УДК 004.9: 658.512

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти *другий (магістерський)*

ГЮИК.504300.004 ПЗ
(позначення документу)

Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів
(тема)

Виконав:
Студент *2* курсу, групи *СПРм-18-2*
Спеціальність *122 – Комп'ютерні науки*
(код і повна назва напряму)
Тип програми *освітньо-наукова*
(освітньо-професійна або освітньо-наукова)
Освітня програма *Системне проектування*
(повна назва освітньої програми)

Сідорчук Є. І.
(прізвище, ініціали)

Керівник *Безкоровайний В.В.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ *Гребеннік І. В.*
(підпис) (прізвище, ініціали)

2020 р.

Атестаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.

Матеріали атестаційної роботи не містять відомостей, що заборонені до опублікування у відкритих виданнях.

Попередній захист проведено.

Керівник атестаційної роботи *В.В.Безкоровайний*

20.05.20

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*

(повна назва)

Кафедра *Системотехніки*

(повна назва)

Рівень вищої освіти *другий (магістерський)*

Спеціальність *122 – Комп'ютерні науки*

(код і повна назва)

Тип програми *освітньо-наукова*

(освітньо-професійна або освітньо-наукова)

Освітня програма *Системне проектування*

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____»_____ 20___ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові *Сідорчук Єлизаветі Іванівні*

(прізвище, ім'я, по батькові)

1. Тема роботи *«Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів»*

затверджена наказом по університету від *«30» березня 2020 р. № 477 Ст*

2. Термін подання студентом роботи (проекту) *25 травня 2020 р.*

3. Вихідні дані до роботи (проекту) *Структура технологічного процесу – лінійна. Кількість фаз – до 20. Кількість типів обладнання – до 10. Обов'язкові характеристики обладнання: продуктивність; вартість. Технічне забезпечення: IBM-сумісний персональний комп'ютер. Програмне забезпечення – можливість реалізації інтерактивного режиму.*

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити) *Вступ. Огляд проблеми проектування гнучких виробничих технологічних процесів (ТП). Гнучкі виробничі ТП як об'єкти проектування. Автоматизація процесів проектування ТП. Огляд систем автоматизованого проектування ТП. Постановка мети та задач дослідження. Методи структурно-параметричної оптимізації ТП. Алгоритми оптимізації ТП. Розробка програмного засобу. Вибір середовища розробки. Основні функції програмного засобу. Аналіз результатів експериментів. Висновки.*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, пла
Плакати або кресленики на аркушах формату А4 по тексту та/або у додатках:
технологічного процесу; схема системи автоматизації проектування ТП; схема ме
оптимізації ТП; ілюстрації результатів розв'язання задачі; графічне подання ча
характеристик методів розв'язання задачі.

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітк
1	*Отримання, аналіз завдання, уточнення*	*30.03.2020*	
	плану роботи		
2	*Аналіз сучасного стану проблеми*		
	проектування технологічних процесів (ТП)	*03.04.2020*	
3	*Огляд існуючих методів оптимізації ТП*	*06.04.2020*	
4	*Розробка математичного забезпечення задачі*	*13.04.2020*	
5	*Розробка програмного забезпечення задачі*	*20.04.2020*	
6	*Проведення експериментальних досліджень*	*28.04.2020*	
7	*Оформлення пояснювальної записки*	*03.05.2020*	
8	*Підготовка презентації*	*09.05.2020*	
9	*Подання закінченої роботи науковому*	*14.05.2020*	
	керівникові		
10	*Подання роботи на рецензування*	*21.05.2020*	
11	*Попередній захист*	*22.05.2020*	
12	*Подання роботи до екзаменаційної комісії*	*25.05.2020*	

Дата видачі завдання *30 березня 2020 р.*

Студент _____ *Сідорчук Є. І.*
(підпис)

Керівник роботи _____ *проф. Безкоровайний В.В.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи: 79 с., 6 табл., 22 рис., 4 додатки, 39 джерел інформації.

AFFINITY PROPAGATION, DBSCAN, K-MEANS, OPTICS, КЛАСТЕРИЗАЦІЯ, МЕТОД ПРОЦЕСУ-АНАЛОГА, СТРУКТУРНО-ПАРАМЕТРИЧНА ОПТИМІЗАЦІЯ, ТЕХНОЛОГІЧНИЙ ПРОЦЕС.

Об’єкт дослідження – лінійні дискретні виробничі технологічні процеси (ТП).

Предмет досліджень – структура та параметри обладнання лінійних дискретних виробничих технологічних процесів.

Метою дослідження є удосконалення методу структурно-параметричної оптимізації гнучких технологічних процесів дискретного виробництва.

Методи дослідження: системний підхід для аналізу технологічного процесу як об’єкта проектування; метод процесу-аналога для вибору базової структури ТП; методи кластеризації (k-means, DBSCAN, OPTICS, Affinity propagation), класифікації (Support Vector Machines, K-Nearest Neighbors) та статистичного аналізу для пошуку виробів та процесів-аналогів; метод спрямованого перебору для удосконалення базового ТП.

У роботі запропоновано удосконалення методу процесу-аналога для структурно-параметричної оптимізації лінійних дискретних виробничих процесів в частині зниження його часової складності, виконана його програмна реалізація. Розроблене математичне та програмне забезпечення для технології проектування ТП дозволить зменшити витрати на виробництво, збільшити конкурентоспроможність продукції, та підвищувати гнучкість виробництва.

Результати дослідження можуть бути використані в проектних компаніях, що займаються розробкою та реінжинірингом технологічних процесів, на виробничих підприємствах з гнучкими технологічними ланцюгами та в навчальному процесі закладів вищої освіти.

ABSTRACT

Research contains: 79 pages, 6 tables, 22 images, 4 applications, 39 sources.

AFFINITY PROPAGATION, CLUSTERIZATION, DBSCAN, K-MEANS, OPTICS, PROCESS-ANALOGUE METHOD, STRUCTURAL-PARAMETRIC OPTIMIZATION, TECHNOLOGICAL PROCESS.

The object of research is linear discrete production technological processes (TP).

The subject of research – the structure and parameters of the equipment of linear discrete production processes.

The purpose of the study is to improve the method of structural and parametric optimization of flexible technological processes of discrete production.

Research methods: a systematic approach to the analysis of the technological process as an object of design; the method of the analog process to select the basic structure of the TP; methods of clustering (k-means, DBSCAN, OPTICS, Affinity propagation), classification (Support Vector Machines, K-Nearest Neighbors) and statistical analysis for finding products and analogous processes; method of directed search to improve the basic TP.

The paper proposes the improvement of the process-analog method for structural and parametric optimization of linear discrete production processes in terms of reducing its time complexity, its software implementation is performed. Developed mathematical and software for TP design technology will reduce production costs, increase product competitiveness, and increase production flexibility.

The results of the study can be used in design companies engaged in the development and reengineering of technological processes, in industrial enterprises with flexible technological chains and in the educational process of higher education institutions.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AP – Affinity propagation.

k-NN – K-Nearest Neighbors.

ГТП – груповий технологічний процес.

ІПС – інформаційно-пошукова система.

МЗ – модуль з’єднання.

МП – модуль поверхонь.

САПР – система автоматизованого проектування.

САПР ТП – система автоматизованого проектування технологічних процесів.

ТП – технологічний процес.

ТТП – типовий технологічний процес.

ВСТУП

Сучасна економіка стикається з серйозними викликами. Перманентні економічні кризи, нестабільна не лише економічна, а й політична ситуація, а також військові, релігійні, етнічні конфлікти, санкції, торгові війни, протекціонізм та інше, визначає хаотичність умов виробництва. Це призводить до постійних змін ринків збуту, а, отже, планів виробництва. Тому на перший план замість потужних стабільних підприємств, які використовують фіксовану структуру виробництва, яка визначена технологічним процесом (ТП), виходять гнучкі виробництва, які мають гнучку структуру виробничого процесу, що дозволяє змінювати окремі частини ТП, тим самим забезпечуючи підвищення ефективності виробництва, за рахунок зниження витрат, збільшення продуктивності і підвищення якості виробництва.

Незважаючи на багаточисельні публікації, що присвячені вирішенню проблем адаптації та реінжинірингу ТП гнучких виробництв, потребують подальшого удосконалення методи оптимізації структур і параметрів гнучких виробничих процесів. Їх впровадження дозволить підвищувати ефективність систем автоматизованого проектування ТП і на цій основі скорочувати терміни переходу на випуск нових видів продукції, знижувати її собівартість.

Метою дослідження є удосконалення методу структурно-параметричної оптимізації гнучких технологічних процесів дискретного виробництва.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– виконати огляд проблеми проектування гнучких виробничих технологічних процесів;

– обрати ефективний базовий метод розв’язання задачі структурно-параметричної оптимізації дискретних виробничих технологічних процесів;

– виконати удосконалення методу структурно-параметричної оптимізації дискретних виробничих технологічних процесів;

– розробити алгоритми та програмне забезпечення розв’язання задачі структурно-параметричної оптимізації технологічних процесів;

– провести експериментальне дослідження ефективності базового й удосконаленого методів оптимізації технологічних процесів.

Об'єктом дослідження є лінійні дискретні виробничі технологічні процеси.

Предметом досліджень є структура та параметри обладнання лінійних дискретних виробничих технологічних процесів.

Методи дослідження: системний підхід для аналізу технологічного процесу як об'єкта проектування; метод процесу-аналога для вибору базової структури ТП; методи кластеризації (k-means, DBSCAN, OPTICS, Affinity propagation), класифікації (Support Vector Machines, K-Nearest Neighbors) та статистичного аналізу для пошуку виробів та процесів-аналогів; метод спрямованого перебору для удосконалення базового ТП.

Наукова новизна дослідження полягає в удосконаленні методу процесу-аналога для структурно-параметричної оптимізації лінійних дискретних виробничих процесів в частині зниження його часової складності.

Практичне значення одержаних результатів полягає в можливості підвищення ефективності лінійних дискретних виробничих технологічних процесів за рахунок підбору кращих за комплексним показником «продуктивність-вартість» варіантів обладнання. Практичне використання розробленого математичного і програмного забезпечення для технології проектування ТП дозволить зменшити витрати на виробництво, збільшити конкурентоспроможність продукції, та підвищувати гнучкість виробництва.

Результати дослідження можуть бути використані в проектних компаніях, що займаються розробкою та реінжинірингом технологічних процесів, на виробничих підприємствах з гнучкими технологічними ланцюгами та в навчальному процесі закладів вищої освіти.

За результатами дослідження підготовлено доповіді на: 24-й Міжнародний молодіжний форум «Радіоелектроніка і молодь в XXI столітті» (м. Харків); звітну наукову конференцію викладачів, докторантів, аспірантів та студентів Прикарпатського національного університету ім. В. Стефаника (м. Івано-Франківськ); міжнародну науково-практичну конференцію «Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі» (м. Київ) [1].

1 ОГЛЯД ПРОБЛЕМИ ПРОЕКТУВАННЯ ГНУЧКИХ ВИРОБНИЧИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

1.1 Гнучкі виробничі технологічні процеси як об’єкти проектування

Важливою особливістю сьогоднішнього виробництва, спрямованого на задоволення зростаючих потреб споживачів, є збільшення кількості дрібних серій виробів чи оброблюваних деталей і збільшення їх різноманітності, що викликає необхідність в частому перегляді технологічного процесу виробництва. Тому в даний час поряд з традиційними вимогами (високої продуктивності, точності і надійності) до обладнання висувають нову вимогу – гнучкість.

Гнучкість – це властивість виробничої системи швидко переналагоджуватися на випуск нової продукції, тобто переходити в межах встановлених технологічних можливостей з одного працездатного функціонального стану в інший для виконання чергового виробничого завдання або нової функції [2].

Розрізняють такі аспекти гнучкості:

– гнучкість стану – здатність системи функціонувати при різних зовнішніх і внутрішніх змінах;

– гнучкість дії – здатність розширювати свої можливості за рахунок включення нового обладнання;

– гнучкість технології – оцінюється числом операцій, що реалізуються в даній виробничій системі;

– організаційна гнучкість – оцінюється складністю переходу на обробку будь-якого виробу даної номенклатури.

Гнучке виробництво в порівнянні з традиційними має більш високу мобільність і дозволяє скоротити терміни освоєння нової продукції; високу продуктивність і якість продукції, що випускається; поліпшені умови праці; можливість скорочення виробничого циклу і зниження експлуатаційних витрат на виробництво [3].

