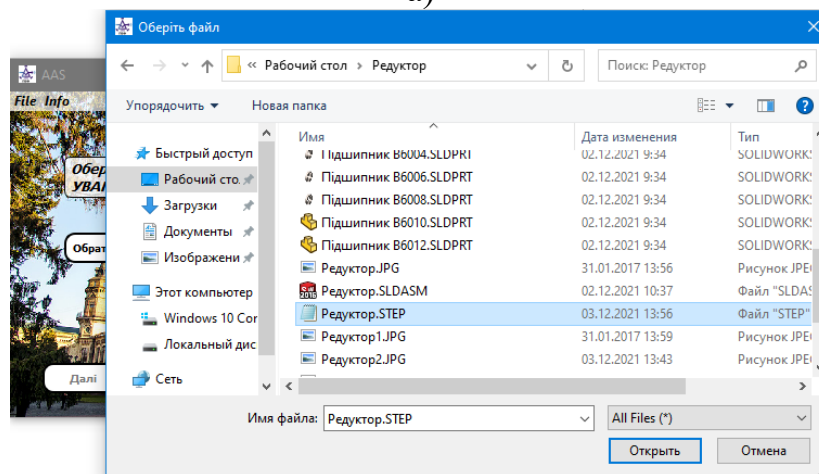
Запустивши програму користувач побачить перше робоче вікно (рис.4.7, а), далі користувач повинен натиснути кнопку «Обрати файл» та у відкритому вікні вибрати необхідний файл з розширення *.STEP або *.stp (рис.4.7, б). Після правильного вибору файлу буде доступна до нажаття кнопка «Далі» (рис.4.7, в), а після натиснення даної кнопки з'явиться друге робоче вікно (рис.4.8, а). В цьому вікні у підпункті план складання показана

матриця зв'язків деталей у приладі, рекомендований план складання, а також побудований на основі матриці і плану складання – граф складання (рис.4.8, б) для обраної 3Д-модулі, граф зберігається під назвою *saved_figure.png* у папці з програмою. Матриця зв'язків деталей у приладі формується на основі аналізу вибраної 3Д-моделі, аналіз моделі та створення матриці реалізовано на основі алгоритмів представлених у 3 розділі дисертації. Рекомендований план складання формується генетичним алгоритмом на основі матриці зв'язків. У підпункті другого вікна, під назвою виробничий процес (рис.4.8, в), на основі згенерованої послідовності складання система пропонує виробничий план складання для обраної моделі, але у користувача є можливість відредагувати автоматично сформовані дані у таблиці. Спочатку необхідно обрати всі необхідні операції, переходи та інструменти складання для двох вибраних деталей в рядку. Час на перехід розраховується автоматично і залежить від вибраної інформації [79]. Вся таблиця є повністю редагованою, також реалізована можливість додавання нових операцій або переходів. Після введення і вибору всієї необхідної інформації з'являється можливість створити операційну карту, вся інформація записується у файл з розширенням *.xlsm, який з легкістю можна редагувати у системі Microsoft Office Excel, в цій же системі є можливість зберегти згенеровану інформацію у формат *.pdf та просто використовувати в подальшому для створення документації.

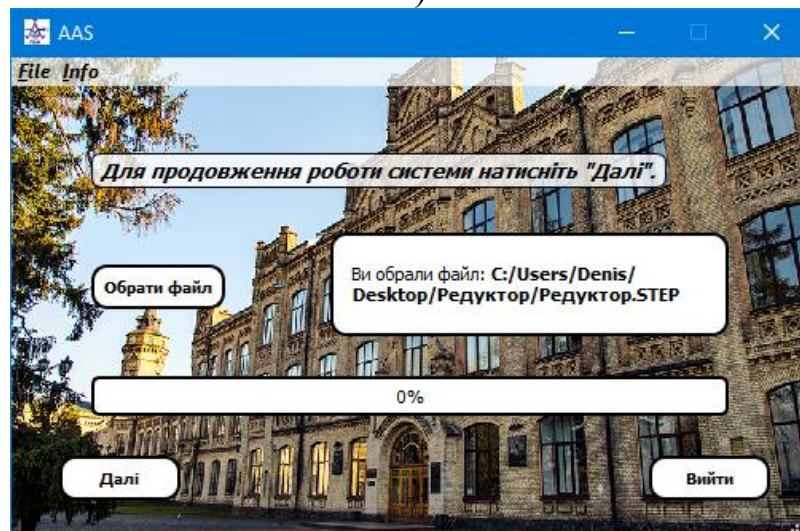
Згенеровану операційну карту можна побачити у додатку Б.



а)



б)



в)

Рис.4.7. Перше робоче вікно інтерфейсу програми:

а) перше вікно;

б) вибір файлу;

в) вікно із правильно вибраним файлом.

Отримані результати:

File Info

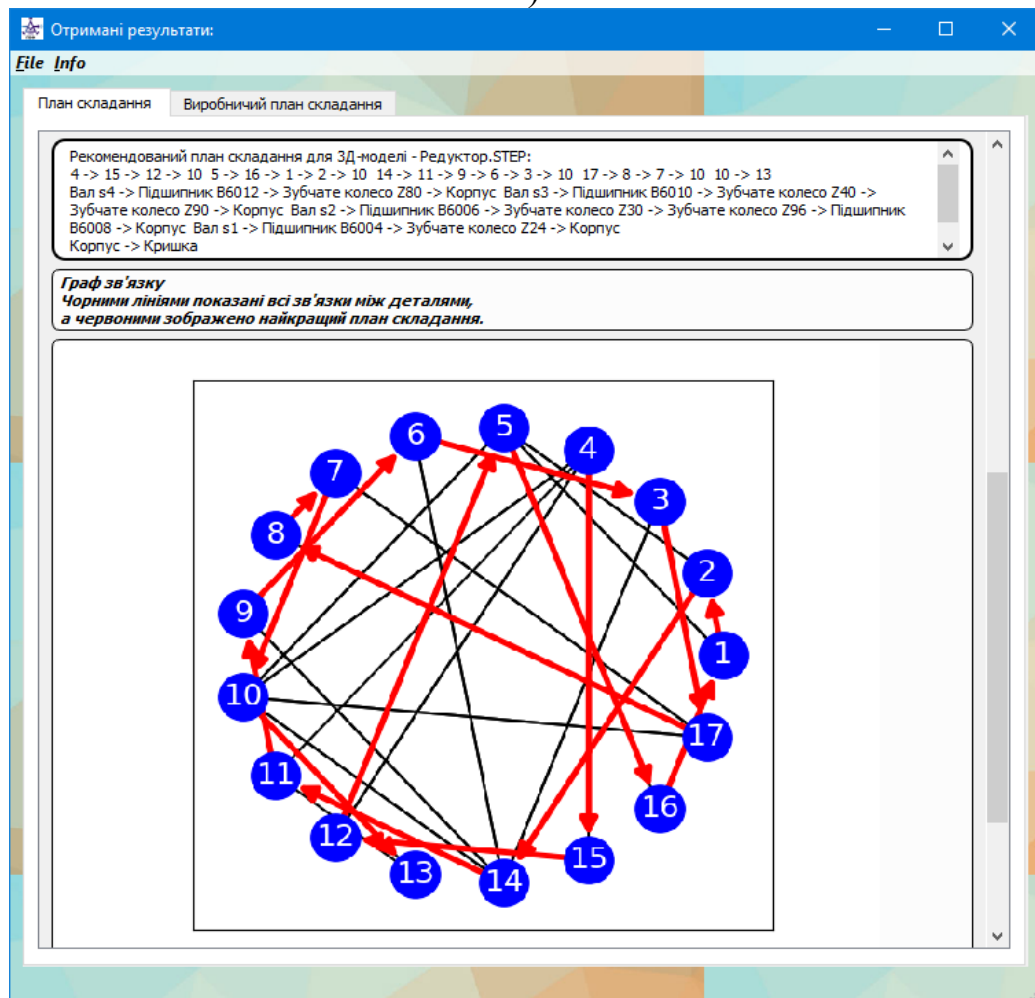
План складання Виробничий план складання

Матриця зв'язку складальних одиниць у файлі 3Д-моделі - Редуктор.STEP

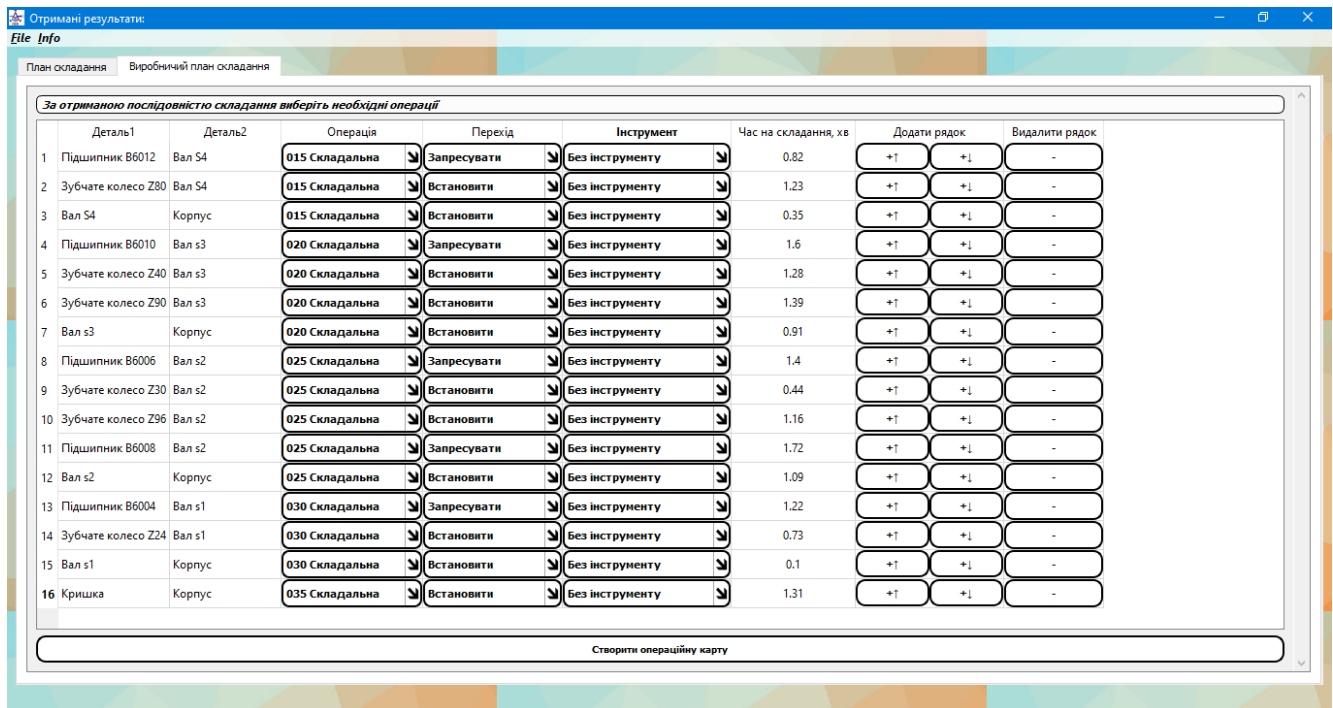
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0
5	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
14	0	0	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Рекомендований план складання для 3Д-моделі - Редуктор.STEP:

а)



б)



в)

Рис.4.8. Друге робоче вікно інтерфейсу програми:

а) друге вікно – план скаладання;

б) граф зв'язків деталей;

в) друге вікно – план скаладання.

Отже, як видно із рис.4.7-рис.4.8 програма має дуже легкий у розумінні інтерфейс (немає необхідності навчати людину для використання даною програмою), при цьому дає можливість на створення плану складання та виробничого плану у досить короткий час, що в свою чергу дає можливість скоротити час на створення документації по складанню виробу та скоротити сам час складання – це призведе до збільшення продуктивності на виробництві та скороченн затрат на нього.

ВИСНОВКИ ДО РОЗДІЛУ

У даному розділі було розглянуто інтерфейс, блок схему, структуру і приклад роботи створеної програми. Показано принцип та простоту роботи з розробленою програмою.

Повністю автоматизоване програмне забезпечення усуває необхідність взаємодії або втручання людини для створення необхідної геометричної вхідної інформації для планування складання, яка була потрібна попереднім

планувальникам складання. Розроблений планувальник складання приймає файл STEP як вхідні дані та автоматично виводить послідовність складання для даної моделі складання, а на основі цих даних дає можливість створити виробничий план.

В результаті запропонований повністю автоматизований планувальник складання підвищує ефективність планування послідовності складання, повністю усуваючи необхідність ручного введення під час процесу планування послідовності складання. Оскільки STEP є міжнародним стандартом, більшість програм САПР забезпечують можливості введення та виведення файлів STEP. В результаті запропоноване програмне забезпечення можна використовувати для аналізу моделей складання з більшості програм САПР. Користувачам просто потрібно запустити програму та імпортувати файл STEP, який вони хотіли б проаналізувати.

РОЗДІЛ 5. РОЗРОБКА СТАРТАП-ПРОЕКТУ «АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ СКЛАДАННЯ НА ОСНОВІ МЕТОДОЛОГІЇ SADT»

5.1. Опис ідеї проекту та аналіз ринкових можливостей запуску стартап-проекту

Розглянувши в попередніх розділах проектування послідовності складання на основі SADT, було створено автоматизовану систему проектування складання. В розділі буде проведено аналіз стартап-проекту, який визначить можливість даного продукту вийти на ринок і конкурувати з продуктами, що вже існують[80].

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Автоматизоване створення послідовності складання 3д-моделі	1. Застосування в виробництві, планування виробництва, розробці та впровадження технологічних процесів складання	Дозволяє використовувати систему на етапі послідовності технологічного процесу складання
	2. Проектування технологічних процесів	Підвищення ефективності проектування технологічних процесів складання

Отже, пропонується новий спосіб створення маршруту автоматизованої послідовності складання та створення операційної карти складання з розрахунком часу відпівідо до операцій в моделюванні технологічних процесів складання, дослідження цих процесів, вивчення можливостей оптимізації та підвищення ефективності функціонування.

Далі буде проведено аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів[80]:

- визначаємо попереднє коло конкурентів або товарів-замінників чи товарів-аналогів, що вже існують на ринку, проводимо збір інформації щодо техніко-економічних показників для ідеї

власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

- проаналізуємо проект для власної ідеї щодо сильних, слабких та нейтральних характеристик порівняно з конкурентами (табл. 5.2).

Конкурентами до розробленої програми визначено:

Конкурент 1 – Моделювання технологічних процесів складання в системі Autodesk Inventor;

Конкурент 2 – Моделювання технологічних процесів складання в системі DFA Product Simplification;

Конкурент 3 – Моделювання технологічних процесів складання в системі Creo Parametric.

Таблиця 5.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Технікоєкономічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент 1	Конкурент 2	Конкурент 3			
		1	2	3	4	5	6	7
1.	Створення бібліотек	+	-	-	-		+	
2.	Необмежені потоки випадкових чисел	+	+	+	+		+	
3.	Розробка власного інтерфейсу	+	+	+	+			+
4.	Імпорт 3Д-моделі	+	+	+	+		+	
5.	Використання будь-яких мов програмування	-	-	-	-	+		

Після порівняння характеристик проекту з конкурентами було визначено лише одна слабка характеристика – використання будь-яких мов програмування, однак ця характеристика є спільною для всіх представлених конкурентів, оскільки кожна система має свою спеціалізовану мову. Було визначено 4 сильні характеристики, а 1 з них є слабкою у конкурентів. Ідея проекту містить 3 нейтральних, 1 слабку та сильну сторони в порівнянні з конкурентами. Порівняльний аналіз характеристик і властивостей ідеї, показав, що власний проект має більше переваг.