Основними характеристиками гнучкого виробництва є [4]:

– багатономенклатурність виробництва, в якому переналагодження обладнання на випуск нового виробу здійснюється паралельно з випуском

виробів старої номенклатури або при мінімальних витратах часу на переналагодження;

– комплексно автоматизоване виробництво;

– багатоаспектна гнучкість;

– поняття гнучкості є ієрархічним: від окремих одиниць обладнання та операційних технологій до заводського виробничого процесу в цілому;

– виробництво ґрунтується на безлюдній (малолюдній) технології та дозволяє відмовитися від традиційної технічної та супровідної документації (безпаперові технології);

– здатність обробляти вироби заданої номенклатури в будь-якому порядку їх слідування;

– виробництво функціонує на основі принципу централізації обробки – об'єкт виробництва не вимагає додаткового доопрацювання на іншому обладнанні;

– виробництво функціонує, як правило, на основі уніфікованих чи типових технологіях.

Як властивість виробничої системи гнучкість має чотири основні ознаки [1]:

– повторюваність – здатність системи багаторазово повертатися до виконання раніше освоєних робіт після завершення даної роботи;

– універсальність – здатність системи обробляти різні вироби і в різних кількостях без будь-якої модифікації;

– пристосовність – здатність системи до переналагодження на новий виріб шляхом впливу ззовні або самонастроювання;

– адаптивність – здатність системи пристосовуватися до зміни зовнішніх або внутрішніх чинників в певних межах без порушення власного функціонування або втрати якості продукції.

Одним із шляхів забезпечення багатономенклатурного автоматизованого виробництва є гнучкі технологічні процеси. Вони базуються на тому, що основою створення гнучких автоматизованих виробництв є багатономенклатурний технологічний процес, за яким будуть виготовлятися вироби довільної номенклатури в заданих межах значень їх характеристик [5].

Забезпечити гнучкість технологічних процесів в певній мірі дозволяють типові, групові та модульні технології.

Типовий технологічний процес характеризується єдністю змісту і послідовності більшості технологічних операцій для групи виробів із загальними конструктивними ознаками.

В основі типової технології лежить класифікація виробів на класи – підкласи – групи – підгрупи – типи. Тип являє собою групу схожих виробів, серед яких вибирається типовий представник, що володіє найбільшою сукупністю властивостей виробів, які увійшли до цієї групи. На типовий представник розробляється типовий технологічний процес, за яким здійснюється виготовлення всіх виробів цього типу [6].

Тим самим типовий процес певною мірою дозволяє вирішити суперечність між великими витратами часу на розробку процесу і малими термінами на виготовлення виробу, так як витрати часу на розробку робочого технологічного процесу для виготовлення конкретного виробу різко скорочуються.

Типові процеси дозволяють уникати повторних і нових розробок при проектуванні робочих технологічних процесів, внаслідок чого полегшується праця технолога і скорочуються витрати часу на розробку [6].

В цілому типова технологія сприяє:

– скороченню різноманітності технологічних процесів і внесенню одноманітності у виготовлення подібних виробів;

– спрощенню розробки робочих технологічних процесів;

– скороченню різноманітності засобів технологічного оснащення технологічних процесів;

– розробці нових високоефективних технологічних процесів.

Груповий технологічний процес – це процес виготовлення групи виробів з різними конструктивними, але спільними технологічними ознаками [7].

Груповий процес знайшов застосування в дрібносерійному і серійному виробництві. Принципова сутність групової технології полягає в групуванні виробів за технологічною подібністю.

Якщо типова технологія спрямована на скорочення трудомісткості технологічної підготовки виробництва, підвищення ефективності технологічних процесів і поширення прогресивних рішень, то групова технологія призначена для підвищення ефективності виробничого процесу.

Груповий технологічний процес розробляють для комплексного виробу. На відміну від типового, комплексний виріб є «збірним», часто такого виробу

взагалі немає у дійсності, він поєднує в собі риси більшості виробів, що увійшли до групи. Для комплексного виробу розробляється технологічний процес і всі вироби цієї групи, будучи, як правило, простішими за комплексний виріб, виготовляють за даним технологічним процесом, пропускаючи окремі технологічні переходи. Всі вироби, закріплені за цим технологічним процесом, виготовляють партіями [7].

У якості комплексного виробу технологічної групи служить якийсь виріб з групи або штучно створений виріб. Наприклад, комплексна деталь формується таким чином: береться найбільш складна деталь, яка включає всі поверхні інших деталей і, якщо вона не містить всіх поверхонь, що містяться в інших деталях групи, то до неї штучно додають відсутні поверхні.

Застосування групової технології особливо ефективно тоді, коли на її основі в серійному і дрібносерійному виробництвах вдається створити групові, потокові або навіть автоматичні лінії виготовлення виробів або деталей окремих груп. Застосування групової технології тим ефективніше, чим більше технологічна група.

Практика впровадження типових і групових технологічних процесів показує, що, незважаючи на очевидні переваги, частка їх впровадження невисока. Однією з головних причин цього є нестача засобів класифікації виробів на типи, групи, якими користуються при розробці типових і групових процесів. Аналіз цих класифікацій показує, що в обох випадках в явному або неявному вигляді в якості відмінних ознак виступають не конструктивні, а технологічні характеристики. Це призводить до того, що на підприємствах, що розрізняються складом технологічних засобів і кваліфікацією працівників, одна і та ж номенклатура виробів буде розбита на різні групи. З іншого боку, якщо змінити на підприємстві технології й устаткування, що використовуються, то доведеться змінювати типи і групи. Щоб звести до мінімуму ці недоліки, треба класифікувати вироби на групи не тільки за технологічними, а й за конструктивними ознаками [8].

Однією з головних причин недоліків всіх видів технологічних процесів є опис виробів на геометричному рівні, коли деталь представляється сукупністю елементарних геометричних поверхонь, а складальна одиниця – сукупністю деталей як геометричних тіл.

Це призводить до того, що технолог, розробляючи технологічний процес, прагне виготовляти на операціях такі сукупності поверхонь, які дозволяють

досягти максимальної продуктивності. Однак при цьому часто порушуються зв’язки між поверхнями, обумовлені спільним виконанням функцій деталі. В результаті, по-перше, з’являється багатоваріантність технологічного процесу через велику кількість комбінацій поверхонь, що виготовляються на операціях, а по-друге, через виготовлення функціонально пов’язаних поверхонь на різних операціях виникають складні технологічні зв’язки, що призводять до необхідності введення додаткових операцій.

Все це призводить до необґрунтованої різноманітності технологічних процесів, підвищення трудомісткості їх розробки, виникають труднощі в типізації технологічних процесів і в групуванні деталей при розробці групових процесів.

Якщо ж деталь описувати функціональними блоками у вигляді модулів поверхонь (МП), об’єднаних спільним виконанням службових функцій, то геометрична ознака стає вторинною, а елементарні поверхні входять до складу модулів поверхонь і не є самостійними об’єктами при розробці технологічних процесів.

З огляду на обмежену номенклатуру МП і їх високу повторюваність, можна істотно знизити різноманітність технологічних операцій за складом МП, що виготовляються. У результаті спроститься розробка технологічних процесів, їх типізація і групування деталей при використанні групових процесів.

Усе викладене справедливо і для складальних технологічних процесів, якщо складальну одиницю розглядати як сукупність модулів з’єднання (МЗ).

З метою реалізації викладених переваг опису вироби як сукупності МП і МЗ, слід розглядати побудову технологічного процесу як компоновку з модулів виготовлення МП, що входять до складу деталі (складальної одиниці).

У зв’язку з цим процес отримав назву модульного технологічного процесу, відповідно він може бути одиничним, типовим, груповим процесом, і являє собою результат подальшого вдосконалення методики розробки технологічних процесів, починаючи з опису виробу.

Модульний технологічний процес – це технологічний процес, побудований з модулів процесів виготовлення МП або МЗ, що входять до складу виробу, що виготовляється. В основі модульного технологічного процесу лежить об’єктивне існування МП і МЗ, які є конструктивними елементами виробів [8].

Оскільки модульне технологічне забезпечення розробляється під типові МП і МЗ з уніфікованими характеристиками, то воно відрізняється високим рівнем узагальнення, отже, широкою сферою застосування.

Найбільшу популярність для гнучких виробничих систем отримала система організації виробництва під назвою «Канбан». У системі «Канбан» під виробничим процесом розуміється безперервне потокове багатопредметне виробництво, в якому «перемішуються» різні вироби, на сьогодні це виріб одного найменування, але різних модифікацій.

Гнучкий технологічний процес створюється з обмеженого числа уніфікованих технологічних елементів, що забезпечує можливість його простої раціональної перебудови в умовах постійно мінливої номенклатури, серійності та партійності виготовлених виробів і конкретної виробничої ситуації. Особливістю проектування гнучкого технологічного процесу є те, що технологія є не просто алгоритмом дій по виготовленню виробів, а методичною основою створення і управління виробництвом [5].

У процесі проектування до гнучких технологічних процесів висуваються наступні вимоги:

– простота переналагодження на виготовлення виробів різної конфігурації, точності та типорозміру;

– високий ступінь взаємозамінності різних елементів і стадій технологічного процесу;

– низька чутливість до похибок попередніх стадій виготовлення виробів;

– можливість автоматизації простими засобами;

– можливість паралельної обробки заготовок, їх поверхонь і складання виробу;

– мінімальний час знаходження виробів в «очікуванні».

Як наслідок з цього, ТП, що реалізуються в гнучких виробничих системах, повинні володіти такими основними властивостями [2]:

– інваріантністю структури – властивістю, що характеризує незмінність структури ТП при виготовленні виробів з різними конструктивними ознаками і характеристиками виробів, що виготовляються;

– альтернативністю – властивістю, що характеризує можливість реалізації ТП за допомогою одного з передбачених варіантів як по послідовності

виконання робіт, так і за ступенем їх концентрації і використовуваних засобів технологічного оснащення;

– універсальністю – властивістю, що характеризує можливість ефективного виготовлення виробів досить широкої номенклатури в установлених межах значень їх конструктивних параметрів і характеристик;

– автоматизацією – властивістю, що характеризує можливість автоматизації основних і допоміжних засобів економічно раціональними способами і можливість програмного управління ТП і його складовими частинами;

– стійкістю – властивістю, що характеризує стабільність показників якості виробів, що виготовляються і ефективність при зміні в певному діапазоні зовнішніх умов (при відмінності в якості вихідного матеріалу, коливань припуску заготовок тощо);

– інтегрованістю – властивістю, що характеризує ступінь завершеності робіт з виготовлення виробів.

1.2 Автоматизація проектування технологічних процесів

Сучасне виробництво являє собою складний процес перетворення сировини, матеріалів, напівфабрикатів та інших предметів праці в готову продукцію, що задовольняє потребам суспільства [9].

Виробничим процесом називається множина всіх дій людей і знарядь праці, що здійснюються на підприємстві з метою виготовлення конкретних видів продукції.

Виробничі процеси складається з процесів трьох видів (рис. 1.1) [10]:

– основні технологічні процеси – це технологічні процеси, під час яких відбуваються зміни геометричних форм, розмірів і фізико-хімічних властивостей продукції;

– допоміжні процеси – це процеси, які забезпечують безперебійне протікання основних процесів (виготовлення і ремонт інструментів і оснастки; ремонт устаткування; забезпечення всіма видами енергій (електроенергією, теплом, парою, водою, стисненим повітрям тощо));

– обслуговуючі процеси – це процеси, пов’язані з обслуговуванням як основних, так і допоміжних процесів і не створюють продукцію (зберігання, транспортування, технічний контроль тощо).

Рисунок 1.1 – Структура виробничих процесів

В умовах автоматизованого, автоматичного і гнучкого інтегрованого виробництв допоміжні та обслуговуючі процеси в тій чи іншій мірі об’єднуються з основними і стають невід’ємною частиною процесів виробництва продукції.

Основною частиною виробничого процесу є технологічні процеси, які містять цілеспрямовані дії по зміні та визначенню стану предметів праці [11].

Метою проектування ТП є розробка його опису. У свою чергу опис ТП – це представлений на деякій мові виклад способу виготовлення виробу. Він складається з упорядкованого набору описів технологічних прийомів, що включає в себе відомості про типи і режими роботи, використовувані засоби технологічного оснащення, технологічні інструкції, норми часу та норми витрат матеріалів.

Опис ТП міститься в технологічних документах, об’єднаних в комплект. Склад комплекту залежить від ступеня деталізації опису, регламентованої нормативними документами та залежить від типу виробництва. Чим тип виробництва ближче до багатосерійного і масового, тим докладніше опис і тим більша кількість документів входить в комплект.

До технологічних процесів висувають наступні вимоги [11]:

– технологічний процес розробляється для виробництва продукту або вдосконалення чинного технологічного процесу відповідно до досягнень науки і техніки;

– технологічний процес розробляється для продуктів (предметів торгівлі), форма (конструкція) яких відпрацьована на технологічність;

– технологічний процес повинен бути прогресивним і забезпечувати підвищення ефективності праці та якості продуктів (предметів торгівлі), скорочення трудових і матеріальних витрат на його реалізацію;

– технологічний процес розробляють на основі наявного типового або групового технологічного процесу, а при їх відсутності на основі використання раніше прийнятих прогресивних рішень, що містяться в чинних одиничних технологічних процесах виготовлення аналогічних продуктів (предметів торгівлі);

– технологічний процес повинен відповідати вимогам техніки безпеки, промислової санітарії та охорони навколишнього середовища.

Технологічні процеси діляться на фази. Фаза – комплекс робіт, виконання яких характеризує завершення певної частини технологічного процесу і пов’язане з переходом предмета праці з одного якісного стану в інший [12].

Наприклад, у машинобудуванні та приладобудуванні технологічні процеси в основному діляться на три фази (рис. 1.2):

– заготівельна;

– обробна;

– складальна.

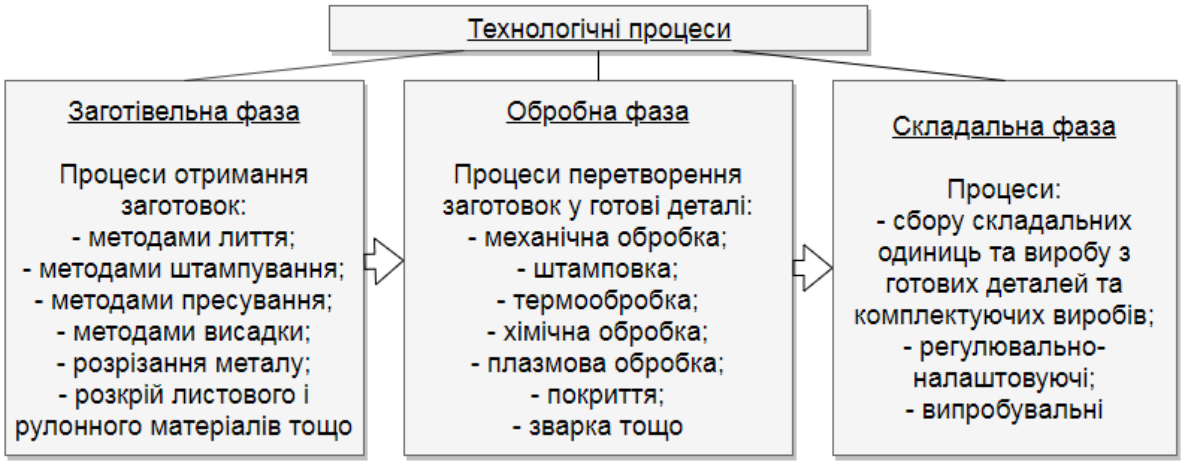

Рисунок 1.2 – Фазна структура технологічних процесів

Технологічний процес складається з послідовно виконуваних над цим предметом праці технологічних дій – операцій.

Операція – частина технологічного процесу, виконувана на одному робочому місці (верстаті, стенді, агрегаті тощо), що складається з ряду дій над кожним предметом праці або групою спільно оброблюваних предметів.

Операції, які не ведуть до зміни геометричних форм, розмірів, фізико-хімічних властивостей предметів праці, відносяться до нетехнологічних операцій (транспортні, вантажно-розвантажувальні, контрольні, випробувальні, комплектувальні тощо) [11].

Операції різняться також залежно від застосовуваних засобів праці [12]:

– ручні, виконувані без застосування машин, механізмів та механізованого інструменту;

– машинно-ручні – виконуються за допомогою машин або ручного інструменту при безперервній участі робітника;

– машинні – виконуються на верстатах, установках, агрегатах за обмеженої участі робітника (наприклад, установка, закріплення, пуск і зупинка верстата, розкріплення і зняття деталі). Решта виконує верстат;

– автоматизовані – виконуються на автоматичному обладнанні або автоматичних лініях.

Апаратурні процеси характеризуються виконанням машинних і автоматичних операцій в спеціальних агрегатах (печах, установках, ваннах тощо).

В наш час системи автоматизованого проектування технологічних процесів (САПР ТП) найчастіше визначають як програмні продукти, що дозволяють автоматизувати процес підготовки виробництва та саме проектування ТП [13].

САПР ТП використовують в проектних, конструкторських, технологічних організаціях та на підприємствах з метою:

– підвищення якості продукції, що проектується та випускається;

– підвищення техніко-економічного рівня об’єктів проектування;

– зменшення термінів і трудомісткості проектування.

Основними завданнями, які необхідно вирішувати при автоматизованому проектуванні технологічного процесу [14]:

– формалізація відомостей про деталі (вироби), які при традиційному ручному проектуванні задаються у вигляді креслення з множиною спеціальних позначень і переліку технічних вимог, викладених у вигляді опису (тексту). Цю інформацію при автоматизованому проектуванні необхідно подати в букво-цифрових кодах. До такого вигляду необхідно привести всю інформацію про деталі, включаючи опис її конфігурації, розмірних зв'язків, технічних вимог;

– створення інформаційних масивів, що містять відомості про обладнання і його технічні характеристики, інструменти, заготівельне виробництво, стандарти, керівні та нормативні матеріали. Для організації проектування необхідно створити інформаційно-довідкову службу, яка могла б забезпечити процес проектування необхідною довідковою документацією. При цьому потрібно не тільки забезпечити формалізований опис та введення цієї інформації в комп'ютер, а й розробити методи пошуку необхідної інформації в пам'яті, а також її виведення (подання) в потрібному вигляді;

– розробка множини типових рішень, на яких базується процес автоматизованого проектування, й алгоритмів їх вибору. Ці рішення потрібно подати формальною мовою, організувати введення, розміщення в пам'яті комп'ютера та передбачити можливість оперативної роботи з ними;

– організація виведення результатів роботи САПР у вигляді екранних зображень, роздруківок (або в іншому вигляді) технологічних карт або іншої документації. Тому потрібні програми для виведення результатів проектування у вигляді, зручному для технологів і робітників-верстатників.

Таким чином, сенс процесу проектування в будь-якій САПР, незалежно від об'єкта проектування, один і той же: отримати відповідно до задуму таку інформаційну систему (модель), яка дозволяє створити систему (оригінал), яка повністю відповідає задуму.

Функціональність існуючих комерційних САПР ТП орієнтована на автоматизацію загальних завдань технологічного проектування для різних видів виробництва: формування структури ТП, підбір обладнання і засобів оснащення, випуск технологічної документації, підтримка довідкових інформаційних масивів. Методологія автоматизованого проектування в САПР ТП у даний час досить відпрацьована та ґрунтується на наборі загальних методів [15]:

– діалогове проектування (з використанням баз даних);

– проектування на основі технічного процесу-аналога;

– проектування із застосуванням часто повторюваних технологічних рішень;

– проектування на основі групових і типових технологічних процесів;

– проектування на основі технологічного опису (кодування) геометрії оброблюваних поверхонь.

Системи САПР ТП у наш час перебувають у фазі розвитку. Серед відомих продуктів можна виділити такі системи як TechCard, T-FLEX, ВЕРТИКАЛЬ тощо.

Процес проектування ТП можна подати у вигляді сукупності алгоритмів, кожен з яких вирішує конкретну задачу і визначає допустиму множину рішень для наступних. Алгоритми будують максимально незалежними один від одного. Отже, є можливість створити бібліотеку універсальних алгоритмів, з яких в подальшому можна генерувати конкретну систему проектування ТП. Узагальнена структурна схема САПР ТП наведена на рисунку 1.3 [16].

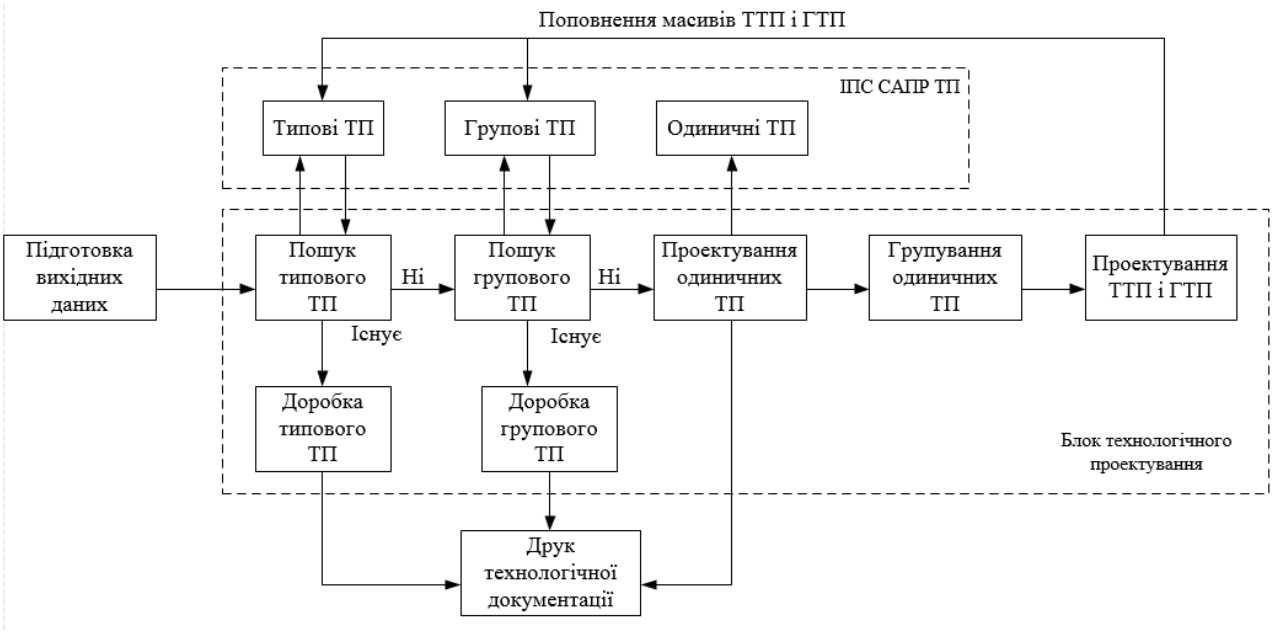

Рисунок 1.3 – Узагальнена структурна схема САПР ТП

Ефективність ТП багато в чому визначається рішеннями, які приймаються уже на етапах їх проектування чи/або реінжинірингу. Формування рішень, що задовольняють технологічним і вартісним обмеженням, передбачає створення інформаційної технології розв’язання множини задач структурної, топологічної та параметричної оптимізації ТП [17].

На нижньому рівні декомпозиції проблеми системної оптимізації ТП як просторово розподіленого об'єкта Pr виділяється множина задач [18]: $Pr = \{ Task_i \},\ i = \overline{1,6}$, де $Task_1$ – вибір принципів побудови ТП; $Task_2$ – вибір структури ТП; $Task_3$ – оптимізація розміщення обладнання ТП; $Task_4$ – вибір алгоритмів функціонування ТП; $Task_5$ – визначення типів чи параметрів обладнання ТП; $Task_6$ – оцінка показників та вибір найкращого варіанту побудови ТП.

У рмках агрегативно-декомпозиційного підходу, методології системного аналізу та системного проектування обрана мережева модель задачі системної оптимізації ТП. На її основі створюється логічна схема, що дозволяє визначити раціональну послідовність розв'язання комплексу задач проблеми системної оптимізації ТП.

Для схеми системної оптимізації ТП S_{TP} визначено п'ятірку множин:

$$S_{TP} = <Tasks, InDat, Res, DesDec, ProcDec>, \qquad (1.1)$$

де $Tasks = < Task_i >,\ i = \overline{1,6}$ – впорядкована множина задач оптимізації ТП;

InDat – множина вхідних даних задач;

Res – множина обмежень задач;

DesDec – множин оптимізаційних рішень;

ProcDec – вирішальна процедура, яка ставить кожній парі вхідних даних та обмежень $<InDat_i, Res_i>$ непорожню підмножину оптимізаційних рішень $\{ DesDec_i \},\ i = \overline{1,6}$.

Моделі задач системної оптимізації ТП, що враховують їхні взаємозв'язки, подаються у вигляді:

$$ModTask_i = \{ InDat_{iI}, InDat_{iE}, Res_i \} \rightarrow DesDec_i,\ i = \overline{1,6}, \qquad (1.2)$$

де $InDat_{iI}$ – множина зовнішніх по відношенню до комплексу задач оптимізації ТП (1.1) вхідних даних;

$InDat_{iE}$ – множина внутрішніх по відношенню до комплексу задач (1.1) вхідних даних;

Res_i – множина обмежень i-ої задачі, $i = \overline{1,6}$;

$DesDec_i$ – розв'язок i-ої задачі, $i = \overline{1,6}$.

На рисунку 1.4 зображена структурна схема технологічного процесу. Через $О_{ij}$ позначено обладнання, необхідне для виготовлення об'єкта (деталі); i – номер фази у технологічному процесі об'єкта, $i = \overline{1,n}$ – кількість фаз у технологічному процесі; j – номер обладнання в рамках фази технологічного процесу об'єкта, $j = \overline{1,m}$; m – загальна кількість обладнання в рамках фази технологічного процесу об'єкта; Ч – черги, що можуть утворюватися при переходами між фазами.