5.2. Технологічний аудит ідеї проекту

В межах даного розділу було проведено аудит технології, за допомогою якої можна реалізувати ідею створення проекту. Бude визначено технологічну здійсненність ідеї проекту, яка передбачає аналіз таких складових (таблиця 5.3).

Таблиця 5.3 Технологічна здійсненність ідеї проекту

п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Проектування технологічних процесів складання	Традиційні мови програмування	Наявні, не потребують доопрацювань	Доступні
2		Спеціальні програмні бібліотеки вже існуючих систем	Наявні, потребують незначних доопрацювань	Доступні
3		Готові технологічні процеси складання	Наявні, потребують незначних доопрацювань	В повному об'ємі недоступні, існують обмежені версії

Проаналізувавши таблицю можна зробити висновок, що розроблений проект можна з легкістю реалізувати, тому що всі необхідні технології реалізації такі як: мова програмування, бібліотеки та технологічні процеси

складання, для цього наявні але необхідні невеликі доопрацювання для 2 та 3 технології реалізації, що в свою чергу принесе ще новизни в розроблений продукт.

Отже на основі аналізу стартап-проекта було сформовано інформаційну карту [81]:

Таблиця 5.4. Інформаційна карта стартап-проекту

Назва блоку	Характеристика
1	2
Загальна характеристика стартап-проекту	
Назва стартап-проекту	AAS(automated assembly sequence)
Проблематика, яку вирішує стартап-проект	Основною задачею є пришвидчення проектування технологічних процесів складання на виробництві та зменшення затрат на це.
Головні цілі та завдання проекту	Розробити програмне забезпечення.
Головні цільові групи, на які спрямований проєкт	Приладобудівні та машинобудівні підприємства.
Автори та команда стартап-проекту	
Автори стартап-проекту	Проскурєнко Д.М.
Команда стартап-проекту	Frontend-розробник, backend-розробник
Опис продукту стартап-проекту	
Назва та коротка характеристика мінімального життєздатного продукту стартапу (MVP)	Назва: AAS. Характеристики: можливість генерації послідовності складання.
Сфера застосування та функціональне призначення продукту	Продукт стартапу задовільняє потреби приладобудівні та машинобудівні підприємства в складанні продуктів.
Опис унікальних властивостей продукту стартапу	Легкий у розумінні інтерфейсу та швидкий у генерації інформації.
Стадія розробки продукту стартапу	Дослідний зразок. У подальшому планується удосконалення.

Продовження таблиці 5.4.

1	2
Технічні характеристики	Програмне забезпечення, реалізовано за допомогою Python, тому підходить під будь-які роздільності екранів. Необхідне встановлення Python, доступ до мережі Інтернет. ОС Windows7 та новіше, Linux-дистрибутиви.
Супровід продукту	Підтримка додатку, оптимізація
Забезпечення стартап-проєкту	
Необхідні ресурси	Потрібна команда розробників, сервер, фінансування для вдосконалення.
Потреба в інвестиціях	1000\$
Інтелектуальна власність	Описується об'єкт інтелектуальної власності (програмне забезпечення), форма його захисту, необхідна або наявна на пропонований продукт (авторське право). Зазначається власних об'єкту інтелектуальної власності: Проскурєнко Д.М.
Результати стартап-проєкту	
Термін реалізації стартап-проєкту	3 місяця.
Плановані кількісні показники стартап-проєкту	Продаж програми приладобудівним та машинобудівним підприємствам та забезпечення підтримки.
Якісні показники стартап-проєкту	Продукт повністю задовільняє необхідні характеристики.
Загальні очікувані результати	Створено програмне забезпечення для проєктування технологічних процесів складання на виробництві.

Визначаємо ринкові можливості, які можна використати під час ринкового впровадження проєкту, та ринкові загрози, які можуть перешкодити його реалізації. Цей аналіз дозволяє спланувати напрями розвитку проєкту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проєктів-конкурентів [81]. Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.5).

Таблиця 5.5. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	650 000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Невелика база технологічних процесів складання, представлення результатів проектування
5	Специфічні вимоги до стандартизації та сертифікації	Не змінна
6	Середня норма рентабельності в галузі (або по ринку), %	65%

За попередньою оцінкою ринок має 3 основних гравця, зростаючу динаміку і хороший попит на запропонований продукт.

Існує невелика база та висока норма рентабельності в галузі, тому можна зробити висновок, що ринок є привабливим для входження, хоча на ньому функціонують іноземні фірми, які працюють багато років і роблять продукцію хорошої якості, але за рахунок нової технології і переваг, які вона надає, продукт є конкурентоспроможним.

Далі визначаємо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи [81] (табл. 5.6).

Таблиця 5.6. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	1	2	3	4
1	Система проектування послідовності процесів складання	Підприємства з виготовленням продукції технологічного призначення	мова програмування ціна продукту наповненість бази системи -обслуговування	до продукції- можливість редагування; заміна бібліотеки на свою; до компанії постачальника: оновлення бібліотеки
2	Швидкість та ефективність	Підприємства з виготовленням продукції технологічного призначення	- швидкість обробки інформації -оптимальність обробки інформації	до продукції: - простота використання; до компанії постачальника: - оперативність налаштування.
3	Результати проектування	Підприємства з виготовленням продукції технологічного призначення	-використання людиною, яка не володіє всіма відповідними знаннями; - наявність простого інтерфейсу;	до продукції: простота використання; доступний інтерфейс; до компанії постачальника: швидке налаштування.

Отже потенційною групою клієнтів продукту мають стати приладобудівні або машинобудівні підприємства, фірми з різною кількістю робочих центрів і з різною кількістю замовлень, які побачать економічну вигоду у автоматизованому проектуванні послідовності процесів

складання виробів та автоматизованому створенні операційної карти за новою методикою.

При застосуванні даної технології існують певні загрози. Для попередження таких ситуацій необхідно якісне обладнання, а також працювати з такими програмами повинні висококваліфіковані фахівці. Необхідною умовою є своєчасне технічне обслуговування даного продукту (таблиця 5.7).

Таблиця 5.7. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Збут	покупець визначає постачальників за ефективністю роботи розробленої системи	Необхідно пропонувати моделі в невеликі компанії, які не бажають витратити великі гроші на дані системи
2	Корисність	Може не відповідати вимогам певних підприємств	Пропонувати методику використання розробленої системи впродовж певного часу безкоштовно
3	Конкуренція	Наявність аналогічних систем	Пропонувати системи на підприємства, які мають курс на оновлення виробництва
4	Обладнання	Застаріле програмне забезпечення	Застосування нового програмного забезпечення
5	Результативність	Виводить тільки кінцеві результати роботи	Надавати роз'яснення для виведених результатів

В таблиці 5.7 було визначено фактори загроз, які перешкоджають ринковому впровадженню розробленого проекту. Проблеми можуть виникнути у збуті, корисності, конкуренції, обладнанні та результативності. Було описано зміст кожної загрози, а також можливу реакцію на фактор щоб звести до мінімуму його негативний та небажаний вплив.

Реакцією компанії є рішення, що враховують специфіку загрози та пропонують підходи для недопущення порушення роботи проекту. Фактори загроз не є значними і не заважатимуть реалізації даного продукту.

Поряд із колом загроз існують і певні можливості для проекту (таблиця 5.8).

Таблиця 5.8. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Наукова новизна	Оновлення на будь-якому етапі роботи	Проводити оновлення та налаштування системи
2	Корисність	Може бути важливим доповненням для проектування послідовності процесів складання	Пропонування системи на підприємства, що оновлюють підходи до виробництва
3	Метод проектування	Оптимальний метод для проектування послідовності процесів складання	Спробувати використовувати інші методи послідовності процесів складання
4	Удосконалення продукту	Вдосконалення продукту може здійснити працівник компанії	Модернізація продукту
5	Швидкий ріст	Збільшення кількості продажів через бажання клієнтів оновлювати підходи до виробництва	Розширення ринку збуту, партнерські програми
6	Економічні	Підтримка інноваційного виробництва	Підвищення/пониження ціни на продукт.

В таблиці 5.8 було визначено фактори можливостей які сприяють ринковому впровадженню розробленого програмного забезпечення. Такими факторами є: наукова новизна, корисність, метод проектування, удосконалення продукту, швидкий ріст та економічний фактор. Було описано зміст можливостей для кожного з факторів та необхідні реакції компанії на той чи інший фактор для забезпечення його впровадження.

Далі було проведено аналіз конкуренції на ринку (таблиця 5.9). Для цього потрібно визначити тип конкуренції, рівень конкурентної боротьби,

залузову ознаку конкуренцію за видами товарів, характер конкурентних переваг та інтенсивність [82].

Таблиця 5.9. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - олігополія	Характеризується кількістю фірм від 2 до 10, обгороджених бар'єрами, які перешкоджають вступу в галузь нових фірм, мають контроль над цінами, але при змові з іншими представниками	Пропонувати системи на підприємства, які зацікавлені в оновленні проектування, інноваціях у виробництві
2. За рівнем конкурентної боротьби- локальний	Місцем використання системи	Пропонувати різні види наповнень бібліотек даних
3. За галузевою ознакою - внутрішньогалузева	Однією галуззю використання	Пропонувати різні види наповнень бібліотек даних
4. Конкуренція за видами товарів: - товарно-видова	Товарами одного виду	пропоонувати різні види наповнень бібліотек даних
5. За характером конкурентних переваг - нецінова	За допомогою вдосконалення якості продукції, технології виробництва, патентування і брендування та умов її продажу	Використовувати різні системи моделювання для порівняння результатів
6. За інтенсивністю - не марочна	Торгова марка дуже не значна, проте може бути присутньою на ринку	Створити бренд, легкий для запам'ятовування

В даній таблиці було проаналізовано ринок збуту розробленого продукту і визначено загальні риси конкуренції на ньому. Було досліджено особливості конкурентного середовища [82]. Визначено як проявляється кожна характеристика та необхідні рішення, які дозволять мінімізувати негативний вплив на підприємство та дозволити йому бути

конкурентоспроможним. Тепер необхідно провести більш детальний аналіз умов конкуренції в галузі (таблиця 5.10).

Таблиця 5.10. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
Складові аналізу	- Autodesk Inventor; - DFA Product Simplification - Creo Parametric.	-Extend -Simulink	-США -СНГ -Європа	Приладобудівні підприємства	Системи побудовані за допомогою інших засобів
Висновки	Інтенсивність з боку прямих конкурентів – є значною	- існує можливість виходу на ринок -є потенційні конкуренти Строки виходу системи на ринок – 2021-2022	Постачальники встановлюють вимоги роботи на ринку - відповідне оформлення системи; - гарантії	Клієнти встановлюють вимоги роботи на ринку: -наповнення бібліотек; -простота інтерфейсу	Обмеження для роботи через товари заміники

Аналіз показав, що головними силами, які впливають на конкуренцію в галузі є постачальники та споживачі[81]. Велике значення має інтенсивність конкуренції між існуючими конкурентами. Прямими конкурентами є три системи, що займаються моделюванням процесів: Autodesk Inventor, DFA Product Simplification, Creo Parametric. Потенційними конкурентами можуть бути Extend та Simulink. Зроблено висновки по кожному з конкурентів, постачальників, клієнтів та товарів-замінників. Сильні позиції проекту за кожним з факторів означають її можливості забезпечити необхідні темпи обороту капіталу та її здатність впливати на інших учасників ринку, диктуючі їм власні умови співпраці.

Фактори конкурентоспроможності розглянуто в таблиці 5.11, а порівняльний аналіз сильних та слабких сторін моделювання технологічних процесів складання наведено в таблиці 5.12[82].

Таблиця 5.11. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Оболонка	В наявності
2	Інтерпретатор	Здійснює перетворення інформації
3	Рівень мови створення	Використано Python
4	Можливість редагування	В головному коді програми
5	Оновлення бібліотек	В головному коді програми, за нестачі інформації
6	Простота використання	Можна використовувати не будучи спеціалістом в даній предметній області

Таблиця 5.12. Порівняльний аналіз сильних та слабких сторін моделювання технологічних процесів складання

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з моделюванням ТП						
			-3	-2	-1	0	+1	+2	+3
1	Оболонка	2					+		
3	Метод представлення	1				+			
4	Можливість доповнення та редагування	1					+		
5	Оновлення бібліотек	1				+			
6	Простота використання	5				+			
7	Підхід побудови алгоритмів в бібліотеках	1					+		
8	Тип набутих знань	2						+	
9	Рівень мови придбання знань	1					+		

Фактори конкурентноспроможності є обґрунтованими та є важливими для порівняння з конкурентами. Порівняльний аналіз сильних і слабких сторін показав, такі показники, як: надійність, довговічність та час роботи, за таку ж саму ціну розроблений продукт, дає перевагу над іншими продуктами і тому проект може стати конкурентоспроможним на ринку [82]. Далі проведемо аналіз сильних і слабких сторін проекту (таблиця 5.13).

Таблиця 5.13. SWOT- аналіз стартап-проекту

Сильні сторони: • використання спеціалізованих мов програмування; • імпорт 3Д-моделі; • можливість вдосконалення продукту.	Слабкі сторони: • розробка власного інтерфейсу; • недостатня кількість навчених розробників
Можливості: • корисність; • ріст попиту; • отримання нових замовлень; • збільшення продажів;	Загрози: • збут; • конкуренція; • поява нових технологій у конкурентів; • результативність.

В таблиці 5.13 було проведено аналіз сильних та слабких сторін проекту. Ринкові загрози та ринкові можливості складаємо на основі факторів загроз і можливостей, які ми складали раніше.

Ринкові загрози та можливості на відміну від факторів ще не є реалізованими на ринку та мають певну ймовірність здійснення[82].

На основі SWOT-аналізу розробляємо альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (таблиця 5.14) [80].

Таблиця 5.14. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Стратегія нейтралізації ринкових загроз	80%	1 рік
2	Стратегія заміни недоліків стартапу наявними ринковими можливостями	60%	6 місяців

Проведено аналіз розроблених альтернатив ринкового впровадження і з зазначених альтернатив обираємо ту, яка має найбільшу ймовірність отримання ресурсів, а також є найшвидшою в реалізації. Обираємо стратегію нейтралізації ринкових загроз, сильними сторонами стартапу наявними ринковими можливостями [82].

5.3. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.15. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Приладобудівні підприємства	65%	80%	80%	15%
2	Машинобудівні підприємства	60%	75%	80%	10%
Які цільові групи обрано: Приладобудівні та машинобудівні підприємства					

За результатами аналізу потенційних груп споживачів обрано цільові групи, для яких будемо пропонувати свою програму для оптимізації робочих центрів – приладобудівні та машинобудівні підприємства.

Визначено стратегію охоплення ринку: стратегію концентрованого маркетингу. Для роботи в обраному сегменті ринку необхідно сформувати базову стратегію розвитку (табл.5.16).

Таблиця 5.16. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Стратегія спеціалізації	Стратегія спеціалізації	Передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок, ціль - задоволення потреб вибраного цільового сегменту краще, ніж конкуренти	Стратегія спеціалізації

За базову стратегію розвитку було взято стратегію спеціалізації, що передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок [80]. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту [80]. Проте низька ринкова доля у разі невдалої реалізації стратегії може істотно підірвати конкурентоспроможність компанії. Зосередження на одному сегменті дасть переваги для вирішення недоліків даного проекту та покращить стан конкурентоспроможності на ринку. Наступним кроком є вибір стратегії конкурентної поведінки (табл. 5.17).

Таблиця 5.17. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	Ні	Пошук нової цільової аудиторії.	Так Покращення візуального вікна та можливостей роботи з 3Д- моделями	Стратегія заняття конкурентної ніші

За базову стратегію конкурентної поведінки прийнято стратегію зайняття конкурентної ніші - компанія в якості цільового ринку вибирає один або декілька ринкових сегментів малого розміру. Головне завдання компанії при цьому – це постійна турбота про підтримку і розвиток своєї конкурентної переваги, формування лояльності і прихильності споживачів, підтримка вхідних бар'єрів.