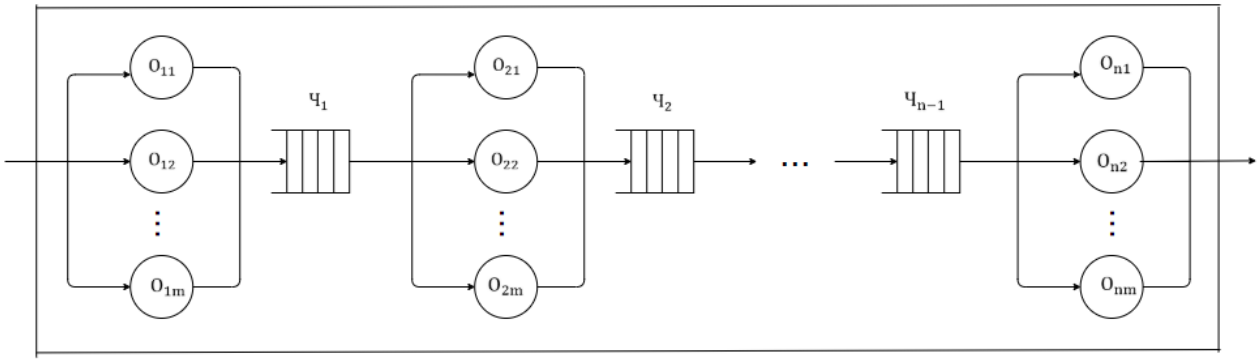

Рисунок 1.4 – Структурна схема лінійного технологічного процесу

Формування технологічного процесу – це складний процес, який включає параметричний та структурний синтез з подальшим аналізом та прийняттям проектних рішень. Окрім розрахункових задач, прийняття рішення по кожній задачі проходить шляхом вибору відомих типових рішень, при цьому необхідно враховувати умови застосовності. Для цього слід описати всі можливі типові рішення, а також умови, через які кожне з рішень може бути обрано [19].

Можна виділити наступні методи автоматизованого проектування ТП:

– метод повторного використання одиничних ТП;
– метод уніфікованого технологічного процесу;
– метод синтезу ТП.

Метод повторного використання одиничних технологічних процесів заснований на використанні готових технологічних рішень шляхом запозичення їх з існуючих одиничних технологічних процесів [19]. Схема проектування технології при використанні розглянутого методу може бути представлена

наступною послідовністю: деталь – деталь-аналог – технологічний процес на деталь-аналог – технологічний процес на деталь.

Проектування технології здійснюється в два етапи: пошук аналогів і подальше перетворення процесів-аналогів. На першому етапі по заданому пошуковому образу в базі даних системи відшукуються деталі-аналоги та технологічні процеси їх виготовлення. Пошук аналогів відбувається з урахуванням принципу «технологічної подоби оброблюваних деталей, технологічних процесів та їх елементів».

На другому етапі здійснюється перетворення процесу-аналога, яке полягає у виключенні або додаванні структурних елементів в технологічному процесі на основі виявлених відмінностей між оброблюваною деталлю і операційними станами деталі-аналога.

Формалізація процесу проектування в даному випадку полягає, по-перше, в розробці методу формування пошукового образу деталей і технологій-аналогів і методу пошуку аналогів, по-друге, в розробці методу перетворення процесів-аналогів. Шуканий образ деталі або процесу складається з набору системних характеристик і їх допустимих відхилень. При цьому дві деталі або дві технології вважаються технологічно подібними (тобто аналогічними), якщо вони мають однин і той же набір системних характеристик, що відрізняються між собою в межах заданих відхилень своїх значень [19].

При виконанні операції «перетворення» процесу-аналога приймаються рішення про виключення або додавання структурних елементів в технологічному процесі на основі аналізу технологічної подоби й операційних станів деталі-аналога й оброблюваної деталі. Досвід застосування даного методу в автоматизації проектування технологій показує, що процес, спроектований за аналогією з існуючим, часто не є оптимальним, тому що він багато в чому залежить від досвіду проектанта і заснований на використанні «випадкового» процесу, не завжди кращого для конкретних умов.

Метод уніфікованого технологічного процесу. Уніфікований технологічний процес являє собою типовий або груповий технологічний процес, що містить готові технологічні рішення, які приймаються при розробці технологій для конкретних деталей. Схема проектування технології з використанням даного методу може бути представлена наступною послідовністю: деталь – УТП – робочий технологічний процес.

Під технологічними рішеннями, що містяться в УТП у вигляді ряду його незмінних елементів, розуміються відомості про заготовки, маршрути (плани) обробки, обладнання що застосовується, пристроях, інструментах.

Вони є загальними для всіх деталей даного типу або групи [20]. Уніфікований технологічний процес створюється на кожну комплексну деталь-представник групи після класифікації і групування деталей за ознаками конструктивно-технологічної подоби. Відмінною рисою комплексної деталі є те, що вона містить в собі геометричні образи всіх деталей групи.

Будь-яку деталь даної групи можна отримати з комплексної шляхом «відсікання» елементів, які не належать даній деталі. Таким чином, технологічний процес виготовлення реальної деталі створюється в результаті послідовної вибірки операцій і переходів з УТП відповідно до особливостей її геометричного образу. Залежно від ступеня конструктивно-технологічної подоби деталей групи УТП може бути типовим або груповим. Груповий технологічний процес за всіма показниками збігається з типовим. Однак для конкретної деталі групи він може містити надлишкову інформацію, яка не включається в реальний технологічний процес [20].

Формалізм даного методу полягає в тому, що конкретні технологічні рішення вибираються за заданими критеріями або пріоритетам з множини рішень, допустимих в конкретній ситуації, яке попередньо формується з множин всіляких технологічних рішень, що містяться в УТП.

Системи автоматизованого проектування ТП, побудовані на основі даного методу, мають високий ступінь автоматизації праці технолога. Однак за допомогою подібних систем можливо проектувати технологічні процеси тільки на деталі, які за своїми конструктивно-технологічним ознаками відносяться до груп, охоплених цими системами.

Метод синтезу технологічного процесу заснований на дослідженні багаторівневої декомпозиції процесів технологічного проектування та типізації технологічних рішень на рівні переходу. Для кожної поверхні деталі проводиться типовий поділ на проміжні стани, і вибираються методи їх досягнення. Проектування технології відбувається на основі аналізу розмірних зв’язків елементів деталі, структурних зв’язків у заготовки та деталі, синтезу схем базування і структури операції [21].

Автоматизоване проектування технологічних процесів, засноване на цьому методі, в загальному випадку відбувається в чотири етапи. На першому

етапі синтезується принципова схема технологічного процесу. Для цього вирішуються такі завдання: призначення заготівельного етапу технологічного процесу; формування чорнових і чистових етапів обробки деталі.

На другому етапі (синтез технологічного маршруту) вирішуються завдання: розчленовування множини методів обробки поверхонь деталі на укрупнені операції; впорядкування укрупнених операцій; розбиття укрупнених операцій на прості; вибір обладнання для кожної простої операції; вибір технологічних баз і схем базування деталі; визначення послідовності операцій.

На третьому етапі синтезуються технологічні операції. Для цього виконується: визначення міжопераційних розмірів; розчленовування операцій на переходи; вибір засобів технологічного оснащення; визначення режимів різання і норм часу; вибір оптимального варіанта операцій. При необхідності здійснюється четвертий етап розробки технологічного процесу – синтез керуючих програм.

Системи автоматизованого проектування ТП, що реалізують розглянутий метод, мають високий ступінь автоматизації праці технолога. Однак досвід створення і впровадження подібних систем показав, що розробка алгоритмів і програм, що реалізують синтез технологічного маршруту і операцій, утруднена складністю синтезу; зниження кількості синтезованих варіантів технологічних процесів з метою скорочення часу і вартості їх розробки призводить до високої частки типових рішень, що враховують специфіку конкретного підприємства [21].

1.3 Огляд систем автоматизованого проектування технологічних процесів

Програмне забезпечення системи TECHCARD включає в себе базові модулі для вирішення задач технологічного проектування та інформаційного забезпечення (бази даних). Система включає окремі підсистеми, що функціонують як в загальному комплексі, так і в автономному режимі [22]:

– підсистема проектування технологічних процесів – дозволяє реалізувати призначення маршрутів, вибір і розрахунок заготовок, розрахунок трудомісткості;

– SEARCH – організація і ведення архіву технологічної та конструкторської документації, управління інформацією про кінцеві вироби;

– CADMECH-T – модуль автоматизації проектування креслень машинобудування для оформлення та побудови як операційних ескізів, так і всіляких графічних зображень, що виводяться в результуючий технологічний документ у середовищі AutoCAD;

– ROTATION – підсистема проектування об’єктів типу «тіло обертання»;

– IMBASE – інформаційно-довідна база даних стандартних матеріалів і елементів;

– TechCard – редактор бази знань – модуль настройки експертної системи, який включає в себе таблиці і формули для автоматичного розрахунку технологічних параметрів. Інформаційне забезпечення системи включає базу даних технологічного призначення та дозволяє створити єдину інтегровану програмне та інформаційне середовище стосовно різних видів виробництва [23].

T-FLEX Технологія – повнофункціональна система автоматизованого проектування, що включає гнучкі сучасні засоби розробки технологічних проектів будь-якої складності. У системі реалізована спеціалізована технологічна мова, застосовується метод проектування за процесом-аналогом, є можливість роботи всіх технологічних підрозділів в єдиному інформаційному просторі [24].

Кінцевим результатом роботи системи T-FLEX Технологія є не тільки створення і наповнення бази знань за технологічними процесами, а й випуск технологічної документації.

T-FLEX Технологія спільно з T-FLEX DOCs дозволяє автоматично відстежувати стан робіт над кожним ТП, автоматично видавати завдання виконавцям, а також надає дані для оцінки термінів відставання від графіка робіт, сповіщає зацікавлених користувачів про завершення окремих етапів контрольованих бізнес-процесів.

Система САПР ТП ВЕРТИКАЛЬ являє собою систему, призначену для автоматизованого вирішення задач технологічного проектування. До складу системи входять окремі підсистеми, які можуть функціонувати як автономно, так і в комплексі. Вона працює і використовує дані в єдиному інформаційному просторі [25].

САПР ТП ВЕРТИКАЛЬ дозволяє:

– проектувати технологічні процеси в автоматизованому режимі;

– розраховувати матеріальні і трудові витрати виробництва;

– формувати всі необхідні комплекти технологічної документації, що використовуються на підприємстві;

– організувати і розвивати технологічні бази даних підприємства;

– передавати дані в різні системи планування і управління (класів PDM/MRP/ERP), а також організовувати спільну роботу з модулями і додатками, розробленими на підприємстві.

Дана система забезпечує автоматичне перенесення даних з креслення, тривимірної моделі, а також даних про деталі (збірки), даних про матеріалі і заготівки.

TechnologiCS – це інформаційна система, розроблена спеціально для машинобудівних заводів. Основне призначення системи – підвищення ефективності процесів конструкторсько-технологічної підготовки, планування й управління виробництвом за рахунок широкого застосування конструкторсько-технологічної інформації в електронному вигляді й організації колективної роботи з електронними даними [26].

Можливості TechnologiCS:

– інтеграція з CAD/CAM/CAE системами;

– перегляд 3D-моделі безпосередньо при роботі з електронними довідниками, специфікаціями, ТП в TechnologiCS;

– робота з документами в електронному архіві при колективній роботі з 3D-моделями складальних одиниць, з коректним розподілом прав доступу при застосуванні запозичених деталей, створення інтерфейсів до різних CAD-систем;

– робота зі специфікаціями;

– ведення складського обліку тощо.

TechnologiCS 4 – досить велика і потужна система, що дозволяє пов’язати рішення відразу цілого спектру завдань машинобудівного підприємства: від розробки виробів до контролю витрат матеріалів на його виготовлення.

Система «МАС ПТП» (Многопользовательская Автоматизированная Система Проектирования ТП) є системою з середнім рівнем автоматизації. Принципова особливість системи – виконання інтерактивного проектування маршрутних і операційних технологічних процесів безпосередньо в середовищі PDM-системи SMARTEAM і, отже, в єдиному інформаційному середовищі з конструкторами та всіма іншими фахівцями, інформація від яких також використовується при проектуванні технологічних процесів.

У системі виконуються основні функції проектування ТП і є можливість формувати та зберігати в електронному архіві комплекти з технологічними картами. Зазначена особливість системи дає можливість розпаралелювати розробку ТП і легко відстежувати проходження всіх стадій затвердження документації. При проектуванні використовується база даних і знань. Вона також функціонує в середовищі SMARTEAM [22].