На основі вимог споживачів з обраного сегменту до постачальника і продукту, а також в залежності від стратегії розвитку та стратегії конкурентної поведінки розроблено стратегію позиціонування яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проект [81].

Таблиця 5.18. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	База бібліотек	Стратегія спеціалізації	-Метод отримання нових даних -наукова новизна	Позиціонування за співвідношенням «ціна-якість» Позиціонування за сферою використання - Стратегія позиціонування за однією ознакою
2	Ефективність та швидкість оновлення	Стратегія спеціалізації	-Корисність -Своєчасне оновлення	
3	Представлення результатів проектування	Стратегія спеціалізації	-Вид представлення	

Результатом даного підрозділу є система рішень щодо ринкової поведінки, вона визначає в якому напрямі буде працювати компанія на ринку.

5.4. Розроблення маркетингової програми стартап-проекту

Під час розроблення маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач [82]. У таблиці 5.19 підсумовуємо результати аналізу конкурентоспроможності товару.

Таблиця 5.19. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Оболонка	Є в наявності	Можливе покращення
2	Інтерпретатор бібліотек	Здійснює швидке та якісне перетворення інформації	+
3	Метод представлення знань	Метод продукцій	Можливе замінення методу в разі необхідності
4	Підхід пошуку інформації в необхідних бібліотеках	Прямий ланцюжок міркування	+
5	Рівень мови придбання знань	Використання Python	+
6	Тип набутих знань	Символьний	Необхідно додати, табличний та зчитування рисунків
7	Оновлення бібліотек	В головному коді програми	Поповнення новими матеріалами
8	Простота використання	Можна використовувати не будучи спеціалістом в даній предметній області	+

Було визначені потреби та вигоди від них. Вибрано шляхи створення ключових переваг товару перед конкурентами. За рахунок ключових переваг товару і стратегії диференціації, що передбачає надання товару важливих з точки зору споживача відмінних властивостей за такою ж ціною

як і у конкурентів, буде розроблено маркетингову програму стартап-проекту [82].

Таблиця 5.20. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Створення послідовності процесу складання
II. Товар у реальному виконанні	Властивості/характеристики
	Комп'ютерна програма; Можливість підвищення швидкості та якості створення ТП; Зниження вартості та доступність застосування системи;
	Підвищення ефективності застосування апаратури
III. Товар із підкріпленням	До продажу: рівень розробки.
	Після продажу: консультування, обслуговування.
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок наявності авторського права.	

В таблиці 5.20 створено трьохрівневу модель нашого товару, що включає задум товару та його вигоди, основні характеристики готового товару, спосіб його пакування та захисту від копіювання та плагіату [82]. Отже розроблений продукт буде захищено за рахунок наявності авторського права, що в свою чергу дасть повну захищеність в межах країни, але для виходу на міжнародний ринок потрібно буде отримати захист і там.

Наступним кроком є визначення цінових меж, якими необхідно керуватися при встановленні ціни на потенційний товар, це передбачає аналіз цін товарів конкурентів, та доходів споживачів продукту (табл. 5.21).

Таблиця 5.21. Визначення меж встановлення ціни

№ п/п	Рівень цін на товаризамінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	800 грн	8000грн	4000грн	1500-3500 грн

В таблиці проаналізовано ринкові ціни на товари аналоги та замітники, а також середній рівень доходів споживачів. За отриманими даними буде встановлена верхня та нижня межа на розроблену програму [82].

Таблиця 5.22. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Звична купівля з деяким змінами, або модифікована закупівля, придбання дещо змінених товарів (послуг), або зміну ціни на товар (послугу), або зміну кількості постачання).	Доставка товару покупцю, встановлення та налаштування.	Канал нульового рівня	Власна система збуту. Виробник безпосередньо продає товар клієнту і використовує три способи прямого продажу: -торгівля через магазини -посилкова торгівля -онлайн сервіси

Спираючись на специфіку закупівельної поведінки цільових клієнтів - звична купівля з деяким змінами, або модифікована закупівля, було обрано власну систему збуту та визначено її функції, коли виробник безпосередньо продає товар клієнту через торгівлю в магазинах, посилками чи в онлайн.

Також за глибину каналу збуту було обрано канал нульового рівня, тому що компанія хоче мати тісні контакти із споживачами на обмеженому цільовому каналі [80].

Таблиця 5.23. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Бути на зв'язку	- електронна пошта - моб. Телефон - месенджери	Проектування послідовності процесів складання	Зацікавити у виборі проекту	«Ваші проблеми – наші рішення»

Простота використання продукту та зменшення часу є головною перевагою продукту, за рахунок якого він буде кращим на ринку. Основним завданням буде зацікавити клієнта в продукті за рахунок надання можливості використовувати його відповідний час безкоштовно [81]. Також завдяки цьому можна буде отримати відповідний відгук від користувачів, що в свою чергу дасть можливість удосконалити продукт та виправити помилки.

Бізнес-модель — це план вашого бізнесу щодо заробітку грошей. Це пояснення того, як ви надаєте своїм клієнтам цінність за належну вартість. Модель включає описи продуктів або послуг, які ви плануєте продавати, ваш цільовий ринок і будь-які необхідні витрати [82].

Бізнес-модель дозволяє підприємцям експериментувати, тестувати та моделювати різні способи структурування витрат і потоків доходів. Для тих, хто тільки починає, вивчення потенційних бізнес-моделей може допомогти визначити, чи є ваша бізнес-ідея життєздатною, залучити інвесторів і керувати загальною стратегією управління. Для відомих компаній модель служить основою для розробки фінансових прогнозів, встановлення етапів та встановлення базової лінії для перегляду вашого бізнес-плану.

Отже, за всім вище переліченим було розроблено бізнес модель (таблиця 5.24).

Таблиця 5.24. Бізнес модель

Ключові партнери  Партнер з питань реклами - має зв'язки з телеканалами. Партнер з розміщенням серверів. Партнер з фінансових питань. Мотивація для партнерства Оптимізація та економія Зменшення ризиків та невизначеності Отримання конкретних ресурсів та дій	Ключові види діяльності  Канали продаж: промислові виставки-ярмарки, сайт компанії-розробника. Зв'язок з покупцями відбувається через сайт, рекламу на інших сайтах. Категорії Виробництво кращих версій продукту; Розв'язання проблем зменшення розриву між теоретичними та практичними знаннями майбутнього фахівця Ключові ресурси  Матеріальні: точки продажу фінансування, Людські: випробування продукту, маркетинг, розробка покращеного продукту та програм до нього Типи ресурсів Інтелектуальні – патент на бренд. Фінансові	Ціннісна пропозиція  Програмне забезпечення є повністю автоматизованим і швидко вирішує поставлену перед ним задачу, на відміну від конкурентів. Забезпечує пришвидчення часу на проектування технологічних процесів складання та зменшує час на складання виробу. Індивідуальна пропозиція – за попередньою домовленістю змінюється база даних.	Відносини з клієнтами  Самообслуговування та автоматизоване обслуговування. Сторінки у соціальних мережах. Відгуки допомагають вносити кращі пропозиції в вдосконалення продукту. Підтримка активних тем модераторами. Цінність створюється спільно зі споживачем Канали  Установка та налаштування програмного забезпечення на комп'ютери замовника здійснюється спеціалістом із стартап-проекту.	Сегменти користувачів  Продукт розрахований на приладобудівні та машинобудівні підприємства де задовільняє потреби користувача у автоматизації технологічних процесів проектування складання. Ринок Нішевий ринок
Структура витрат  Найбільш значні затрати пішли на розробку бази даних та сервера. Залежність від витрат на витратні матеріали, автоматизацію спілкування з клієнтами, дослідження ринку Фінансові витрати (зарплата, комунальні послуги)		Джерела доходів  Оплата здійснюється на банківський рахунок компанії електронними та готівковими платежами Типи: Продаж активів. Оренда. Реклама Фіксована ціна: Ціна по прейскуранту. Залежить від властивостей продукту. Ціна залежно від обсягу		

ВИСНОВКИ ДО РОЗДІЛУ

Ринкова комерціалізація плану даного проекту заснована на наявному попиту, динаміці ринку та рентабельності роботи на ринку. При використанні створеної системи моделювання відбудеться підвищення ефективності та оптимізація технологічного процесу складання виробів на приладобудівному підприємстві, а також пришвидчиться розробка технологічної документації та створення нормативів. Такий оптимізований процес матиме кращі властивості при роботі порівняно із продуктами конкурентів.

Бар'єром входження на ринок є певна кількість товарів аналогів, тобто схожих за призначенням систем моделювання. Проте порівняно із ними дана система оптимізує роботу технологічного процесу ефективніше, є повністю автоматизованою, надає можливість швидкого переналагодження системи, тісний зв'язок із користувачами, можливість внесення змін для створення або покращення факторів, що надають перевагу над конкурентами, за рахунок чого вона може стати конкурентоспроможною на ринку.

Для впровадження на ринок була розроблена система на основі стратегії зайняття конкурентної ніші, якщо певна компанія в якості цільового ринку вибирає один чи декілька ринкових сегментів невеликого розміру. Було визначено ключові переваги концепції потенційного товару, що дозволить проекту конкурувати на зайнятій ринковій ніші. Був виконаний опис трьох рівнів моделі товару та визначено межі встановлення ціни. Для реалізації продукції сформовано систему збуту та розроблено відповідну концепцію маркетингових комунікацій.

Отже, подальша імплементація продукту є доцільною за рахунок сильних сторін продукту. Вивчені фактори можливостей і загроз та відповідні реакції на них, що враховуючи наявний попит на ринку в зайнятій ніші, дозволяє вважати проект конкурентоспроможним в порівнянні з заміниками даного продукту.

ЗАГАЛЬНІ ВИСНОВКИ ДО МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ

На планування складання припадає більше 50% загальних витрат, тому генерація послідовності складання є одним з основних компонентів у плануванні складання. Традиційно досвідчені інженери створюють послідовність складання для складання на основі свого попереднього досвіду. Автоматизація генерації послідовності складання призведе до значної економії часу та грошей для конструктора. Щоб автоматизувати процес складання, потрібна контактна інформація та інформація розташування різних деталей у виробі. Дані САПР беруться як вхідні дані для визначення просторового розташування деталей відносно один одного. Графік зв'язку представляє контактні відносини між різними деталями в виробі. На основі даних про взаємозв'язок деталей виробу та за допомогою генетичного алгоритму можна отримати рекомендований план послідовності складання, який далі можна використати для створення виробничого плану складання.

Отже, у проведеному дослідженні був розроблений повністю автоматизований планувальник послідовності складання, який потім використовувався для пошуку оптимальних або майже оптимальних послідовностей складання для реальної моделі, а потім використовувався для розробки та аналізу потенційних процесів складання. З цього дослідження можна зробити наступні висновки:

1. Робота з підготовки вручну, необхідна в уже створених програмних продуктах, вилучення геометричної інформації, що міститься у файлах проектування САПР, а потім переклад геометричної інформації в розпізнавану комп'ютером інформацію про просторові обмеження, може бути повністю комп'ютеризована.
2. Геометричну інформацію, яка описує моделі складання САПР розроблених виробів, можна зберегти у файлах STEP. Потім збережені файли STEP можна використовувати як вхідні файли

для повністю автоматизованого планувальника послідовності складання, який повністю автоматизує підготовчі роботи. Використання формату файлу STEP також може зробити отриманий планувальник складання в цілому корисним для різних систем САПР.

3. Генерування можливих послідовностей складання за допомогою генетичного алгоритму, засновано на просторових відносинах між усіма частинами моделі, можна інтегрувати безпосередньо з комп'ютеризованим аналізом інформації для автоматичного знаходження оптимальних або майже оптимальних послідовностей складання.
4. Використання розробленого програмного забезпечення для визначення послідовності складання може скоротити загальний час складання в реалістичному процесі складання, який використовує робота або оператора. А система проектування процесів складання надає можливість пришвидшити створення документації.

Тож було розроблено автоматизований планувальник послідовності складання на основі методології SADT, який безпосередньо приймає файли STEP як вхідні дані та усуває необхідність взаємодії людини під час підготовки до планування послідовності складання та створення виробничих процесів складання. Для цього було проведено дослідження математичної моделі 3д моделі САПР, функціональної моделі виробничого процесу складання; було розроблено метод вилучення даних САПР, необхідних для створення матриці зв'язків деталей у приладі, із файлів STEP, а на основі цих даних та завдяки генетичному алгоритму було розроблено алгоритм для оптимізації послідовності складання. Далі використовуючі всі вже отримані дані розробляються виробничі процеси складання.

На основі розробленої програми було створено стартап-проект; визначено основних конкурентів та плюси продукту перед ними; проаналізовано всі загрози та методи їх вирішення; визначено базової концепції розвитку; для впровадження на ринок була розроблена система на основі стратегії зайняття конкурентної ніші. Отже, було визначено, що подальша імплементація продукту є доцільною, а проект можна вважати конкурентоспроможним в порівнянні з замінниками даного продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nof, S. Y., Wilhelm, W. E., and Warneke, H.-J., 1997, Industrial Assembly (London: Chapman and Hall).
2. Дэвид А. Марка, Клемент МакГоуэн. SADT-методология структурного анализа и проектирования. -М.: Метатехнология, 1993.
3. Юлия, Никитенкова, Александр Горшенин, Использование методологии SADT / Юлия Никитенкова, Александр Горшенин. - М.: LAP Lambert Academic Publishing, 2015. - 100 с.
4. Усков А.А., Жукова А.Г. Матричное представление иерархических визуальных моделей для CASE-средств // Информационные технологии. 2014. № 11. С. 36–39.
5. Усков А.А., Жукова А.Г. Подход к оценке сложности диаграмм SADT (IDEF0) // Программные продукты и системы. 2015. № 1. С. 34–37.
6. Cauffriez L, Renaux D, Bonte T, Cocquebert E. Systemic modeling of integrated systems for decision making early on in the design process. Cybernetics and Systems 2013; 44:1–22.
7. Tseng HE, Tang CE (2006) A sequential consideration for assembly sequence planning and assembly line balancing using the connector concept. Int J Prod Res 44(1):97–116
8. Qin YF, Xu ZG (2007) Assembly process planning using a multi-objective optimization method. In: Proceedings of the 2007 IEEE international conference on mechatronics and automation, ICMA 2007, 4303610, pp 593–598
9. Sinanoglu C, Boklu HR(2005) An assembly sequence-planning system for mechanical parts using neural network. Assem Autom 25(1):38–52
10. Chen G, Zhou J, Cai W, Lai X, Lin Z, Menassa R (2006) A framework for an automotive body assembly process design system. CAD Comput Aided Des 38(5):531–539

- 11.Chen WC, Tai PH, Deng WJ, Hsieh LF (2008) A three-stage integrated approach for assembly sequence planning using neural networks. *Expert Syst Appl* 34(3):1777–1786
- 12.Mitrovic-Minic S, Krishnamurti V (2006) The multiple TSP with time windows: Vehicle bounds based on precedence graphs. *Oper Res Lett* 34(1):111–120
- 13.Wang Y, Liu JH (2010) Chaotic particle swarm optimization for assembly sequence planning. *Robot Comput Integr Manuf* 26(2):212–222
- 14.De Mello, L. S. H., and Sanderson, A. C. (1991). Representations of mechanical assembly sequences. *IEEE Transactions on Robotic and Automation*, 7(2), 211–227.
- 15.Mantripragada, R., and Whitney, D. E. (1998). The datum flow chain: A systematic approach to assembly design and modeling. *Research in Engineering Design*, 10(3), 150–165
- 16.De Mello, L. S. H., and Sanderson, A. C. (1989) A correct and complete algorithm for the generation of mechanical assembly sequences. In: *Proceedings IEEE international conference on robotics and automation (ICRA)*, (vol. 1, pp. 56–61).
- 17.Wolter, J. (1991). A combinatorial analysis of enumerative data structures for assembly planning. In: *Proceedings of the IEEE international conference on robotics and automation* (pp 611–618)
- 18.Romney, B., Godard, C., Goldwasser, M., and Ramkumar, G. (1995). An efficient system for geometric assembly sequence generation and evaluation. In: *Proceedings of the ASME international computers in engineering conference* (pp. 699–712)
- 19.Wilson, R., and Rit, J. (1990) Maintaining geometric dependencies in an assembly planner. In: *Proceedings of the 1990 IEEE international conference on robotics and automation* (vol. 2, pp. 890–895) Scottsdale, AZ