1.4 Постановка мети та задач дослідження

Проведений аналіз показав, що незважаючи на велику кількість публікацій, які присвячені вирішенню проблем реінжинірингу та адаптації технологічних процесів гнучких виробництв, методи оптимізації структур і параметрів гнучких виробничих процесів потребують подальшого удосконалення. Їх впровадження дозволить підвищувати ефективність систем автоматизованого проектування технологічних процесів та завдяки цьому зменшувати терміни переходу на випуск продукції нового виду, знижувати її собівартість.

Задача структурно-параметричної оптимізації лінійного ТП з однотипним обладнанням кожної фази розглядається у такій постановці. Задано: кількість фаз (операцій) ТП n; кількість типів обладнання, що може бути задіяне на кожній з фаз m_i, $i=\overline{1,n}$; продуктивність обладнання p_{ij}, $i=\overline{1,n}$, $j=\overline{1,m_i}$; приведена вартість обладнання c_{ij}, $i=\overline{1,n}$, $j=\overline{1,m_i}$; необхідна продуктивність ТП p^*.

Необхідно знайти варіант побудови ТП $v(k,x)\in V$, що визначається його структурою (кількістю обладнання на кожній з фаз k_i, $i=\overline{1,n}$) і параметрами (типом обладнання на кожній з фаз), який з мінімальними приведеними витратами забезпечує необхідну продуктивність:

$$C(v)=\sum_{i=1}^{n}\sum_{j=1}^{m_i} k_i c_{ij} x_{ij} \to \min_{x,k},$$

$$\sum_{i=1}^{n} x_{ij}=1 \ \ \forall j=\overline{1,m_i}, \ \ \sum_{j=1}^{m_i} x_{ij}=1 \ \ \forall i=\overline{1,n}, \ \ \sum_{i=1}^{n} p_{ij}x_{ij}\geq p^* \ \ \forall i=\overline{1,n}, \qquad (1.3)$$

де $x = [\, x_{ij} \,]$, $i = \overline{1,n}$, $j = \overline{1,m_i}$ – матриця, що визначає тип обладнання кожної з фаз (x_{ij} – булева змінна: $x_{ij} = 1$, якщо на i-тій фазі використовується обладнання j-го типу; $x_{ij} = 0$ – в інших випадках);

$k = [\, k_i \,]$, $i = \overline{1,n}$ – вектор, що визначає кількість обладнання на кожній з фаз ТП.

З метою спрощення процедури розв'язання задачі (1.3) її запропоновано реалізовувати на основі підходу повторного використання одиничних ТП. Він заснований на адаптації готових технологічних рішень шляхом запозичення їх з існуючих одиничних ТП за схемою:

Деталь → Деталь-аналог → ТП на деталь-аналог → ТП на деталь.

Пошук аналогів відбувається з урахуванням принципу «технологічної подібності» оброблюваних деталей, технологічних процесів та їх елементів.

Для пошуку аналогів запропоновано використовувати методи кластерного аналізу, що дозволяють здійснювати автоматичне розбиття деталей і процесів на групи (кластери) за принципом схожості ТП.

У залежності від типу та структури об'єкта-аналога змінюється структура технологічного процесу, кількість і послідовність етапів, операцій, переходів, типів обладнання. Засіб проектування включає множину методів оптимізації ТП, що розрізняються за точністю та часовою складністю. У ньому передбачений інтерактивний режим, що дозволяє враховувати знання та досвід проектувальників.

2 МЕТОДИ ОПТИМІЗАЦІЇ СТРУКТУРИ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

2.1 Визначення виробу-аналога методом кластеризації

Серед методів оптимізації технологічних процесів у наш час особливої популярності набули методи на основі вибору аналогів. Для групування об’єктів з подібними технологічними процесами може використовуватися кластерний аналіз. Він дозволить розбити всі вироби на кластери, які будуть характеризуватися схожими технологічними процесами.

Кластерний аналіз – це метод класифікаційного аналізу. Його основне призначення – розбиття множини досліджуваних об’єктів і ознак на однорідні в деякому сенсі групи, або кластери [27]. Він відноситься до багатовимірних статистичних методів, тому передбачається, що вихідні дані можуть бути значного обсягу, тобто істотно великим може бути як кількість об’єктів дослідження (спостережень), так і ознак, що характеризують ці об’єкти.

Даний метод може використовуватися для групування виробів чи технологічних процесів, як об’єктів за їх структурними, функціональними, іншими характеристиками. Методи кластерного аналізу дозволяють розбивати існуючі вироби-аналоги та ТП виготовлення аналогів на кластери, які будуть характеризуватися схожими показниками [28].

Наприклад, будь-яку деталь, що виготовляється можна представити у вигляді множини значень конструктивно-технологічних ознак:

$$D_i = \{O_{i1}, O_{i2}, \dots, O_{il}\}, \;\; i = \overline{1, m},$$

де O_{ij} – значення i -тої ознаки j-тої деталі;

m – кількість розглянутих деталей;

l – кількість ознак, що характеризують кожну деталь.

Ознаки враховують як форму, розміри і технологічні вимоги виготовлення деталей і окремих її поверхонь, так і програму випуску, матеріал деталі, вид заготовки тощо.

На підставі значень конструктивно-технологічних ознак всі деталі, що виготовляються підприємствами можна об’єднати в окремі групи, що характеризуються спільністю конструкції, розмірів, функціонального

призначення та технологічних процесів їх виготовлення. Із усієї сукупності деталей (об'єктів) D_i, що виготовляються і описуються l ознаками (властивостями), можна сформулювати k груп:

$$S = \{S_1, S_2, \dots, S_k\}, \;\; 1 \le k \le m,$$

де S_i – групи, сформовані за ознаками (властивостями);

k – кількість сформованих груп;

m – кількість розглянутих деталей.

Різні варіанти розбиття об'єктів на k груп оцінюються по одному з критеріїв таксономії. Якщо ознаки (властивості) об'єкта приймають кількісні значення, представляються у вигляді координат метричного простору, то кожен об'єкт зі своїми значеннями буде відображатися у вигляді точки цього простору.

Два об'єкти з майже однаковими ознаками відображаються у вигляді двох близьких точок, а об'єкти з ознаками, які суттєво відрізняються, будуть представлені точками, що розташовані далеко одна від одної.

Згустки точок можна виділити в окремі структурні частини множини деталей – в групи (кластери, таксони). Таким чином, можна отримати опис k груп, кожна з яких об'єднує деталі з близькими за значенням ознаками. Надалі кожну нову деталь, що підлягає виготовленню чи оптимізації, можна автоматично віднести до тієї чи іншої групи. Отримана таким чином група характеризується областю застосовуваного обладнання, інструменту, оснащення, операцій і переходів технологічного процесу.

Наприклад, деталі однієї групи мають подібну геометричну форму, близькі за величиною розміри, подібні за методом отримання заготовки, однакові точність і шорсткість основних поверхонь тощо, можуть виготовлятися в одному цеху, дільниці та в одних виробничих умовах.

Задача групування деталей на класи (кластери, таксони) вирішується успішно, якщо до багатовимірних об'єктів, до яких можна віднести деталі, застосувати методи розпізнавання образів і, зокрема, методи кластерного аналізу. При цьому кожну деталь можна розглядати як точку в l-вимірному просторі її ознак (властивостей). Тоді кожному класу буде відповідати певна група точок в цьому просторі.

У порівнянні з математико-статичними методами, кластерний аналіз не накладає умов на тип даних об’єктів, він дозволяє досліджувати різноманіття вихідних даних довільної природи [28]. Загальна схема методу кластеризації представлена на рис. 2.1.

Перший етап полягає в підготовці даних для кластерного аналізу. У більшості випадків, дані описують у вигляді таблиць, де стовпчик є одним з атрибутів, а рядок об’єктом даних.

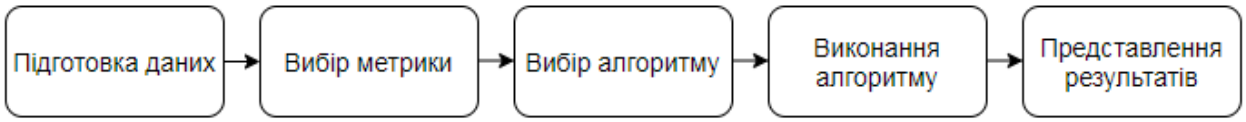

Рисунок 2.1 – Загальна схема методу кластеризації

На другому етапі відбувається вибір метрики, за допомогою якої визначається схожість об’єктів. Метрику підбирають для кожного конкретного типу даних індивідуально. Результати кластеризації можуть істотно відрізнятися при використанні різних вимірів близькості.

На наступному етапі вибирається алгоритм, за допомогою якого групуються об’єкти. Вибір алгоритму є складним завданням, так як отриманий результат багато в чому залежить від нього. Найчастіше доводиться використовувати кілька алгоритмів, тобто комбінувати їх для отримання більш точного результату.

На четвертому етапі відбувається реалізація обраного алгоритму (або декількох алгоритмів). Результатом даного етапу є групування об’єктів по кластерам.

П’ятий етап передбачає подання отриманого угруповання в зручному для інтерпретації вигляді. Представлення результатів кластеризації покликане допомогти найбільш точно інтерпретувати результати виконання алгоритму [29].

В даний час існує досить багато алгоритмів кластерного аналізу. Прийнято всі алгоритми поділяти на дві категорії: ієрархічні та неієрархічні. Розподіл алгоритмів обумовлюється вихідними даними.

Всі методи кластеризації працюють з даними у вигляді векторів у багатовимірному просторі. Кожен вектор визначається значеннями декількох напрямків, а напрямки і є відомі характеристики.

2.1.1 Алгоритм k-means

Найбільш поширений серед неієрархічних методів алгоритм k-means, також званий швидким кластерним аналізом. На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо кількості кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш ймовірну кількість кластерів [30].

Метод k-means використовується для кластеризації даних на основі алгоритму розбиття векторного простору на заздалегідь визначену кількість кластерів *k* таким чином, щоб члени одного кластеру були найбільш однорідними. Процес роботи алгоритму наведений на рисунку 2.1.

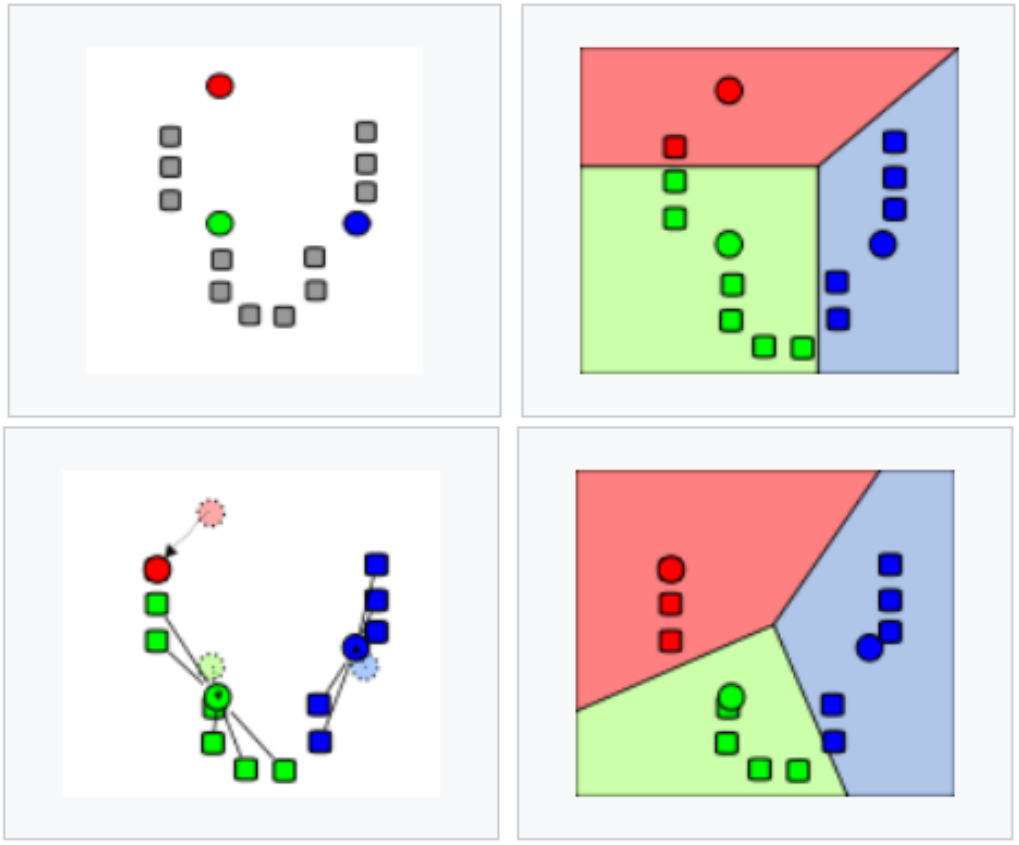

Рисунок 2.1 – Етапи кластеризації алгоритмом k-means

Алгоритм являє собою ітераційну процедуру, в якій виконуються наступні кроки.