20. Thomas, U., Barrenscheen, M., and Wahl, F. (2003). Efficient assembly sequence planning using stereographical projections of c-space obstacles. In: Proceedings of the 2003 IEEE international symposium on assembly and task planning (ISATP 2003) (pp. 96–102).
21. Lee, B., and Saitou, K. (2003). Decomposition-based assembly synthesis for in-process dimensional adjustability. *Journal of Mechanical Design*, 125(3), 464–473
22. Bourjault, A. (1984). Contribution a une approche methodologique de l'assemblage automatise: Elaboration automatique des sequences operatoires. PhD thesis, Universite de Franche-Compte.
23. Fazio, T. L. D., and Whitney, D. E. (1987). Simplified generation of all mechanical assembly sequences. *IEEE Journal of Robotics and Automation*, RA-3(6), 640–658
24. Bonneville, F., Henrioud, J., and Bourjault, A. (1995). Generation of assembly sequences with ternary operations. *IEEE international symposium on assembly and task planning* (pp. 245–249)
25. Naphade, K. S., Storer, R. H., and Wu, S. D. (1999a). Graph-theoretic generation of assembly plans, part i: Correct generation of precedence graphs. Technical Report IMSE Technical Report 99T-003, Lehigh University, Bethlehem, Pennsylvania
26. Huang, Y., and Lee, C. (1991) A framework of knowledge-based assembly planning. In: *IEEE international conference on robotics and automation*, (vol. 1, pp. 599–604)
27. Goldwasser, M. H., and Motwani, R. (1999). Complexity measures for assembly sequences. *International Journal of Computational Geometry and Application*, 9(4/5), 371–417
28. Schweikard, A., and Schwarzer, F. (1998). Detecting geometric infeasibility *Artificial Intelligence*, 105(1-2), 139–159
29. Wilson, R. H., and Latombe, J. C. (1994). Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2), 371–396

30. Wilson, R. H., Kavraki, L., Lozano-Pérez, T., and Latombe, J. C. (1995). Two-handed assembly sequencing. *International Journal of Robotics Research*, 14(4), 335–350
31. LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press, Cambridge, UK
32. Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of robot motion: Theory, algorithms, and implementations*. Cambridge, MA: MIT Press
33. Goldwasser, M. H., and Motwani, R. (1999). Complexity measures for assembly sequences. *International Journal of Computational Geometry and Application*, 9(4/5), 371–417
34. Martelli, A., and Montanari, U. (1973). Additive and/or graphs. In: *Proceedings of the third international joint conference on artificial intelligence* (pp. 1–11)
35. Caselli, S., and Zanichelli, F. (1995). On assembly sequence planning using Petri Nets. In: *IEEE proceedings of the international symposium on assembly and task planning* (pp. 239– 244).
36. Cao, T., and Sanderson, A. C. (1998). And/or net representation for robotic task sequence planning. *IEEE Transaction on Systems, Man and Cybernetics, Part C Applications and Reviews*, 28(8), 104–218
37. Zha, X. F., Lim, S. Y. E., and Fok, S. C. (1998b). Integrated knowledge-based petri net intelligent flexible assembly planning. *Journal of Intelligent Manufacturing*, 9(3), 235–250
38. Ben-Arieh, D., Kumar, R. R., and Tiwari, M. K. (2004). Analysis of assembly operations difficulty using enhanced expert high-level colored fuzzy petri net model. *Robotics and Computer-Integrated Manufacturing*, 20(5), 385–403.
39. Motavalli, S., and Islam, A. U. (1997). Multi-criteria assembly sequencing. *Computer and Industrial Engineering*, 32(4), 743– 751

- 40.[23] Chen, C. L. P. (1992). Design of a real-time and/or assembly scheduler on an optimization neural network. *Journal of Intelligent Manufacturing*, 3(4), 251–261
- 41.Holland JH. *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press, 1975
- 42.Bonneville F, Perrard C, Henrioud JM. A genetic algorithm to generate and evaluate assembly plans. *Proceedings of the IEEE Symposium on Emerging Technology and Factory Automation*, 1995. p. 231–9
- 43.Sebaaly MF, Fujimoto H. A genetic planner for assembly automation. *Proceedings of the IEEE Conference on Evolutionary Computation*, 1996. p. 401–6
- 44.Chen S-F. Assembly planning—a genetic approach. *Proceedings of the 24th ASME Design Automation Conference*, 1998. Paper number DETC98/DAC-5798.
- 45.Hong DS, Cho HS. A genetic-algorithm-based approach to the generation of robotic assembly sequences. *Control Eng Practice* 1999;7(2):151–9.
46. Senin N, Groppetti R, Wallace DR. Concurrent assembly planning with genetic algorithms. *Robotics Comput Integrated Manuf* 2000;16(4):65–72.
- 47.Грувер, М. САПР и автоматизация производства / М. Грувер, Э. Зиммерс. - Москва: Наука, 1987. - 528 с.
- 48.Ушаков, Д. М. Введение в математические основы САПР / Д.М. Ушаков. - М.: ДМК Пресс, 2012. - 208 с.
- 49.Автоматизація проектування технологічних процесів в машинобудуванні / В[.С.Корсаков, Н.М.Капустін, К.Х.Темпельгоф, Х.Ліхтен-берг; За заг. ред. Н.М.Капустіна. - М.: Машинобудування, 1985. - 305 с.
- 50.Щербаков Н.П. Автоматизация технологического проектирования: Учеб. пособие / Алт. гос. техн. ун-т. – Барнаул.: Изд-во АлтГТУ, 2002. – 426 с.

- 51.Пасічник В.А., Кореньков В.М. Принципи формування математичної моделі складальної одиниці в виді бінарних відношень обмежень рухливості // Машиностроение и техносфера XXI века // Сборник трудов XV международной научно-технической конференции в г.Севастополе 15-20 сентября 2008г. В 4-х томах. – Донецк: ДонНТУ, 2008. Т.3. С. 64-70
- 52.Пасічник В.А., Семута Р.Р.: Виявлення значущих координатних напрямків і технологічних груп деталей у складанні // Вісник ЖІТІ. – 2002 / Спеціальний випуск / ІКТ 2002. – С. 152-157.
- 53.Пасічник В.А., Семута Р.Р. Автоматизоване формування математичної моделі складального виробу // Вісник Технологічного університету Поділля, Хмельницький. – 2003. – № 4, Ч.2 (53). – С. 236 – 242.
- 54.Кореньков В.М.: Автоматизований синтез маршрутних технологічних процесів складання. Дис.канд.техн.наук: 05.02.08. – К.: НТУУ «КПІ». – 2005. – 171 с.
- 55.Давигора В.М. Методика автоматизованого синтезу раціональних порядків складання / В.М. Давигора // Technologia i automatyzacja montazu. – Warszawa: OBR TEKOMA. – 2001. – № 2. – С. 3– 10.
- 56.Технологическая декомпозиция сборочных изделий [Текст] : дис... канд. техн. наук: 05.02.08 / Лапковский Сергей Викторович ; Национальный технический ун-т Украины "Киевский политехнический ин-т". - К., 1998. - 192 л. - л. 150-156
- 57.Філіппова М. В., Демченко М. О., Матвієнко С. М. Моделювання технологічного процесу складання за допомогою IDEF0 //Eastern-European Journal of Enterprise Technologies. – 2013. – Т. 2. – №. 3. – С. 62.
58. Проскурєнко Д.М. Застосування методології SADT у технологічній підготовці виробництва // XVI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Ефективність

- та автоматизація інженерних рішень у приладобудуванні», 08-09 грудня 2020 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2020. – С. 115-117.
- 59.Z. Xiaoming, D. Pingan, “A model-based approach to assembly sequence planning” *International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9-10, pp. 983–994, 2008.
 - 60.M. Wu, Y. Zhao, and C. Wang, “Knowledge-based approach to assembly sequence planning for wind-driven generator,” *Mathematical Problems in Engineering*, vol. 2013, 2013.
 - 61.Dini G and Santochi M. Automated sequencing and subassembly detection in assembly planning. *CIRP Ann Manuf Technol* 1992; 41: 1–4.
 - 62.Ou LM and Xu X. Relationship matrix based auto-matic assembly sequence generation from a CADmodel.*Comput Aided Des*2013; 45: 1053–1067.
 - 63.Філіппова М.В., Вислоух С.П. Методика автоматизованого проектування технології складання виробів приладобудування // *Процеси механічної обробки в машинобудуванні: Зб. наук. праць ЖДТУ – 2006. – Вип. 4. – с. 18-30.*
 - 64.Mok SM, Ong K and Wu CH. Automatic generation of assembly instructions using STEP. In: *IEEE international conference on robotics and automation. Proceedings 2001 ICRA, 2001, Vol. 1, pp. 313–318.* COEX. Seoul, South Korea: IEEE
 - 65.Pan C, Smith SS and Smith GC. Determining interference between parts in CAD STEP files for automatic assembly planning.*J Comput Inf Sci Eng*2005; 5: 56–62.
 - 66.Smith SS, Smith GC and Liao X. Automatic stable assembly sequence generation and evaluation.*J Manuf Syst*2001; 20: 225–235.
 - 67.Choi, Y. K., Lee, D. M., & Cho, Y. B. (2009). An approach to multi-criteria assembly sequence planning using genetic algorithms. *International Journal of Advanced Manufacturing Technology*, 42, 180–188.

- 68.Chen, S., & Liu, Y. J. (2001). An adaptive genetic assembly sequence planner. *International Journal of Computer Integrated Manufacturing*, 14(5), 489–500.
- 69.Bourjault A. Contribution a` une approche me`thodologique de l'assemblage automatise': e`laboration automatique des se`quences ope´ratoires. PhD Thesis, Universite de Franche-Comte, France, 1984.
- 70.De Fazio T and Whitney DE. Simplified generation of all mechanical assembly sequences. *IEEE J Robot Automat* 1987; 3: 640–658
- 71.Li Y, Gao J, Shi L, et al. Assembly modeling for 3d component based on polychromatic sets. *Adv Mater Res* 2012; 411: 388–392
- 72.Bahubalendruni MVAR and Biswal BB. Computer aid for automatic liaisons extraction from CAD based robotic assembly. In: *IEEE sponsored 8th international conference on intelligent systems and control*, Coimbatore, India, 2014, pp.42–45
- 73.Homem de Mello L and Sanderson A. Automatic generation of mechanical assembly sequences Report no. CMU-FU-TR-88-19, December 1988. Pittsburgh: Carnegie Mellon University.
- 74.Python [Электронный ресурс] - Режим доступа: <https://www.python.org> — 10.11.2021 г.
- 75.Python PyQt5[Электронный ресурс] - Режим доступа: <https://www.riverbankcomputing.com/static/Docs/PyQt5/> — 10.11.2021 г.
- 76.Python PythonOCC[Электронный ресурс] - Режим доступа: <https://github.com/tpaviot/pythonocc-core> — 10.11.2021 г.
- 77.Python Pyevolve[Электронный ресурс] - Режим доступа: <https://github.com/perone/Pyevolve> — 10.11.2021 г.
- 78.Python NetworkX[Электронный ресурс] - Режим доступа: <https://networkx.org/> — 10.11.2021 г.
- 79.Общемашиностроительные нормативы времени вспомогательного на обслуживание рабочего места и подготовительно-заключительного

для технического нормирования станочных работ. Серийное производство. Изд. 2-е. М., «Машиностроение», 1974, 421 с. (ЦБПНТ при НИИТруда).

- 80.Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.
- 81.Гавриш, О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів. Конспект лекцій : навчальний посібник для студентів спеціальностей 151 – «Автоматизація та комп'ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірвальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,88 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 188 с. URL: <http://ela.kpi.ua/handle/123456789/29447>
- 82.Гавриш, О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів: практикум: навчальний посібник для студентів спеціальностей 151 – «Автоматизація та комп'ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірвальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,11 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 116 с. URL: <http://ela.kpi.ua/handle/123456789/29450>

ДОДАТОК А

Код програми

```

main.py
from GUI import *
import sys
def start():
    app = QApplication(sys.argv)
    app.setStyleSheet(style)
    ex = main_menu()
    sys.exit(app.exec_())
if __name__ == '__main__':
    start()

GUI.py
# -*- coding: utf-8 -*-
import time
import re
import os
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.QtGui import *
import traceback
import openpyxl
from GA import *
from graph_import import create_graph
import random
style = '''
QMenu {
    background-color: white;
    margin: 2px; /* some spacing around the menu */
}
QMenu::item {
    padding: 2px 25px 2px 20px;
    border: 1px solid transparent; /* reserve space for selection border */
}
QMenu::item:selected {
    border-color: darkblue;
    background: rgba(100, 100, 100, 150);}
QMenu::icon:checked { /* appearance of a 'checked' icon */
    background: gray;
    border: 1px inset gray;
    position: absolute;
    top: 1px;
    right: 1px;
    bottom: 1px;
    left: 1px;
}
QMenu::separator {
    height: 2px;
    background: lightblue;
    margin-left: 10px;
    margin-right: 5px;
}
QMenu::indicator {
    width: 13px;
    height: 13px;
}
QProgressBar{
    border: 2px solid black;
    border-radius: 5px;
    text-align: center
}
QMenuBar {
    font: bold italic;
    color: black;
    background-color: rgba(255,255,255,0.7);
}
QMenuBar::item {
    spacing: 3px; /* spacing between menu bar items */
    padding: 1px 4px;
    background: transparent;
    border-radius: 4px;
}

QMenuBar::item:selected { /* when selected using mouse or keyboard */
    background: white;
}

QMenuBar::item:pressed {
    background: white;
}
QLabel{
    font: bold italic;
    color: black;
    background-color: rgba(255,255,255,0.8);
    border: 1px solid black;
    border-radius: 6px;
    padding: 2px;
}
QPlainTextEdit {
    border: 2px solid black;