Крок 1. Обирається кількість кластерів *k*.

Крок 2. З вихідної множини даних випадковим чином вибираються *k* спостережень, які будуть служити початковими центрами кластерів.

Крок 3. Для кожного спостереження вихідної множини визначається найближчий до неї центр кластера (відстані вимірюються в метриці Евкліда). При цьому об’єкти, «притягнуті» певним центром, утворюють початкові кластери.

Крок 4. Обчислюються центроїди – центри тяжкості кластерів. Кожен центроїд – це вектор, елементи якого являють собою середні значення відповідних ознак, обчислені за всіма об’єктами кластера.

Крок 5. Центр кластера зміщується в його центроїд, після чого центр ваги стає центром нового кластера.

Крок 6. 3-й і 4-й кроки ітеративно повторюються.

Очевидно, що на кожній ітерації відбувається зміна меж кластерів і зміщення їх центрів. У результаті мінімізується відстань між елементами всередині кластерів і збільшуються міжкластерні відстані.

Зупинка алгоритму відбувається тоді, коли кордони кластерів і розташування центроїдів не перестануть змінюватися від ітерації до ітерації. Тобто на кожній ітерації в кожному кластері буде залишатися один і той же набір спостережень. На практиці цей алгоритм зазвичай знаходить набір стабільних кластерів за кілька десятків ітерацій.

Переваги алгоритму k-means:

– простота використання;

– швидкість роботи;

– зрозумілість і прозорість алгоритму.

Недоліки алгоритму k-means:

– не гарантується досягнення глобального мінімуму сумарного квадратичного відхилення, а тільки одного з локальних мінімумів;

– результат залежить від вибору вихідних центрів кластерів, їх оптимальний вибір невідомий;

– кількість кластерів треба знати заздалегідь.

2.1.2 Алгоритм DBSCAN

DBSCAN – алгоритм кластеризації, заснований на щільності даних. Алгоритм був запропонований для вирішення проблеми розбиття даних на кластери довільної форми. Більшість алгоритмів, які виконують розбиття, створюють кластери за формою близькі до сферичних, так як мінімізують відстань до центру кластера. DBSCAN здатний розпізнати кластери різної форми [31].

Ідея, покладена в основу алгоритму, полягає в тому, що всередині кожного кластера спостерігається типова щільність точок (об'єктів), яка помітно вище, ніж щільність зовні кластера, а також щільність в областях з шумом нижче щільності будь-якого з кластерів. Для кожної точки кластера її сусідство заданого радіуса має містити не менше певної кількості точок, це число точок задається граничним значенням.

Вхідні дані алгоритму: множина об'єктів X, для яких задана метрична функція відстані ρ, а також ε – максимальна відстань між сусідніми об'єктами, і $MinPts$ –мінімальна кількість сусідніх об'єктів, необхідних для утворення кластеру.

Об'єкт $p \epsilon X$ називається ядровим, якщо в ε-околі точки p знаходяться $MinPts$ об'єктів (включаючи сам об'єкт p). Ці об'єкти називаються безпосередньо досяжними з p. Об'єкт $q \epsilon X$ називається досяжним з p, якщо існує шлях $p_1, p_2, \ldots, p_n$, де $p_1 = p$ і $p_n = q$, а кожен об`єкт p_{i+1} безпосередньо досяжний з p_i. Таким чином всі об'єкти в шляху, крім об'єкта q, повинні бути ядровими (рис. 2.2).

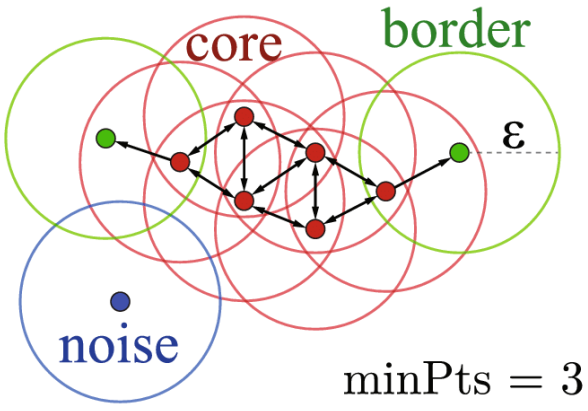

Рисунок 2.2 – Кластеризація алгоритмом DBSCAN

Викидами (або шумом) називаються всі об’єкти, недосяжні ні з одного іншого об’єкта. Кластером є множина ядрових точок, досяжних один з одного, а також граничні неядрові точки, які досяжні з будь-якої ядрового точки кластера.

Алгоритм DBSCAN складається з наступних кроків.

Крок 1. Обирається необроблений об’єкт p.

Крок 2. Об’єкт p позначається як оброблений.

Крок 3. Знаходяться сусідні об’єкти в ε-околі об’єкта p.

Крок 4. Порівнюється кількість сусідніх об’єктів з $MinPts$, визначається, чи є p ядровим об’єктом:

– якщо об’єкт p є ядровим, то створюється новий кластер і запускається пошук в ширину з даного об’єкта по інших невідвіданих об’єктах, знаходячи всі об’єкти кластера;

– якщо об’єкт p не є ядровим, то він відзначається як викид (або шум).

Крок 5. Якщо присутні необроблені об’єкти, то повертаємося до кроку 1.

Переваги алгоритму DBSCAN:

– не вимагає вказувати кількість кластерів;

– може визначати кластери довільної форми;

– може визначати шум і стійкий до викидів.

Недоліки алгоритму DBSCAN:

– недетермінований: граничні точки, досяжні з більш, ніж одного кластера, можуть бути частиною будь-якого з них в залежності від порядку оброблюваних даних;

– необхідно підбирати значення параметрів ε і $MinPts$;

– не може виділяти кластери, що мають різну щільність, так як параметри алгоритму не можуть бути вибрані окремо для кожного кластера.

2.1.3 Алгоритм OPTICS

Впорядкування точок для виявлення кластерної структури (Ordering points to identify the clustering structure, OPTICS) – це алгоритм знаходження кластерів у просторових даних на основі щільності.

Основна ідея алгоритму схожа на DBSCAN, але алгоритм призначений для позбавлення від одного з головних недоліків алгоритму DBSCAN – проблеми виявлення змістовних кластерів в даних, що мають різні щільності. Щоб це

зробити, точки бази даних лінійно упорядковуються так, що просторово близькі точки стають сусідніми в упорядкуванні. Крім того, для кожної точки запам'ятовується спеціальна відстань, що представляє щільність, яку слід прийняти для кластера, щоб точки належали одному кластеру [32].

OPTICS дозволяє вирішити задачу кластеризації в умовах, коли кластери мають не тільки різну форму, але і різну щільність розподілу даних у кожному класі. OPTICS, крім основних понять і визначень, використовуваних в DBSCAN, вводить додаткові характеристики для кожного спостереження такі, як внутрішня відстань (core distance) і відстань досяжності (reachability distance). OPTICS структурно еквівалентний DBSCAN, має розширені функціональні можливості, проте з обчислювальної точки зору значно складніший і повільніший прототипу.

Для роботи алгоритму OPTICS (як і DBSCAN) потрібно задати два параметри – ε (описує максимальну відстань або радіус, який приймається до уваги) і $MinPts$ (описує число точок, необхідних для утворення кластера).

Алгоритм OPTICS дозволяє впорядкувати вихідну множину та спростити процес кластеризації. Відповідно до нього будується діаграма досяжності, завдяки якій з'являється можливість при фіксованому значенні $MinPts$ обробити не тільки задане значення ε, але і всі $\varepsilon^* < \varepsilon$. Точка x є ядровою точкою, якщо щонайменше $MinPts$ точок знаходяться в її ε -околиці $N_\varepsilon(x)$ (рис. 2.3).

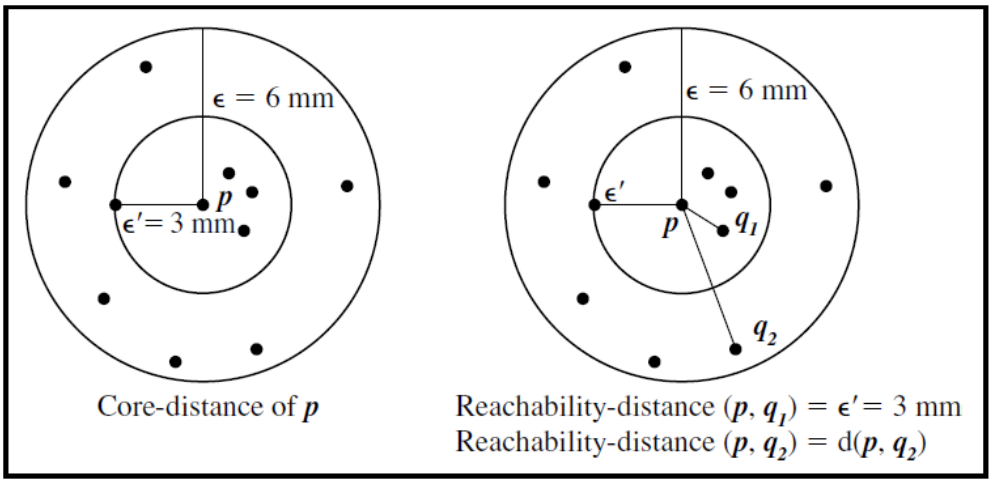

Рисунок 2.3 – Кластеризація алгоритмом OPTICS

Для упорядкування множини X для кожного його елемента обчислюється два параметра – ядерна відстань (core distance, CD) і найменша з відстаней досяжності (reachability distance, RD):

$$CD(x) = \begin{cases} +\infty, & \text{якщо } |N_\varepsilon(x)| < MinPts, \\ MinPts\_dist(x), & \text{інакше;} \end{cases}$$

$$RD(x,y) = \begin{cases} +\infty, & \text{якщо } |N_\varepsilon(y)| < MinPts, \\ \max\{CD(y), D(x,y)\}, & \text{інакше,} \end{cases}$$

де $MinPts\_dist(x)$ – відстань від точки x до її $MinPts$-го сусіда.

Простіше кажучи, ядерна відстань – це найменше значення ε^*, при якому x є ядром, а відстань досяжності – значення ε^*, при якому x стає безпосередньо досяжна з y. Відповідно, найменша з відстаней досяжності для x – це значення ε^*, при якому x стає досяжною хоча б з одного ядра (перестає бути шумом). Залежно від комбінації цих параметрів, при фіксованому значенні ε^* точка x може бути внутрішньою ($CD(x) \le \varepsilon^*$) або граничною точкою ($CD(x) > \varepsilon^*$), ($RD(x) \le \varepsilon^*$) точкою кластера , а також бути шумом ($CD(x) > \varepsilon^*, RD(x) > \varepsilon^*$).

Завдяки сортуванню за допомогою додаткових полів, класифікація вибірки алгоритмом DBSCAN з параметрами $\varepsilon^* \le \varepsilon$, $MinPts$ зводиться до послідовного перебору впорядкованої вибірки і присвоєнню кожному її елементу номера відповідного класу (обрізання діаграми досяжності на потрібному рівні і виділенню на ній кластерів).

Алгоритм OPTICS можна представити у вигляді наступних кроків.

Крок 1. Вибирається довільний необроблений об’єкт x у якості поточного об’єкта.

Крок 2. Знаходяться сусідні об’єкти ε-околиці об’єкта x, визначається ядерна відстань, відстань досяжності встановлюється значенням «не визначена».

Крок 3. Об’єкт x позначається як оброблений.

Крок 4. Об’єкт x поміщається в упорядкований список:

– якщо x не є ядровим об’єктом, відбувається перехід до наступного об’єкта;

– якщо x – ядровий об’єкт, то для кожного об’єкта y в ε-околиці x оновлюється його відстань досяжності від x і вставляється y у впорядкований список, якщо y ще не оброблено.

Крок 5. Якщо присутні необроблені об’єкти, то повертаємося до кроку 1.

Переваги алгоритму OPTICS:

– не вимагає вказувати кількість кластерів;

– дозволяє виділяти кластери, що мають різну щільність;

– результат роботи алгоритму OPTICS значно менше залежить від параметра ε, ніж результат DBSCAN.

Недоліки алгоритму OPTICS:

– необхідність підбору значень параметрів ε і $MinPts$;

– при виборі занадто маленького значення ε структура класів може залишитися непоміченою (аж до віднесення всієї вибірки до шуму), а при занадто великому значенні ε обчислювальна складність алгоритму значно зростає;

– алгоритм OPTICS повільніший ніж DBSCAN.

2.1.4 Алгоритм Affinity propagation

Метод поширення близькості (Affinity propagation, AP) являє собою алгоритм кластеризації на основі концепції «передачі повідомлень» між точками даних. На відміну від алгоритмів кластеризації, таких як k-середніх не вимагає кількості кластерів, які будуть визначені або оцінені перед запуском алгоритму [33].

AP працює на основі оцінок схожості між парами точок даних і одночасно розглядає всі точки даних як потенційні центри кластерів, звані лідерами. Існує два види повідомлень, якими обмінюються точки даних. Перший називається – відповідальність, а другий – доступність.

Відповідальність (responsibility, таблиця R з елементами $r(i,k)$) відповідає за те, наскільки i хоче бачити k своїм лідером. Відповідальність покладається кожною точкою на кандидата в лідери групи. Доступність (availability, таблиця A з елементами $a(i,k)$) – є відповідь від потенційного лідера k, наскільки добре готова представляти інтереси i (рис. 2.4).

AP виділяє серед об’єктів «лідерів» (exemplar) і формує кластери навколо них. На вибір точки у якості лідера впливають три параметри: схожість s, відповідальність r та доступність a.

На початку, кожна точка i покладає відповідальність на всі точки, включаючи й саму себе. Найближча (найбільш схожа) точка задає розподіл відповідальності для всіх інших точок. Відповідальність, покладена на найближчу точку також залежить від розташування другої найближчої. Якщо в радіусі досяжності є кілька більш-менш схожих на неї кандидатів, на тих буде

покладена приблизно однакова відповідальність, що виступає свого роду обмежувачем – якщо якась точка занадто сильно схожа на всі інші, їй нічого не залишається, крім як сподіватися тільки на себе.

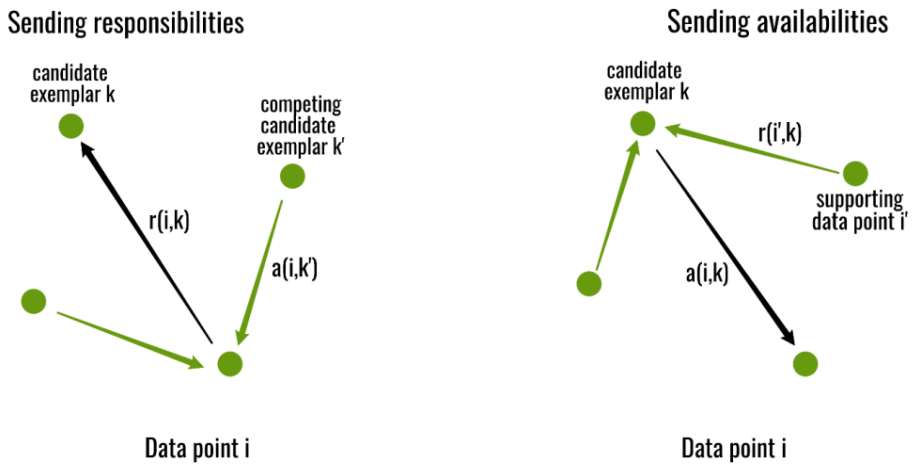

Рисунок 2.4 – Кластеризація алгоритмом AP

Вхідні дані алгоритму: метрична функція схожості s, яка кількісно визначає схожість між двома точками. Алгоритм роботи складається з таких кроків.

Крок 1. Матриці r (матриця відповідальності) і a (матриця доступності) ініціюються нулями.

Крок 2. Даний крок виконується T разів:

а) Оновити матрицю відповідальності r наступним чином:

$$r[i,j] = (1-\lambda)\rho[i,j] + \lambda r[i,j];$$

$$\rho[i,j] = \begin{cases} s[i,j] - max_{k \neq j}\{a[i,k] + s[i,k]\}, \text{якщо } (i \neq j), \\ s[i,j] - max_{k \neq j}\{s[i,k]\}, \text{якщо } (i = j), \end{cases}$$

де λ – коефіцієнт загасання, введений з метою уникнення численних коливань;

$\rho[i,j]$ – поширювана відповідальність;

$s[i,k]$ – функція схожості;

б) оновити матрицю доступності a наступним чином:

$$a[i,j] = (1-\lambda)\gamma[i,j] + \lambda a[i,j];$$

$$\gamma[i,j] = \begin{cases} \min\left(0, r[j,j] + \sum_{k \neq i,j} \max(0, r[k,j])\right), \text{якщо } (i \neq j), \\ \sum_{k \neq i,j} \max(0, r[k,j]), \text{якщо } (i = j), \end{cases}$$

де $\gamma[i,j]$ – доступність, що розповсюджується;

λ – коефіцієнт загасання, введений з метою уникнення численних коливань.

в) визначаються лідери. Лідерами вважаються точки, що задовільняють умові:

$$r(i,i) + a(i,i) > 0,$$

де $r(i,i)$ – відповідальність i-ої точки;

$a(i,i)$ – доступність i-ої точки.

Крок 4. Обчислити зразок для кожної точки: знайти лідера, з яким точка максимально схожа.

Переваги методу Affinity propagation:

– автоматичний вибір кількості кластерів;

– можливість роботи з різними типами даних, включаючи асиметричні дані;

– усі точки розглядаються як потенційні лідери.

Недоліком методу Affinity propagation є те, що відносно швидко зростає час його роботи зі зростанням вибірки даних.

2.2 Визначення типового технологічного процесу

Для вирішення поставленої задачі необхідно перебрати всі можливі комбінації обладнання, щоб знайти оптимальну серед них. Нажаль, це може привести до комбінаторного вибуху, що спричинить значні витрати часу і розрахункових ресурсів. Для подолання даної проблеми, пропонується розбити ТП на групи (кластери). Для кожного кластера буде визначено типовий технологічний процес як центр ваги відповідного кластера.

У табл. 2.1 наведені значення вихідних даних для розв’язання задачі. Вказані коди обладнання для 100 ТП та 3 показника Мі 1 – складність, Мі 2 – обсяг, Мі 3 – характер типу ТП [34].

Таблиця 2.1 – Приклад набору нормованих значень характеристик ТП

Код	Мі 1	Мі 2	Мі 3	Код	Мі 1	Мі 2	Мі 3	Код	Мі 1	Мі 2	Мі 3
0	56,68	11,24	43	34	27,00	157,10	37	67	28,93	111,81	32
1	91,64	577,01	16	35	51,64	95,23	43	68	69,87	276,88	12
2	57,82	741,38	12	36	51,98	470,37	43	69	70,24	823,22	7
3	21,67	105,92	39	37	11,47	846,50	18	70	60,18	910,41	15
4	84,54	683,07	5	38	68,87	899,75	23	71	19,36	971,88	28
5	34,31	756,25	2	39	92,15	408,31	43	72	29,41	345,17	21
6	96,29	856,30	42	40	31,99	683,68	12	73	14,44	656,82	49
7	6,85	687,46	44	41	40,82	610,66	42	74	81,15	892,34	4
8	2,64	35,81	38	42	14,18	795,03	48	75	11,61	497,03	38
9	45,74	934,58	21	43	54,65	248,06	29	76	32,01	599,76	31
10	33,36	772,24	28	44	2,04	833,06	14	77	81,17	340,14	4
11	69,93	82,12	19	45	37,61	959,44	47	78	74,76	168,52	22
12	43,26	87,15	11	46	44,09	164,29	22	79	99,05	571,72	38
13	44,35	616,32	39	47	68,94	363,66	33	80	9,38	850,19	37
14	51,07	823,83	40	48	0,54	939,93	45	81	88,66	504,68	6
15	84,33	588,86	35	49	69,09	821,33	6	82	66,99	29,71	28
16	37,85	261,00	9	50	11,05	756,65	27	83	92,60	946,53	2
17	61,68	118,49	31	51	25,69	148,38	47	84	17,99	579,63	10
18	19,69	879,50	7	52	89,88	728,44	30	85	22,83	502,80	5
19	93,03	89,92	5	53	54,42	402,23	43	86	44,46	277,85	12
20	91,59	142,81	11	54	52,78	862,34	36	87	74,06	203,96	21
21	60,20	199,62	30	55	44,81	508,98	16	88	54,04	790,01	41
22	93,15	16,65	27	56	73,51	214,35	32	89	94,92	698,78	33
23	23,51	422,59	22	57	21,09	79,77	48	90	24,97	805,18	47
24	29,13	835,90	48	58	27,14	678,47	15	91	56,72	246,00	42
25	63,34	920,24	26	59	64,67	66,48	5	92	53,53	398,30	8
26	53,70	793,81	19	60	56,05	695,44	21	93	12,94	449,98	18
27	91,88	482,47	30	61	30,06	241,05	20	94	51,68	986,18	37
28	33,01	890,39	21	62	60,99	74,82	26	95	48,36	239,94	41
29	82,59	914,45	27	63	8,83	494,08	8	96	31,39	381,54	9
30	38,94	547,30	38	64	16,04	530,56	27	97	45,55	560,03	12
31	13,67	176,10	9	65	80,28	920,89	12	98	80,68	439,00	42
32	59,39	142,40	46	66	30,27	19,89	46	99	67,37	988,06	44
33	60,65	846,53	7								

Під складністю Мі 1 розуміються сукупні витрати часу трудовими ресурсами, інтелектуальна місткість тощо, для використання конкретного обладнання у конкретному ТП.

Під обсягом Мі 2 розуміються сукупні витрати матеріальних та фінансових ресурсів.

Під характером типу ТП Мі 3 розуміється сукупність характеристик, які однозначно ідентифікують можливість використовувати обладнання у виробництві [34].

На рис.2.2 наведено графічне відображення вихідних даних, які наведені у табл. 2.1.

Результати кластеризації цієї множини об’єктів з використанням алгоритмів k-means, DBSCAN, OPTICS, Affinity propagation наведені на рис. 2.3-2.6.