```

```

        border-radius: 10px;
        padding: 0 8px;
        background: white;
        selection-background-color: darkgray;
    }
    QLineEdit {
        border: 2px solid black;
        border-radius: 10px;
        padding: 0 8px;
        background: white;
        selection-background-color: darkgray;
    }
    QPushButton {
        background-color: white;
        border-style: outset;
        border-width: 2px;
        border-radius: 10px;
        border-color: black;
        font: bold 10px;
        min-width: 5em;
        padding: 6px;
    }
    QPushButton:pressed {
        background-color: rgb(148, 148, 148);
        border-style: inset;
    }
    QComboBox {
        font: bold;
        border: 2px solid gray;
        border-radius: 8px;
        border-color: black;
        padding: 2px 18px 2px 3px;
        min-width: 6em;
    }
    QComboBox:on { /* shift the text when the popup opens */
        padding-top: 3px;
        padding-left: 4px;
    }
    QComboBox::drop-down {
        subcontrol-origin: padding;
        subcontrol-position: top right;
        width: 15px;
        border-left-width: 1px;
        border-left-color: darkgray;
        border-left-style: solid; /* just a single line */
        border-top-right-radius: 3px; /* same radius as the QComboBox */
        border-bottom-right-radius: 3px;
    }
    QComboBox::down-arrow {
        image: url(downarrow.png); width: 20px; height: 20px;
    }

    QComboBox::down-arrow:on { /* shift the arrow when popup is open */
        top: 1px;
        left: 1px;
    }
    QComboBox QAbstractItemView {
        border: 2px solid darkgray;
        selection-background-color: lightgray;
    }
}'''
def size_table(size):
    if size <=3:
        return 100
    elif 3<size <=5:
        return 150
    elif 5<size<=8:
        return 250
    elif 8<size<=10:
        return 350
    elif 10<size<=12:
        return 400
    else:
        return 500
class Thread(QThread):
    _signal = pyqtSignal(int)
    def __init__(self):
        super(Thread, self).__init__()
    def __del__(self):
        self.wait()
    def run(self):
        for i in range(100):
            time.sleep(0.01)
            self._signal.emit(i)
class main_menu(QMainWindow):
    def __init__(self):
        super().__init__()
        self.initUI()
    def initUI(self):

```

```

self.num_models = 0
exitAction = QAction(QIcon('exit.png'), '&Вийти', self)
exitAction.setShortcut('Ctrl+C')
exitAction.triggered.connect(qApp.quit)
Action = QAction(QIcon('help.png'), '&Допомога', self)
Action.setShortcut('Ctrl+H')
Action.triggered.connect(self.help)
Action1 = QAction(QIcon('info.png'), '&Інформація', self)
Action1.setShortcut('Ctrl+A')
Action1.triggered.connect(self.info)
self.statusBar()
menubar = self.menuBar()
fileMenu = menubar.addMenu('&File')
fileMenu1 = menubar.addMenu('&Info')
fileMenu.addAction(Action)
fileMenu.addAction(exitAction)
fileMenu1.addAction(Action1)
self.lb1 = QLabel(self)
self.lb1.setText('Оберіть 3D-модель виробу\нУВАГА! Розширення файлу *.STEP')
self.lb1.setFont(QFont('bold italic', 10))
self.lb1.adjustSize()
self.lb1.move(50,60)
self.openDirButton = QPushButton("Обрати файл", self)
self.openDirButton.clicked.connect(self.getFile)
self.openDirButton.resize(self.openDirButton.sizeHint())
self.openDirButton.move(50, 130)
self.plainTextEdit = QPlainTextEdit(self)
self.plainTextEdit.move(200, 130)
self.qbtn = QPushButton('Вийти', self)
self.qbtn.clicked.connect(QCoreApplication.instance().quit)
self.qbtn.resize(self.qbtn.sizeHint())
self.qbtn.move(400, 250)
self.btn = QPushButton('Далі', self)
self.btn.clicked.connect(self.on_click)
self.btn.resize(self.btn.sizeHint())
self.btn.move(30, 250)
self.btn.setEnabled(False)
self.setFixedSize(500, 300)
self.center()
self.progress = QProgressBar(self)
self.progress.setGeometry(50, 200, 400, 25)
#self.progress.setMaximum(100)
self.progress.setValue(0)
self.progress.hide()
self.setWindowIcon(QIcon('web1.png'))
oImage = QImage("image1.jpg")
sImage = oImage.scaled(QSize(500, 300))
palette = QPalette()
palette.setBrush(QPalette.Window, QBrush(sImage))
self.setPalette(palette)
self.setWindowTitle("AAS")
self.show()
def center(self):
    qr = self.frameGeometry()
    cp = QDesktopWidget().availableGeometry().center()
    qr.moveCenter(cp)
    self.move(qr.topLeft())
@pyqtSlot()
def getFile(self): # <-----
    try:
        self.dirfile = QFileDialog.getOpenFileName(self, "Оберіть файл", ".")[0]
        self.plainTextEdit.clear()
        self.plainTextEdit.appendHtml("<br>Ви обрали файл: <b>{}/<b>".format(self.dirfile))
        self.plainTextEdit.move(200, 110)
        self.plainTextEdit.resize(250, 65)
        if ".stp" in self.dirfile or ".STEP" in self.dirfile:
            self.lb1.setText('Для продовження роботи системи натисніть "Далі".')
            self.lb1.adjustSize()
            self.btn.setEnabled(True)
            self.progress.show()
            self.progress.setValue(0)
        except Exception:
            print(str(traceback.format_exc()))
@pyqtSlot()
def info(self):
    QMessageBox.question(self, 'Інформація про програму:',
        'Магістерська робота студента 6 курсу.\нВиконав: Проскурєнко Денис.', QMessageBox.Ok,
        QMessageBox.Ok)
@pyqtSlot()
def on_click(self):
    try:
        self.thread = Thread()
        self.thread._signal.connect(self.signal_accept)
        self.thread.start()
        self.btn.setEnabled(False)
    except Exception:
        print(str(traceback.format_exc()))
def signal_accept(self, msg):

```

```

self.progress.setValue(int(msg))
if self.progress.value() == 99:
    self.progress.setValue(0)
    self.progress.setEnabled(True)
if self.progress.value() == 90:
    try:
        stpfile = open(self.dirfile)
        self.name_models = []
        k2 = ""
        while True:
            data = stpfile.readline().rstrip()
            if data == "END-ISO-10303-21;":
                break
            k = re.search(r".*= PRODUCT \( '(.*?)', '.*', '.*', data)
            k1 = re.search(r"FILE_NAME \( '(.*?)\.(STEP|stp)',.*", data)
            if k1:
                k2 = str(k1.group(1))
                k1 = re.search(r"FILE_NAME \( '(.*?)\.(STEP|stp)',.*", data)
                k3 = str(k1.group(1))
            if k:
                if k.group(1) != k2:
                    self.name_models.append(k.group(1))
        matrix = matr(self.name_models)
        num = len(self.name_models)
        model = GA_algorithm(num, matrix)
        create_graph(model[0], num, model[1])
        model.append(self.name_models)
        model.append(k3)
    except Exception:
        print(str(traceback.format_exc()))
    exPopup = answer_menu(model, self)#505 300
    exPopup.setGeometry(350, 70, 750, 650)
    exPopup.setWindowTitle("Отримані результати:")
    exPopup.show()
def closeEvent(self, event):
    reply = QMessageBox.question(self, 'Ви впевнені?',
                                "Ви впевнені?", QMessageBox.Yes |
                                QMessageBox.No, QMessageBox.No)

    if reply == QMessageBox.Yes:
        event.accept()
    else:
        event.ignore()
class answer_menu(QMainWindow):
    def __init__(self, name, parent=None):
        super().__init__(parent)
        self.name = name
        self.table_widget = MyTableWidget(self.name, self)
        self.setCentralWidget(self.table_widget)
        act = QAction(QIcon('save.png'), '&Зберегти', self)
        act.setShortcut('Ctrl+S')
        act.triggered.connect(self.save)
        exitAction = QAction(QIcon('exit.png'), '&Вийти', self)
        exitAction.setShortcut('Ctrl+C')
        exitAction.triggered.connect(qApp.quit)
        Action = QAction(QIcon('help.png'), '&Допомога', self)
        Action.setShortcut('Ctrl+H')
        Action.triggered.connect(self.help)
        Action1 = QAction(QIcon('info.png'), '&Інформація', self)
        Action1.setShortcut('Ctrl+A')
        Action1.triggered.connect(self.info)
        self.statusBar()
        menubar = self.menuBar()
        fileMenu = menubar.addMenu('&File')
        fileMenu1 = menubar.addMenu('&Info')
        fileMenu.addAction(act)
        fileMenu.addAction(exitAction)
        fileMenu1.addAction(Action1)
        self.setWindowIcon(QIcon('web1.png'))
        oImage = QImage("image.jpg")
        sImage = oImage.scaled(QSize(505, 300))
        palette = QPalette()
        palette.setBrush(QPalette.Window, QBrush(sImage))
        self.setPalette(palette)
        self.show()
    @pyqtSlot()
    def save(self):
        with open('result.txt', "w", encoding="utf-8") as f:
            f.write(self.name)
        QMessageBox.question(self, 'Result',
                            'Дані було збережено як result.txt в папці з програмою.',
                            QMessageBox.Ok,
                            QMessageBox.Ok)
    @pyqtSlot()
    def help(self):
        QMessageBox.question(self, 'Допомога:',

```

```

        'Це меню з результатами виконання розрахунків, якщо ви десь знайшли помилку, повідомне
        автору(info).',
        QMessageBox.Ok,
        QMessageBox.Ok)

    @pyqtSlot()
    def info(self):
        QMessageBox.question(self, 'Інформація про програму:', 'РГР студента 5 курсу.\nВиконав: Проскурєнко Денис.',
        QMessageBox.Ok,
        QMessageBox.Ok)

class MyTableWidget(QWidget):
    def __init__(self, name, parent):
        super(QWidget, self).__init__(parent)
        self.name = name
        self.initUI()
    def initUI(self):
        self.layout = QVBoxLayout(self)
        # Initialize tab screen
        self.scroll = QScrollArea()
        self.scroll1 = QScrollArea()
        self.tabs = QTabWidget()
        self.tab1 = QWidget()
        self.tab2 = QWidget()
        # Add tabs
        self.tabs.addTab(self.tab1, "План складання")
        self.tabs.addTab(self.tab2, "Виробничий план складання")
        # План складання
        self.tab1.layout = QVBoxLayout(self)
        text = "Матриця зв'язку складальних одиниць у файлі 3Д-моделі - "+self.name[3]
        label0 = QLabel(text)
        self.table = QTableWidget(self)
        self.table.setColumnCount(len(self.name[1][0]))
        self.table.setRowCount(len(self.name[1]))
        nameC = []
        for i in range(len(self.name[1][0])):
            nameC.append(str(i + 1))
        self.table.setHorizontalHeaderLabels(nameC)
        for i in range(len(self.name[1])):
            for j in range(len(self.name[1][i])):
                it = QTableWidgetItem(str(self.name[1][i][j]))
                it.setTextAlignment(Qt.AlignCenter)
                self.table.setItem(i, j, it)
        self.table.resizeColumnsToContents()
        self.table.setMinimumHeight(size_table(self.table.rowCount()))
        self.scroll.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOn)
        self.scroll.setHorizontalScrollBarPolicy(Qt.ScrollBarAlwaysOff)
        self.scroll.setWidgetResizable(True)
        text = "Рекомендований план складання для 3Д-моделі - "+self.name[3]+":\n\n"
        text1 = ""
        for i in range(len(self.name[0])):
            if i < (len(self.name[0]) - 1):
                text += str(self.name[0][i]) + " -> "
                text1 += str(self.name[2][self.name[0][i]-1]) + " -> "
            else:
                text += str(self.name[0][i])
                text1 += str(self.name[2][self.name[0][i] - 1])
        text+="\n"+text1
        label = QPlainTextEdit(self)
        label.appendPlainText(text)
        #label = QLabel(text)
        pixmap = QPixmap("saved_figure.png")
        lbl = QLabel(self)
        lbl.setPixmap(pixmap)
        self.hbox = QHBoxLayout()
        self.hbox.addWidget(lbl)
        label1 = QLabel("Граф зв'язку\nЧорними лініями показані всі зв'язки між деталями, \nна червоних зображено
найкращий план складання.")
        self.widget = QWidget()
        self.vbox = QVBoxLayout()
        self.vbox.addWidget(label0)
        self.vbox.addWidget(self.table)
        self.vbox.addWidget(label)
        self.vbox.addWidget(label1)
        self.vbox.addLayout(self.hbox)
        self.widget.setLayout(self.vbox)
        self.scroll.setWidget(self.widget)
        self.tab1.layout.addWidget(self.scroll)
        self.tab1.setLayout(self.tab1.layout)
        #Виробничий план
        self.tab2.layout = QVBoxLayout(self)
        self.scroll1.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOn)
        self.scroll1.setHorizontalScrollBarPolicy(Qt.ScrollBarAlwaysOff)
        self.scroll1.setWidgetResizable(True)
        label1 = QLabel("За отриманою послідовністю складання виберіть необхідні операції")
        self.widget1 = QWidget()
        self.vbox1 = QVBoxLayout()
        self.vbox1.addWidget(label1)
        self.table1 = QTableWidget(self)
        self.table1.setColumnCount(8)

```

```

self.table1.setRowCount(len(self.name[0])-1)
nameC = ["Деталь1", "Деталь2", "Операція", "Перехід", "Інструмент", "Час на складання, хв", "Додати
рядок", "Видалити рядок"]
self.table1.setHorizontalHeaderLabels(nameC)
options = []
self.options = []
op = 15
for i in range(len(self.name[0])+10):
    options.append("0"+str(op)+' Складальна')
    self.options.append("0"+str(op)+' Складальна')
    op+=5
self.options1_1 = ["Виберіть перехід", "Встановити", "Загвинтити", "Закріпити", "Центрувати",
"Запесувати",
"Збалансувати", "Зпаяти", "Склеїти", "Зварити", "Зклепати", "Застібнути", "Стопорувати"]
self.options2_2 = ["Виберіть інструмент", "Викрутка", "Гайковий ключ", "Паяльник", "Тиски", "Зварювальний
апарат", "Без інструменту"]
self.options1 = []
self.options2 = []
self.names = []
for i in self.name[0]:
    self.names.append(self.name[2][i-1])
for i in range(len(self.name[0])-1):
    self.table1.setItem(i, 0, QTableWidgetItem(str(self.names[i])))
    self.table1.setItem(i, 1, QTableWidgetItem(str(self.names[i+1])))
    combo = QComboBox()
    combo.activated.connect(self.comboActivated)
    combo.addItem(options)
    j = self.options[i]
    combo.setCurrentText(j)
    combo1 = QComboBox()
    combo1.activated.connect(self.comboActivated1)
    combo1.addItem(self.options1_1)
    combo2 = QComboBox()
    combo2.activated.connect(self.comboActivated2)
    combo2.addItem(self.options2_2)
    self.options1.append(i)
    self.options2.append(i)
    self.layout.addWidget(combo)
    self.layout.addWidget(combo1)
    self.layout.addWidget(combo2)
    self.qbtn1_1 = QPushButton('+', self)
    self.qbtn1 = QPushButton('+', self)
    self.layout.addWidget(self.qbtn1_1)
    self.layout.addWidget(self.qbtn1)
    self.qbtns = QWidget(self)
    self.qbtns.setLayout(self.layout)
    self.qbtn2 = QPushButton('-', self)
    self.qbtn1.clicked.connect(self.plus)
    self.qbtn1_1.clicked.connect(self.plus1)
    self.qbtn2.clicked.connect(self.minus)
    self.table1.setCellWidget(i, 6, self.qbtns)
    #self.table1.setCellWidget(i, 6, EditButtonsWidget())
    self.table1.setCellWidget(i, 7, self.qbtn2)
    self.table1.setCellWidget(i, 2, combo)
    self.table1.setCellWidget(i, 3, combo1)
    self.table1.setCellWidget(i, 4, combo2)
self.table1.resizeColumnsToContents()
self.table1.setMinimumHeight(size_table(self.table1.rowCount()))
self.vbox1.addWidget(self.table1)
self.qbtn = QPushButton('Створити операційну карту', self)
self.qbtn.clicked.connect(self.create)
self.vbox1.addWidget(self.qbtn)
self.widget1.setLayout(self.vbox1)
self.scroll1.addWidget(self.widget1)
self.tab2.layout.addWidget(self.scroll1)
self.tab2.setLayout(self.tab2.layout)
# Add tabs to widget
self.layout.addWidget(self.tabs)
self.setLayout(self.layout)
def plus(self):
    button = self.sender()
    if button:
        row = self.table1.indexAt(button.parent().pos()).row()
        self.table1.insertRow(row+1)
        combo = QComboBox()
        combo.activated.connect(self.comboActivated)
        combo.addItem(self.options)
        combo1 = QComboBox()
        combo1.activated.connect(self.comboActivated1)
        combo1.addItem(self.options1_1)
        combo2 = QComboBox()
        combo2.activated.connect(self.comboActivated2)
        combo2.addItem(self.options2_2)
        try:
            j = self.options[row]
            combo.setCurrentText(j)
            new_list = []
            for k in range(len(self.options)):

```

```

        if k == int(row+1):
            new_list.append(j)
            new_list.append(self.options[k])
        self.options = new_list
        j = self.options1[row]
        new_list = []
        if int(row + 1) == len(self.options1):
            self.options1.append(j)
        else:
            for k in range(len(self.options1)):
                if k == int(row + 1):
                    new_list.append(j)
                    new_list.append(self.options1[k])
            self.options1 = new_list
        j = self.options2[row]
        new_list = []
        if int(row + 1) == len(self.options2):
            self.options2.append(j)
        else:
            for k in range(len(self.options2)):
                if k == int(row + 1):
                    new_list.append(j)
                    new_list.append(self.options2[k])
            self.options2 = new_list
        print(self.options)
        print(self.options1)
        print(self.options2)
    except Exception:
        print(str(traceback.format_exc()))
    self.lay = QHBoxLayout()
    self.lay.setContentsMargins(0, 0, 0, 0)
    self.lay.setSpacing(0)
    self.qbtn1_1 = QPushButton('+↑', self)
    self.qbtn1 = QPushButton('+↓', self)
    self.lay.addWidget(self.qbtn1_1)
    self.lay.addWidget(self.qbtn1)
    self.qbtns = QWidget(self)
    self.qbtns.setLayout(self.lay)
    self.qbtn2 = QPushButton('-', self)
    self.qbtn1.clicked.connect(self.plus)
    self.qbtn2.clicked.connect(self.minus)
    self.table1.setCellWidget(row+1, 6, self.qbtns)
    self.table1.setCellWidget(row+1, 7, self.qbtn2)
    self.table1.setCellWidget(row+1, 2, combo)
    self.table1.setCellWidget(row+1, 3, combo1)
    self.table1.setCellWidget(row+1, 4, combo2)

def plus1(self):
    button = self.sender()
    if button:
        row = self.table1.indexAt(button.parent().pos()).row()
        self.table1.insertRow(row)
        combo = QComboBox()
        combo.activated.connect(self.comboActivated)
        combo.addItems(self.options)
        combo1 = QComboBox()
        combo1.activated.connect(self.comboActivated1)
        combo1.addItems(self.options1_1)
        combo2 = QComboBox()
        combo2.activated.connect(self.comboActivated2)
        combo2.addItems(self.options2_2)
    try:
        j = self.options[row]
        combo.setCurrentText(j)
        self.options.insert(row, j)
        j = self.options1[row]
        self.options1.insert(row, j)
        j = self.options2[row]
        self.options2.insert(row, j)
    except Exception:
        print(str(traceback.format_exc()))
    self.lay = QHBoxLayout()
    self.lay.setContentsMargins(0, 0, 0, 0)
    self.lay.setSpacing(0)
    self.qbtn1_1 = QPushButton('+↑', self)
    self.qbtn1 = QPushButton('+↓', self)
    self.lay.addWidget(self.qbtn1_1)
    self.lay.addWidget(self.qbtn1)
    self.qbtns = QWidget(self)
    self.qbtns.setLayout(self.lay)
    self.qbtn2 = QPushButton('-', self)
    self.qbtn1.clicked.connect(self.plus)
    self.qbtn2.clicked.connect(self.minus)
    self.table1.setCellWidget(row, 6, self.qbtns)
    self.table1.setCellWidget(row, 7, self.qbtn2)
    self.table1.setCellWidget(row, 2, combo)
    self.table1.setCellWidget(row, 3, combo1)
    self.table1.setCellWidget(row, 4, combo2)

def minus(self):

```

```

button = self.sender()
if button:
    row = self.table1.indexAt(button.pos()).row()
    del self.options[row]
    del self.options1[row]
    del self.options2[row]
    self.table1.removeRow(row)
def comboActivated(self, option):
    combo = self.sender()
    #print('опція: %s, "%s"' % (option, combo.itemText(option)))
    index = self.table1.indexAt(combo.pos())
    self.options[index.row()] = combo.itemText(option)
    #print('рядок: %s, колонка: %s' % (index.row(), index.column()))
def comboActivated1(self, option):
    combo = self.sender()
    #print('опція: %s, "%s"' % (option, combo.itemText(option)))
    index = self.table1.indexAt(combo.pos())
    self.options1[index.row()] = combo.itemText(option)
    #print('рядок: %s, колонка: %s' % (index.row(), index.column()))
time = {"Встановити": "0.6", "Загвинтити": "1.1", "Закріпити": "1.5", "Центрувати": "2", "Запресувати":
        "0.88",
        "Збалансувати": "6.7", "Зпаяти": "4.2", "Склеїти": "5", "Зварити": "6.2", "Застібнути": "1",
        "Стопорувати": "3.5", "Зклепати": "1.5"}
time = round(random.uniform(float(time[combo.itemText(option)])-0.5, float(time[combo.itemText(option)]+1)),2)
it = QTableWidgetItem(str(time))
it.setTextAlignment(Qt.AlignCenter)
self.table1.setItem(index.row(), index.column() + 2, it)
def comboActivated2(self, option):
    combo = self.sender()
    #print('опція: %s, "%s"' % (option, combo.itemText(option)))
    index = self.table1.indexAt(combo.pos())
    self.options2[index.row()] = combo.itemText(option)
    #print('рядок: %s, колонка: %s' % (index.row(), index.column()))
def create(self):
    try:
        switch = 1
        for i in self.options1:
            if type(i) == int:
                switch = 0
                break
        for i in self.options2:
            if type(i) == int:
                switch = 0
                break
        if switch == 1:
            column = self.table1.columnCount()
            row = self.table1.rowCount()
            table = []
            for i in range(row):
                table11 = []
                for j in range(column):
                    if j not in [2,3,4,6,7]:
                        table11.append(self.table1.item(i, j).text())
                    elif j == 2:
                        table11.append(self.options[i])
                    elif j == 3:
                        table11.append(self.options1[i])
                    elif j == 4:
                        if self.options2[i] == "Без інструменту":
                            table11.append("_")
                        else:
                            table11.append(self.options2[i])
                table.append(table11)
            print(table)
            m = os.getcwd()
            file = m+'\\OK1.xlsx'
            file1 = m+'\\OK.xlsm'
            srcfile = openpyxl.load_workbook(file, read_only=False, keep_vba=True)
            num = 20
            sheetname = srcfile.get_sheet_by_name('OK2') # get sheetname from the file
            oper = table[0][2]
            print(oper)
            num1 = 3
            num2 = 5
            for i in table:
                print(i[2])
                if oper == i[2]:
                    print(1)
                    sheetname["P12"] = i[2]
                    x = "P" + str(num)
                    sheetname[x] = i[0]+" " + i[3]+" " + i[1]
                    y = "KI" + str(num)
                    sheetname[y] = i[5]
                    num += 2
                    x = "P" + str(num)
                    sheetname[x] = i[4]
                    num += 2

```

```

        if num > 42:
            sheetname = srcfile.get_sheet_by_name("OK" + str(num1))
            num1 += 1
            num = 2
            sheetname["LG45"] = num2
            num2 += 1
        else:
            sheetname = srcfile.get_sheet_by_name("OK" + str(num1))
            num1 += 1
            oper = i[2]
            sheetname["P12"] = i[2]
            num = 20
            x = "P" + str(num)
            sheetname[x] = i[0] + " " + i[3] + " " + i[1]
            y = "KI" + str(num)
            sheetname[y] = i[5]
            num += 2
            x = "P" + str(num)
            sheetname[x] = i[4]
            num += 2
            sheetname["LG45"] = num2
            num2 += 1
        sheet = srcfile.sheetnames
        for i in range(num1, 21):
            pfd = srcfile[sheet[i]]
            srcfile.remove(pfd)
            k = re.search(r"0(.*) Складальна", table[-1][2])
            k = int(k.group(1))
            sheetname = srcfile.get_sheet_by_name("Випробування")
            sheetname["P12"] = "0" + str(k + 5) + " Випробування"
            sheetname["LG45"] = num2
            num2 += 1
            sheetname = srcfile.get_sheet_by_name("Контроль")
            sheetname["P12"] = "0" + str(k + 10) + " Контроль"
            sheetname["LG45"] = num2
            num2 += 1
            srcfile.save(file1)
            srcfile.close()
            QMessageBox.about(self, "Операція виконана", "Для перегляду створеної операційної карти відкрийте
файл OK.xlsm")
        else:
            QMessageBox.about(self, "Помилка", "Будь-ласка оберіть всі операції та інструменти!!!")
    except Exception:
        print(str(traceback.format_exc()))

GA.py
import csv
from random import shuffle
from pyevolve import G1DList, GSimpleGA, Consts, Initializators, Selectors, Mutators, Crossovers
precedence_list = []
orientations = []
tools_grippers = []
def check_precedence_criteria(sequence):
    for i, part in enumerate(sequence):
        to_search = precedence_list[part]
        done_sequence = sequence[0:i]
        to_search_final = [x for j, x in enumerate(to_search) if j not in done_sequence]
        for a in to_search_final:
            if a == '1':
                return False
    return True
def check_precedence_swaps(sequence):
    score = 0
    for i, part in enumerate(sequence):
        to_search = precedence_list[part]
        done_sequence = sequence[0:i]
        to_search_final = [x for j, x in enumerate(to_search) if j not in done_sequence]
        for a in to_search_final:
            if a == '1':
                score += 1
    return score
def fitness_func(chromosome):
    chromosome = list(chromosome)
    if check_precedence_criteria(chromosome):
        return 0.0005 * (calc_tool_changes(chromosome) + calc_orientation_changes(chromosome))
    else:
        return check_precedence_swaps(chromosome) * 0.01
def calc_orientation_changes(sequence):
    orien_changes = 0
    current_orientation = orientations[sequence[0]]
    for part in sequence[1:]:
        if orientations[part] != current_orientation:
            orien_changes += 1
            current_orientation = orientations[part]
    return orien_changes
def calc_tool_changes(sequence):
    tool_changes = 0
    current_tool = tools_grippers[sequence[0]]
    for part in sequence[1:]:

```

```

        if tools_grippers[part] != current_tool:
            tool_changes += 1
            current_tool = tools_grippers[part]
    return tool_changes
def init_pop(genome, **args):
    shuffle(genome.genomeList)
def GA_algorithm(NUM_PARTS, matrix):
    for i in matrix:
        precedence_list.append(i)
    genome = G1DList.G1DList(NUM_PARTS)
    genome.setParams(rangemin=0, rangemax=NUM_PARTS - 1)
    genome.genomeList = list(range(0, NUM_PARTS))
    genome.initializator.set(init_pop)
    for i in range(NUM_PARTS):
        orientations.append(ors[i])
    for i in range(NUM_PARTS):
        tools_grippers.append(tos[i])
    genome.mutator.set(Mutators.G1DListMutatorSwap)
    genome.crossover.set(Crossovers.G1DListCrossoverCutCrossfill)
    genome.evaluator.set(fitness_func)
    ga = GSimpleGA.GSimpleGA(genome)
    ga.setGenerations(220)
    ga.setPopulationSize(80)
    ga.setMutationRate(0.01)
    ga.setCrossoverRate(0.985)
    # Set Selection scheme
    # ga.selector.set(Selectors.GRouletteWheel)
    ga.selector.set(Selectors.GTournamentSelector)
    # Set type of objective/ fitness function: Convergence
    ga.setMinimax(Consts.minimaxType["minimize"])
    ga.evolve(freq_stats=50)
    if check_precedence_criteria(list(ga.bestIndividual())) == True:
        sequence = [m + 1 for m in list(ga.bestIndividual())]
        #print("Final Sequence: ", sequence)
        #print("Tool Changes: ", calc_tool_changes(list(ga.bestIndividual())))
        #print("Orientation Changes: ", calc_orientation_changes(list(ga.bestIndividual())))
        for i in range(len(precedence_list)):
            for j in range(len(precedence_list[i])):
                precedence_list[i][j] = int(precedence_list[i][j])
        return [sequence, precedence_list]
    else:
        return [0]
stepanalyzer.py
#!/usr/bin/env python
from OCC.Core.IFSelect import IFSelect_RetDone
from OCC.Core.STEPControl import STEPControl_Reader
from OCC.Core.TDF import TDF_Label, TDF_LabelSequence
from OCC.Core.TCollection import TCollection_ExtendedString
from OCC.Core.TDocStd import TDocStd_Document
from OCC.Core.XCAFApp import XCAFApp_Application_GetApplication
from OCC.Core.XCAFDoc import XCAFDoc_DocumentTool_ShapeTool
class StepAnalyzer():
    def __init__(self, document=None, filename=None):
        """Supply one or the other: document or STEP filename."""
        self.parts = []
        self.parts_name = []
        self.uid = 1
        self.indent = 0
        self.output = ""
        self.fname = filename
        if filename:
            self.doc = self.read_file(filename)
        elif document:
            self.doc = document
            self.shape_tool = XCAFDoc_DocumentTool_ShapeTool(self.doc.Main())
        else:
            print("Supply one or the other: document or STEP filename.")
    def read_file(self, fname):
        doc = TDocStd_Document(TCollection_ExtendedString("STEP"))
        app = XCAFApp_Application_GetApplication()
        app.NewDocument(TCollection_ExtendedString("MDTV-XCAF"), doc)
        self.shape_tool = XCAFDoc_DocumentTool_ShapeTool(doc.Main())
        self.shape_tool.SetAutoNaming(True)
        step_reader = STEPControl_Reader()
        step_reader.SetColorMode(True)
        step_reader.SetLayerMode(True)
        step_reader.SetNameMode(True)
        step_reader.SetMatMode(True)
        status = step_reader.ReadFile(fname)
        if status == IFSelect_RetDone:
            step_reader.Transfer(doc)
        return doc
    def dump(self):
        if self.fname:
            self.output += f"Assembly structure of file: {self.fname}\n\n"
        else:
            self.output += "Assembly structure of doc:\n\n"
        self.indent = 0

```

```

# Find root label of step doc
labels = TDF_LabelSequence()
self.shape_tool.GetShapes(labels)
nbr = labels.Length()
rootlabel = labels.Value(1)
name = rootlabel.GetLabelName()
entry = rootlabel.EntryDumpToString()
is_assy = self.shape_tool.IsAssembly(rootlabel)
if is_assy:
    self.output += f"{self.uid}\t[{entry}] название модели {name}\t"
    self.uid += 1
    self.indent += 2
    top_comps = TDF_LabelSequence()
    subchilds = False
    is_assy = self.shape_tool.GetComponents(rootlabel, top_comps,
                                           subchilds)
    self.output += f"Number of labels at root = {nbr}\n"
    if top_comps.Length():
        self.find_components(top_comps)
return self.output, self.parts, self.parts_name
def find_components(self, comps):
    for j in range(comps.Length()):
        c_label = comps.Value(j+1)
        c_name = c_label.GetLabelName()
        c_entry = c_label.EntryDumpToString()
        ref_label = TDF_Label()
        is_ref = self.shape_tool.GetReferredShape(c_label, ref_label)
        if is_ref:
            ref_entry = ref_label.EntryDumpToString()
            ref_name = ref_label.GetLabelName()
            ref_label.
            indent = "\t" * self.indent
            self.output += f"{self.uid}{indent} [{c_entry}] {c_name}"
            self.output += f" => [{ref_entry}] то что нужно {ref_name}\n"
            self.uid += 1
            if self.indent >= 3:
                self.parts[-1].append(ref_name)
            else:
                self.parts_name.append(ref_name)
                self.parts.append([ref_name])
            if self.shape_tool.IsAssembly(ref_label):
                self.indent += 1
                ref_comps = TDF_LabelSequence()
                subchilds = False
                _ = self.shape_tool.GetComponents(ref_label, ref_comps,
                                                  subchilds)
                if ref_comps.Length():
                    self.find_components(ref_comps)
        self.indent -= 1
graph_.py
import csv
import networkx as nx
import matplotlib.pyplot as plt
def create_graph(mas, NUM_PARTS, matrix):
    plt.figure(figsize=(6, 6))
    plan_edges = []
    for i in range(len(mas)):
        if i != NUM_PARTS-1:
            edges = tuple([str(mas[i]), str(mas[i+1])])
            plan_edges.append(edges)
            edges = ()
    precedence_list = matrix
    for i in range(len(precedence_list)):
        for j in range(len(precedence_list[i])):
            precedence_list[i][j] = int(precedence_list[i][j])
    G = nx.DiGraph()
    for i in range(NUM_PARTS):
        G.add_node(str(i+1))
    for i in range(len(precedence_list)):
        for j in range(len(precedence_list[i])):
            if precedence_list[i][j] == 1:
                G.add_edge(str(i+1), str(j+1), weight=30)
    pos = nx.circular_layout(G)
    black_edges = [edge for edge in G.edges() if edge not in plan_edges]
    nx.draw_networkx_nodes(G, pos, node_color='b', node_size = 800)
    nx.draw_networkx_labels(G, pos, font_color='w', font_size=18)
    nx.draw_networkx_edges(G, pos, edgelist=plan_edges, edge_color='r', arrows=True, node_size =
                        800, width=3.5, arrowsize=20)
    nx.draw_networkx_edges(G, pos, edgelist=black_edges, arrows=False, node_size = 800, width=1.5)
    plt.savefig('saved_figure.png')

```

ДОДАТОК Б

Результат роботи програми

Матриця зв'язку складальних одиниць у файлі 3Д-моделі -

Редуктор.STEP:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0
5	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	1
11	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
14	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0

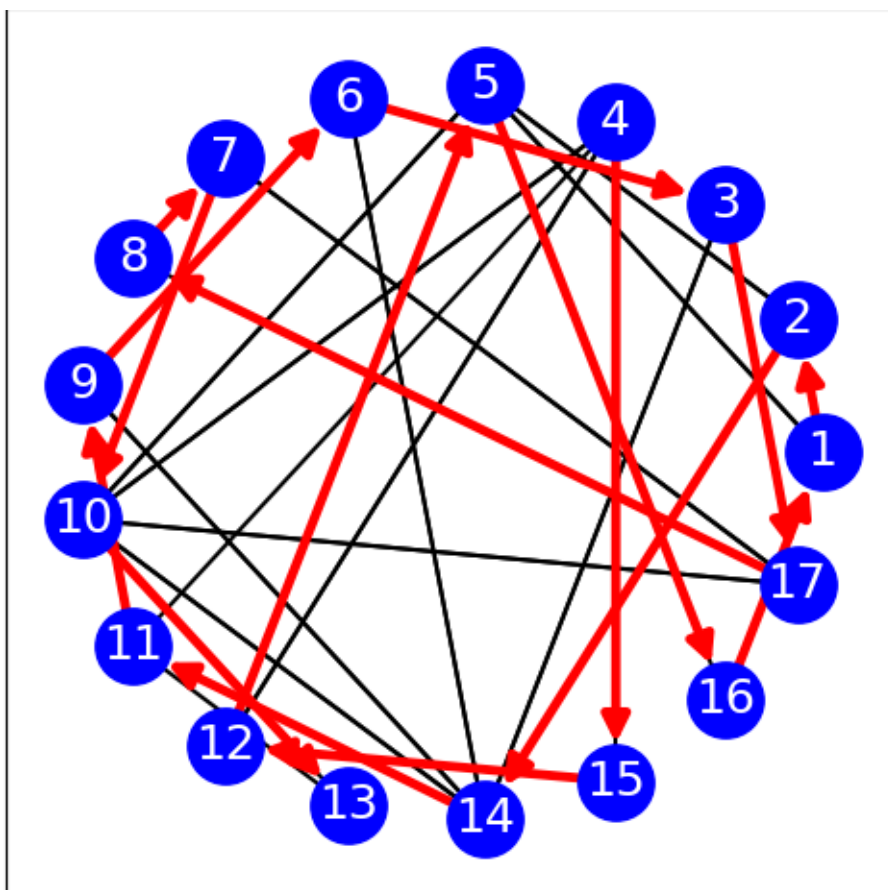
Рекомендований план складання для 3Д-моделі - Редуктор.STEP:

4 -> 15 -> 12 -> 10 -> 5 -> 16 -> 1 -> 2 -> 10 -> 14 -> 11 -> 9 -> 6 -> 3 -> 10 ->
17 -> 8 -> 7 -> 10 -> 10 -> 13

Вал s4 -> Підшипник B6012 -> Зубчате колесо Z80 -> Корпус -> Вал s3 ->
Підшипник B6010 -> Зубчате колесо Z40 -> Зубчате колесо Z90 -> Корпус -
> Вал s2 -> Підшипник B6006 -> Зубчате колесо Z30 -> Зубчате колесо Z96
-> Підшипник B6008 -> Корпус -> Вал s1 -> Підшипник B6004 -> Зубчате
колесо Z24 -> Корпус -> Корпус -> Кришка

Граф зв'язку

Чорними лініями показані всі зв'язки між деталями, а червоними зображено найкращий план складання.



Операційна карта

[illegible]

[illegible]

Дубл.																			
Взам.																			
Подл.																			
Разраб.																			
Проверил																			
Утвердил																			
Н.контр.																			
Код, наименование операции		Обозначение документа										МИ							
A 01	030 Складальна																		
Код, наименование оборудования		Тв										То							
B 02	Стіл для складання																		
К/М	Наименование детали, сб. единицы или материала										Код, обозначение		ОПП	ЕВ	ЕН	КИ	Н.расх.		
Р	ПС	V	t c., min	T-PA, oC	T-PA Ж, oC	ПЛ.	Q Г	НМ	Т Н, С		Т ОХ, С								
О	03 Підшипник B6004 запресувати на Вал s1										1.22								
Т	04																		
05	Зубчатє колесо Z24 встановити на Вал s1										0.73								
06																			
07	Вал s1 встановити на Корпус										0.1								
08																			
09																			
10																			
11																			
12																			
13																			
14																			
15																			
OK												7							

Дубл.																			
Взам.																			
Подл.																			
Разраб.																			
Проверил																			
Утвердил																			
Н.контр.																			
Код, наименование операции		Обозначение документа										МИ							
A 01	035 Складальна																		
Код, наименование оборудования		Тв										То							
B 02	Стіл для складання																		
К/М	Наименование детали, сб. единицы или материала										Код, обозначение		ОПП	ЕВ	ЕН	КИ	Н.расх.		
Р	ПС	V	t c., min	T-PA, oC	T-PA Ж, oC	ПЛ.	Q Г	НМ	Т Н, С		Т ОХ, С								
О	03 Кришка встановити на Корпус										1.31								
Т	04																		
05																			
06																			
07																			
08																			
09																			
10																			
11																			
12																			
13																			
14																			
15																			
OK												8							

[illegible]

ДОДАТОК В

Список наукових праць
Проскурєнко Дєниса Миколайовича

№ з/п	Назва праці	Назва видання та його вихідні відомості, що дозволяють ідентифікувати та відрізнити це видання від усіх інших	Кількіс ть друкова них аркушів	Прізвища співавторів
1	2	3	4	5
1	Застосування методології SADT у технологічній підготовці виробництва	XVI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні», 08-09 грудня 2020 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2020. – С. 115-117.	3	-
2	Моделювання інформаційних процесів забезпечення технологічності виробів на ранніх етапах проектування	XIV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «ПОГЛЯД У МАЙБУТНЄ ПРИЛАДОБУДУВАННЯ», 18-19 травня 2021р. - К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2021. – С. 162-166.	5	-
3	Ідентифікація вузлів виробу шляхом використання проектної інформації в 3д-моделях	XVII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні», 07-08 грудня 2021 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2021	4	-
4	Автоматизоване планування послідовності складання	Науковий журнал «Енергетика та автоматика», №5 – Київ: НУБіП України, – 2021		Філаппова М.В Демченко М.О. Третяк О.В.

Автор

В.О. завідувача кафедри

Проскурєнко Д.М.

Антонюк В.С.



Григорій ТИМЧИК

« 11 » 12 2021р.

АКТ

про впровадження результатів дисертаційного дослідження в освітній процес кафедри виробництва приладів

магістранта Проскурєнка Дениса Миколайовича

на тему:

«Проектування процесів виробництва на основі методології SADT»

Комісії у складі:

Голова – Заступник декана приладобудівного факультету з навчально-методичної роботи, к.т.н. доцент Філіппова М.В.:

Члени комісії: - в.о. завідувача кафедри виробництва приладів, д.т.н., професор Антонюк В.С.
- доцент кафедри виробництва приладів, к.т.н., доцент Подолян О.О.

цим актом засвідчує те, що результати дисертаційного дослідження Проскурєнка Дениса Миколайовича на тему: «Проектування процесів виробництва на основі методології SADT» - а саме, виконані дослідження у ході виконання даної дисертації, що використовується при проведенні лекцій а лабораторних практикумів у навчальному процесі викладачами кафедри виробництва приладів приладобудівного факультету КПІ ім. Ігоря Сікорського.

Голова комісії М.В. Філіппова

Члени комісії В.С. Антонюк

О.О. Подолян

« ____ » _____ 2021 р.

ДОДАТОК Г

Презентація

Проектування процесів виробництва на основі методології SADT

Виконав:

Проскурєнко Д.М. студент II курсу ПБ-01мп

Науковий керівник – доцент, к.т.н. Філіппова
М.В.

Мета та задачі дослідження

Метою роботи є розробка принципів та методів проєктування процесів виробництва, спрямована на удосконалення технологічної підготовки виробництва шляхом створення базису для наскрізної автоматизації виробництва.

Для досягнення мети були поставлені наступні задачі:

- Створення функціональних моделей об'єктів та процесів предметної області проєктування виробничих процесів виготовлення виробів;
- Розроблення моделі представлення формалізованих знань про предметну область і методики формування цієї моделі;
- Розроблення методів і алгоритмів прийняття рішень у задачах проєктування маршрутів складання виробів;
- Створення концепції автоматизованого проєктування виробничих процесів, що базується на застосуванні апарату обробки формалізованих знань.

2

Об'єкт та предмет дослідження

- **Об'єкт дослідження** - виробничий процес виготовлення виробів,
- **Предмет дослідження** – функціональні моделі виробничих процесів складання.

3

Наукова новизна та практична цінність

Наукова новизна, заключається у адаптації методології SADT для роботи розробленого програмного забезпечення, яке створене для побудови плану складання та проектування технологічних процесів складання.

Практична цінність:

- Розроблено генетичний алгоритм для вирішення задачі автоматизації послідовності складання;
- розроблено програмне забезпечення з інтерфейсом користувача для швидкого аналізу 3Д-модулі виробу, автоматизованого створення послідовності складання та генерації виробничого плану.

4

Актуальність

Складання є дуже важливим виробничим процесом. На складання зазвичай припадає значний відсоток загального часу та вартості виробництва. У традиційному промисловому виробництві процеси складання займають близько 50% загального часу виробництва і становлять понад 20% загальної вартості виробництва. Таким чином, щоб бути конкурентоспроможними в сучасному ринковому середовищі, виробники повинні постійно прагнути до скорочення часу та витрат на складання. Автоматизація процесу планування складання допоможе проектувальникам приймати рішення, які призведуть до кращого планування складання і, отже, скорочення часу виробництва та витрат. Перевагами автоматизованого планування складання є:

- Можна створити декілька можливих послідовностей складання, і інженер може вибрати потрібну послідовність на основі таких критеріїв, як наявність машини, вартість утримання тощо.
- Виробничі обмеження можуть бути включені швидше в процес проектування, що призведе до скорочення часу проектування.
- При серійному виробництві час переналадки набагато менше. Автоматичне планування складання допоможе швидшій заміні людей, машин, інструментів тощо.

5

Математична модель складального виробу

Запропоновано математичну модель подати як поєднання функцій :

$$R \subset A_1 \times A_2 \times \dots \quad (1)$$

– $a_i \in A_1$, $a_j \in A_2$ і так далі – номери деталей виробу;

$$F_G: R \times T \rightarrow \Theta \quad (2)$$

– деталь a_i обмежує можливе переміщення a_j по траєкторії $t \in T$ на відстані $\Theta \in \Theta$ (тут під t слід розуміти довільну криву, як найпростіший і найчастіший випадок – поступальне переміщення вздовж координатного напрямку або поворот навколо осі);

$$F_P: R \times H \rightarrow \Omega \quad (3)$$

– з'єднання a_i та a_j належить певному типу сполук H з відомою характеристикою $\omega \in \Omega$;

$$F_E: R \times O \rightarrow Z \quad (4)$$

– з'єднання a_i і a_j може бути отримане на устаткуванні $o \in O$ з фізичним(технологічним) принципом функціонування $\zeta \in Z$.

6

Математична модель процесу проектування технології

Процес синтезу технологічного процесу складання (ТПС) може бути уніфікований і зведений до кількох стандартних підпроцесів подальшого опрацювання прийнятих рішень. Інакше висловлюючись, задається процес Π , який відображає проектування маршрутного ТПС, а його складові можна представити як реалізацію послідовності підпроцесів Π_i , в кожному з яких автоматизовано вирішується ряд завдань:

$$\Pi \rightarrow \bigcup_{i=1}^k \Pi_i \rightarrow \begin{cases} \bigcup_{j=1}^f P_{T_j}^1(Q, M_j) \rightarrow G \\ \bigcup_{t=1}^n P_{T_j}^2(G, D, W_t) \rightarrow R \end{cases} \quad (5)$$

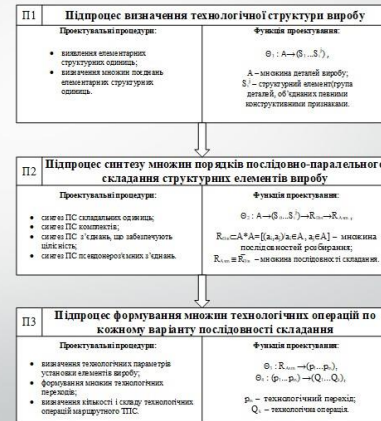


Рис.1. Структура процесу синтезу маршрутного ТПС виробів.

7

Функціональна модель автоматизованого проектування технологічного процесу складання



Рис.2. Контекстна діаграма.

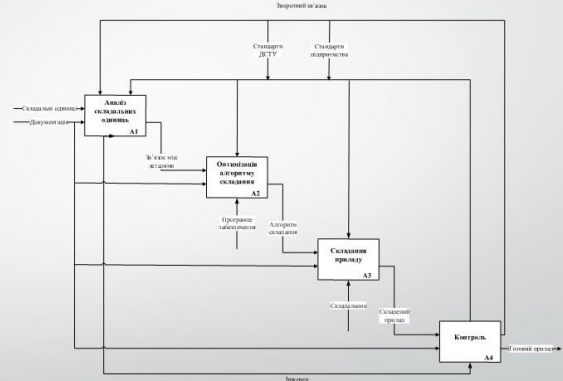


Рис.3. Діаграма декомпозиції.

8

ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ВИРОБНИЧИХ ПРОЦЕСІВ СКЛАДАННЯ

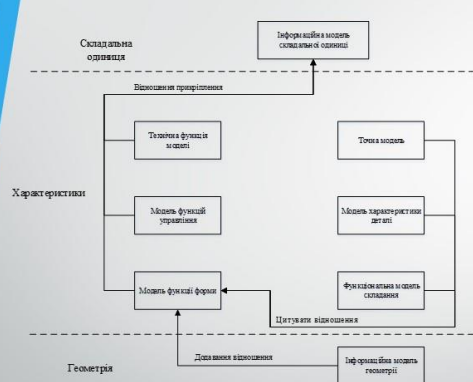


Рис.4. Інформаційна модель складальної одиниці.

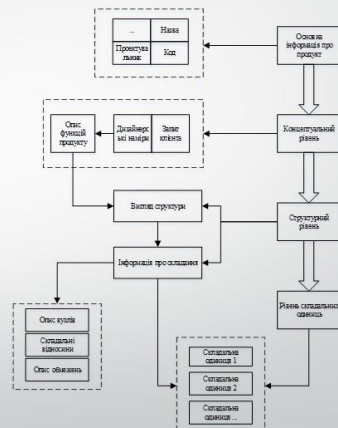
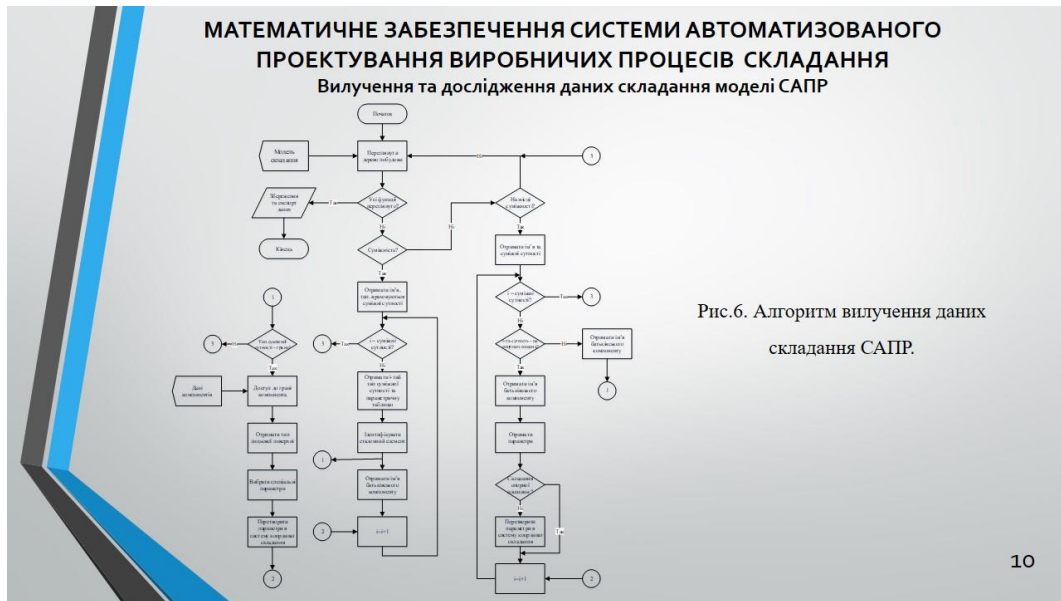


Рис.5. Структура складальної моделі.

9



ДОСЛІДНА ЕКСПЛУАТАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ СКЛАДАННЯ

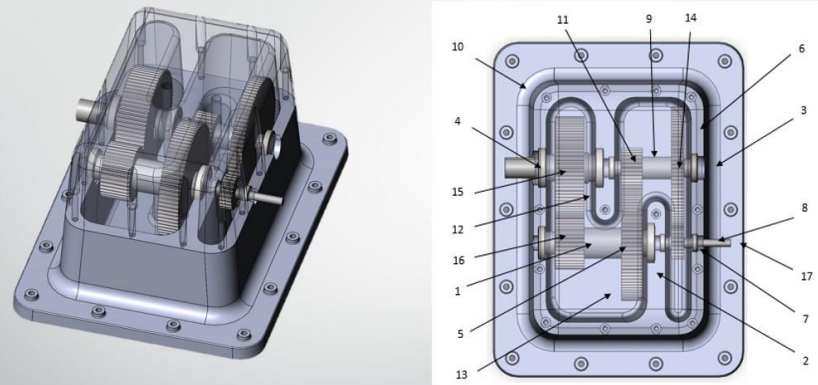


Рис.10. 3Д-модель редуктора з нумерацією деталей:

де, 1 – Зубчате колесо Z40; 2 – Зубчате колесо Z90; 3 – Підшипник B6008; 4 – Вал s4; 5 – Вал s3; 6 – Зубчате колесо Z96; 7 – Зубчате колесо Z24; 8 – Підшипник B6004; 9 – Зубчате колесо Z30; 10 – Корпус; 11 – Підшипник B6006; 12 – Зубчате колесо Z80; 13 – Кришка; 14 – Вал s2; 15 – Підшипник B6012; 16 – Підшипник B6010; 17 – Вал s1.

13

ДОСЛІДНА ЕКСПЛУАТАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ СКЛАДАННЯ



а)



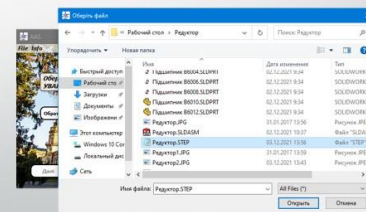
б)

Рис.11. Перше робоче вікно інтерфейсу програми:

а) перше вікно;

б) вибір файлу;

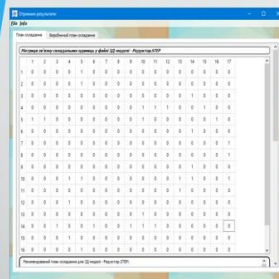
в) вікно із правильно вибраним файлом.



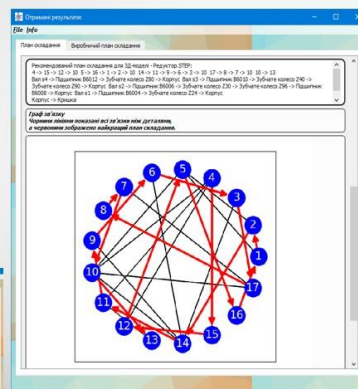
в)

14

ДОСЛІДНА ЕКСПЛУАТАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ СКЛАДАННЯ



а)



б)

№	Назва	Вид	Матеріал	Кількість	Вартість	Всього
1	Підшипник B6004	Вал	Сталь	1	1.00	1.00
2	Підшипник B6004	Вал	Сталь	1	1.00	1.00
3	Підшипник B6004	Вал	Сталь	1	1.00	1.00
4	Підшипник B6004	Вал	Сталь	1	1.00	1.00
5	Підшипник B6004	Вал	Сталь	1	1.00	1.00
6	Підшипник B6004	Вал	Сталь	1	1.00	1.00
7	Підшипник B6004	Вал	Сталь	1	1.00	1.00
8	Підшипник B6004	Вал	Сталь	1	1.00	1.00
9	Підшипник B6004	Вал	Сталь	1	1.00	1.00
10	Підшипник B6004	Вал	Сталь	1	1.00	1.00
11	Підшипник B6004	Вал	Сталь	1	1.00	1.00
12	Підшипник B6004	Вал	Сталь	1	1.00	1.00
13	Підшипник B6004	Вал	Сталь	1	1.00	1.00
14	Підшипник B6004	Вал	Сталь	1	1.00	1.00
15	Підшипник B6004	Вал	Сталь	1	1.00	1.00
16	Підшипник B6004	Вал	Сталь	1	1.00	1.00
17	Підшипник B6004	Вал	Сталь	1	1.00	1.00
18	Підшипник B6004	Вал	Сталь	1	1.00	1.00
19	Підшипник B6004	Вал	Сталь	1	1.00	1.00
20	Підшипник B6004	Вал	Сталь	1	1.00	1.00
21	Підшипник B6004	Вал	Сталь	1	1.00	1.00
22	Підшипник B6004	Вал	Сталь	1	1.00	1.00
23	Підшипник B6004	Вал	Сталь	1	1.00	1.00
24	Підшипник B6004	Вал	Сталь	1	1.00	1.00
25	Підшипник B6004	Вал	Сталь	1	1.00	1.00
26	Підшипник B6004	Вал	Сталь	1	1.00	1.00
27	Підшипник B6004	Вал	Сталь	1	1.00	1.00
28	Підшипник B6004	Вал	Сталь	1	1.00	1.00
29	Підшипник B6004	Вал	Сталь	1	1.00	1.00
30	Підшипник B6004	Вал	Сталь	1	1.00	1.00
31	Підшипник B6004	Вал	Сталь	1	1.00	1.00
32	Підшипник B6004	Вал	Сталь	1	1.00	1.00
33	Підшипник B6004	Вал	Сталь	1	1.00	1.00
34	Підшипник B6004	Вал	Сталь	1	1.00	1.00
35	Підшипник B6004	Вал	Сталь	1	1.00	1.00
36	Підшипник B6004	Вал	Сталь	1	1.00	1.00
37	Підшипник B6004	Вал	Сталь	1	1.00	1.00
38	Підшипник B6004	Вал	Сталь	1	1.00	1.00
39	Підшипник B6004	Вал	Сталь	1	1.00	1.00
40	Підшипник B6004	Вал	Сталь	1	1.00	1.00
41	Підшипник B6004	Вал	Сталь	1	1.00	1.00
42	Підшипник B6004	Вал	Сталь	1	1.00	1.00
43	Підшипник B6004	Вал	Сталь	1	1.00	1.00
44	Підшипник B6004	Вал	Сталь	1	1.00	1.00
45	Підшипник B6004	Вал	Сталь	1	1.00	1.00
46	Підшипник B6004	Вал	Сталь	1	1.00	1.00
47	Підшипник B6004	Вал	Сталь	1	1.00	1.00
48	Підшипник B6004	Вал	Сталь	1	1.00	1.00
49	Підшипник B6004	Вал	Сталь	1	1.00	1.00
50	Підшипник B6004	Вал	Сталь	1	1.00	1.00
51	Підшипник B6004	Вал	Сталь	1	1.00	1.00
52	Підшипник B6004	Вал	Сталь	1	1.00	1.00
53	Підшипник B6004	Вал	Сталь	1	1.00	1.00
54	Підшипник B6004	Вал	Сталь	1	1.00	1.00
55	Підшипник B6004	Вал	Сталь	1	1.00	1.00
56	Підшипник B6004	Вал	Сталь	1	1.00	1.00
57	Підшипник B6004	Вал	Сталь	1	1.00	1.00
58	Підшипник B6004	Вал	Сталь	1	1.00	1.00
59	Підшипник B6004	Вал	Сталь	1	1.00	1.00
60	Підшипник B6004	Вал	Сталь	1	1.00	1.00
61	Підшипник B6004	Вал	Сталь	1	1.00	1.00
62	Підшипник B6004	Вал	Сталь	1	1.00	1.00
63	Підшипник B6004	Вал	Сталь	1	1.00	1.00
64	Підшипник B6004	Вал	Сталь	1	1.00	1.00
65	Підшипник B6004	Вал	Сталь	1	1.00	1.00
66	Підшипник B6004	Вал	Сталь	1	1.00	1.00
67	Підшипник B6004	Вал	Сталь	1	1.00	1.00
68	Підшипник B6004	Вал	Сталь	1	1.00	1.00
69	Підшипник B6004	Вал	Сталь	1	1.00	1.00
70	Підшипник B6004	Вал	Сталь	1	1.00	1.00
71	Підшипник B6004	Вал	Сталь	1	1.00	1.00
72	Підшипник B6004	Вал	Сталь	1	1.00	1.00
73	Підшипник B6004	Вал	Сталь	1	1.00	1.00
74	Підшипник B6004	Вал	Сталь	1	1.00	1.00
75	Підшипник B6004	Вал	Сталь	1	1.00	1.00
76	Підшипник B6004	Вал	Сталь	1	1.00	1.00
77	Підшипник B6004	Вал	Сталь	1	1.00	1.00
78	Підшипник B6004	Вал	Сталь	1	1.00	1.00
79	Підшипник B6004	Вал	Сталь	1	1.00	1.00
80	Підшипник B6004	Вал	Сталь	1	1.00	1.00
81	Підшипник B6004	Вал	Сталь	1	1.00	1.00
82	Підшипник B6004	Вал	Сталь	1	1.00	1.00
83	Підшипник B6004	Вал	Сталь	1	1.00	1.00
84	Підшипник B6004	Вал	Сталь	1	1.00	1.00
85	Підшипник B6004	Вал	Сталь	1	1.00	1.00
86	Підшипник B6004	Вал	Сталь	1	1.00	1.00
87	Підшипник B6004	Вал	Сталь	1	1.00	1.00
88	Підшипник B6004	Вал	Сталь	1	1.00	1.00
89	Підшипник B6004	Вал	Сталь	1	1.00	1.00
90	Підшипник B6004	Вал	Сталь	1	1.00	1.00
91	Підшипник B6004	Вал	Сталь	1	1.00	1.00
92	Підшипник B6004	Вал	Сталь	1	1.00	1.00
93	Підшипник B6004	Вал	Сталь	1	1.00	1.00
94	Підшипник B6004	Вал	Сталь	1	1.00	1.00
95	Підшипник B6004	Вал	Сталь	1	1.00	1.00
96	Підшипник B6004	Вал	Сталь	1	1.00	1.00
97	Підшипник B6004	Вал	Сталь	1	1.00	1.00
98	Підшипник B6004	Вал	Сталь	1	1.00	1.00
99	Підшипник B6004	Вал	Сталь	1	1.00	1.00
100	Підшипник B6004	Вал	Сталь	1	1.00	1.00

в)

Рис.12. Друге робоче вікно інтерфейсу програми:

а) друге вікно – план складання;

б) рекомендований план складання та граф зв'язків деталей;

в) друге вікно – план складання.

15

Висновки

- Тож було розроблено автоматизований планувальник послідовності складання на основі методології SADT, який безпосередньо приймає файли STEP як вхідні дані та усуває необхідність взаємодії людини під час підготовки до планування послідовності складання та створення виробничих процесів складання. Для цього було проведено дослідження математичної моделі 3D моделі САПР, функціональної моделі виробничого процесу складання; було розроблено метод вилучення даних САПР, необхідних для створення матриці зв'язків деталей у приладі, із файлів STEP, а на основі цих даних та завдяки генетичному алгоритму було розроблено алгоритм для оптимізації послідовності складання. Далі використовуючи всі вже отримані дані розробляються виробничі процеси складання.
- На основі розробленої програми було створено стартап-проект; визначено основних конкурентів та плюси продукту перед ними; проаналізовано всі загрози та методи їх вирішення; визначено базової концепції розвитку; для впровадження на ринок була розроблена система на основі стратегії зайняття конкурентної ніші. Отже, було визначено, що подальша імплементація продукту є доцільною, а проект можна вважати конкурентоспроможним в порівнянні з замінниками даного продукту.

16

Публікації

№ з/п	Назва праці	Назва видання та його вихідні відомості, що дозволяють ідентифікувати та відрізнити це видання від усіх інших	Кількість друкованих аркушів
1	2	3	4
1	Застосування методології SADT у технологічній підготовці виробництва	XVI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні», 08-09 грудня 2020 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2020. – С. 115-117.	3
2	Моделювання інформаційних процесів забезпечення технологічності виробів на ранніх етапах проектування	XIV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених "ПОГЛЯД У МАЙБУТНЄ ПРИЛАДОБУДУВАННЯ", 18-19 травня 2021р. - К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2021. – С. 162-166.	5
3	Ідентифікація вузлів виробу шляхом використання проектної інформації в 3D-моделях	XVII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні», 07-08 грудня 2021 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2021	4

Дякую за увагу

18