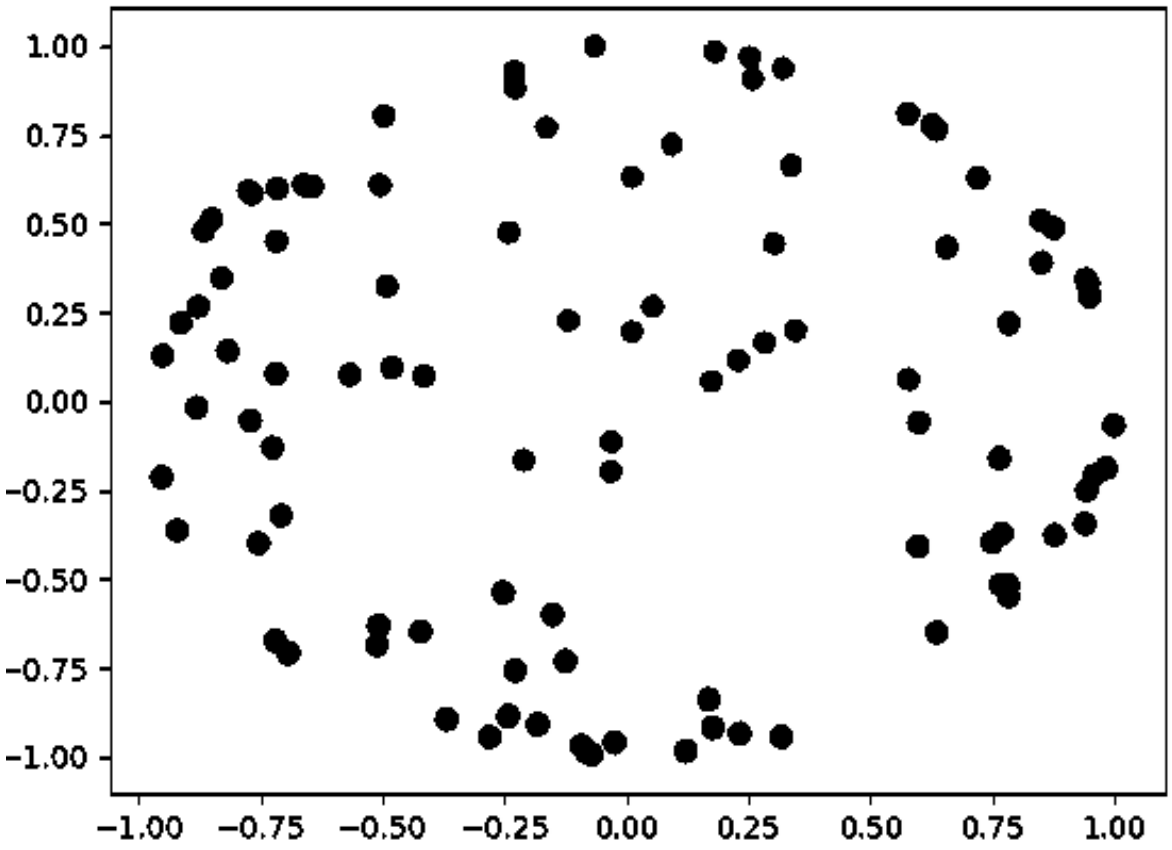

Рисунок 2.2 – Приклад вихідної інформації для розподілу ТП на кластери

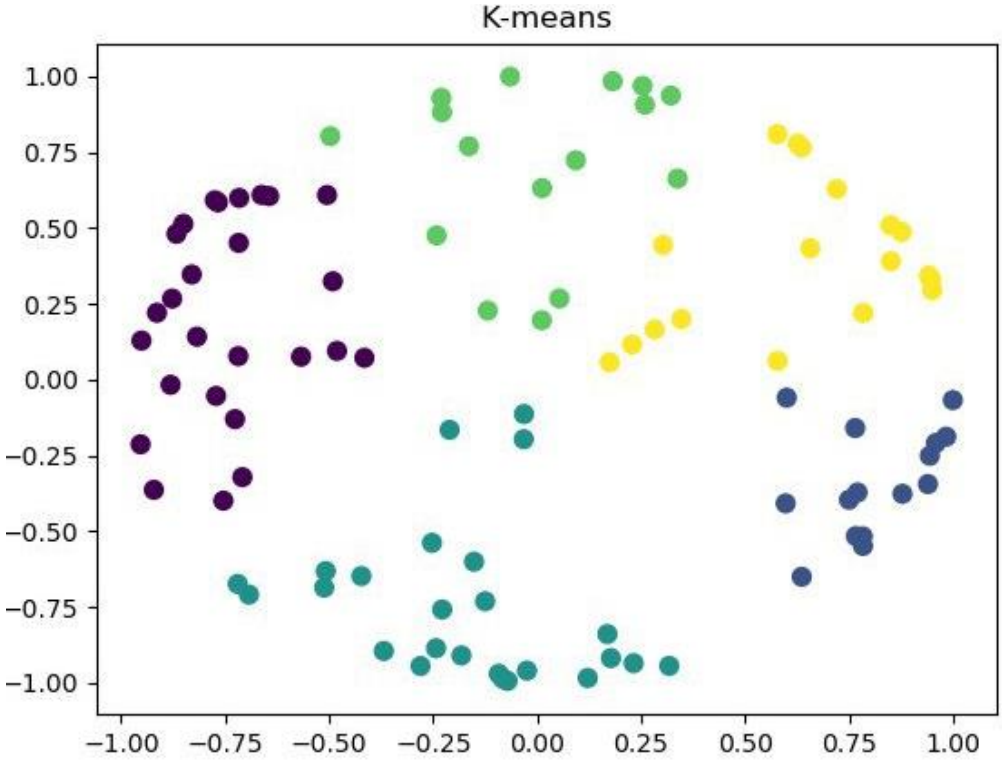

Рисунок 2.3 – Приклад розподілу ТП на кластери (алгоритм k-means)

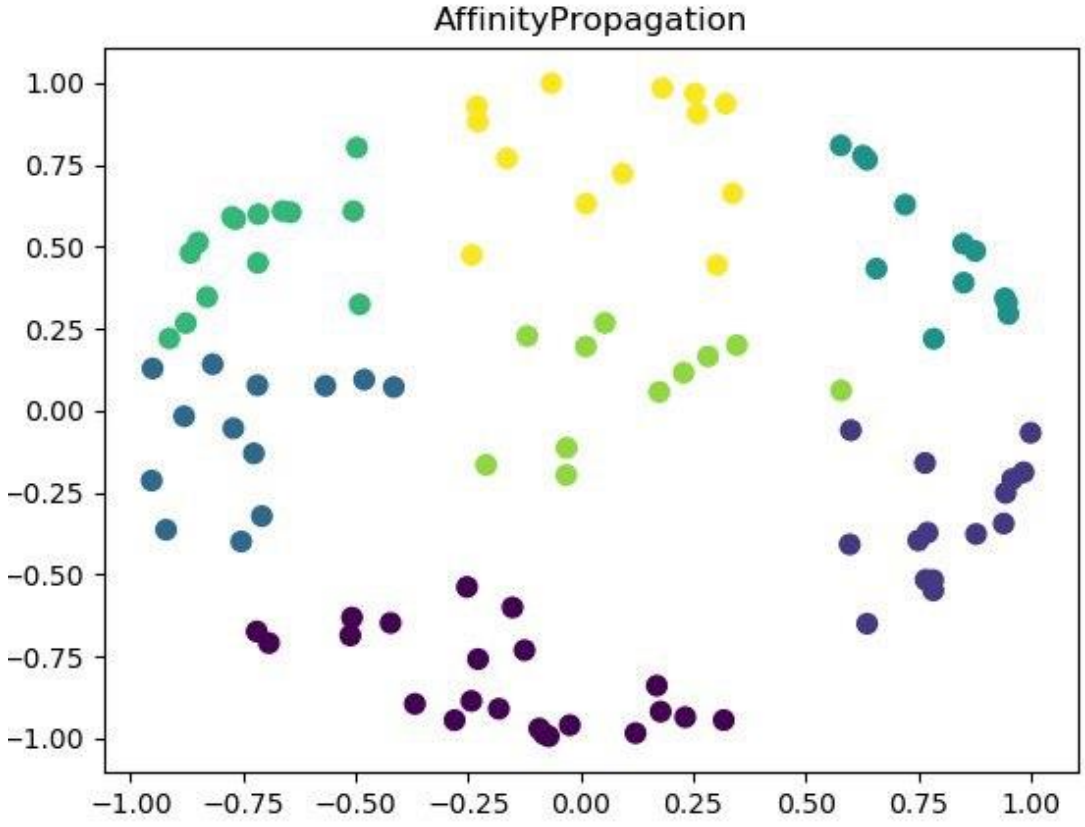

Рисунок 2.4 – Приклад розподілу ТП на кластери (алгоритм AP)

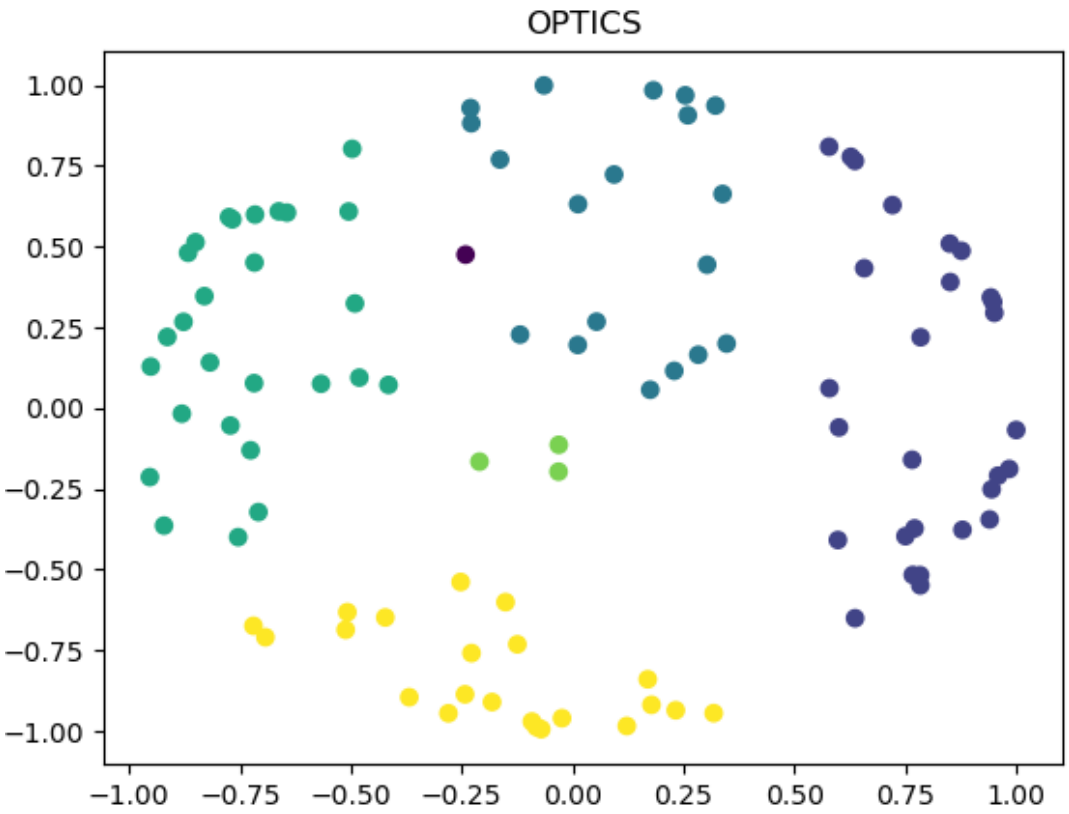

Рисунок 2.5 – Приклад розподілу ТП на кластери (алгоритм OPTICS)

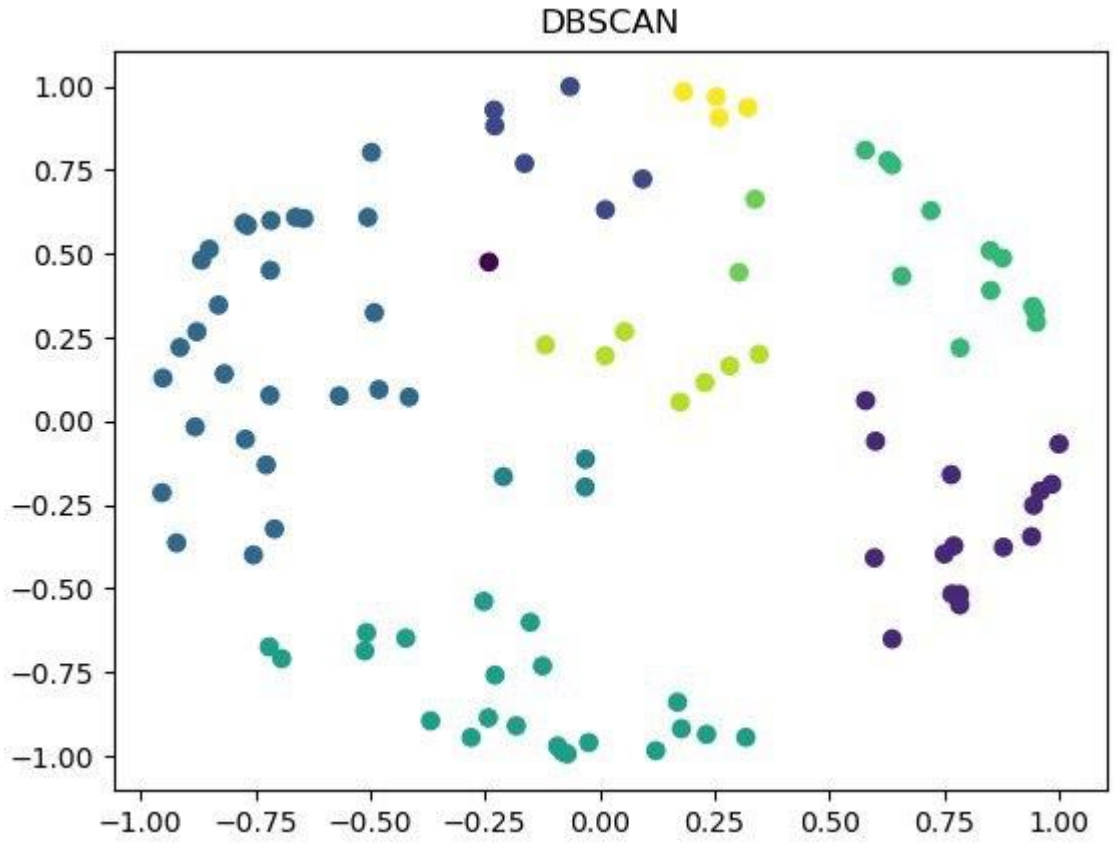

Рисунок 2.6 – Приклад розподілу ТП на кластери (алгоритм DBSCAN)

Після проведення кластеризації та отримання типових ТП з'явиться можливість співвіднести варіант ТП, що проектується, з одним з типових ТП (завдання класифікації), і провести подальшу добірку обладнання з метою мінімізації витрат зі збереженням продуктивності.

2.3 Алгоритми класифікації ТП

Для класифікації ТП відносно типових ТП запропоновано використати наступні алгоритми: метод опорних векторів (SVM) та k-найближчих сусідів (KNN).

2.3.1 Метод опорних векторів (SVM)

Основна ідея класифікатора на опорних векторах полягає в тому, щоб будувати розділяючу поверхню з використанням тільки невеликої підмножини точок, що лежать в зоні, критичної для поділу, тоді як інші спостереження навчальної вибірки, що класифікуються, поза цією зоною ігноруються (точніше, є «резервуаром» для оптимізаційного алгоритму) [35].

Розглядається задача навчання на прецедентах $\langle X, Y, y^*, X^l \rangle$, де X – простір об`єктів, Y – множина відповідей, $y^*: X \to Y$ – цільова залежність, значення якої відомі тільки на об'єктах навчальної вибірки $X^l = (x_i, y_i)_{i=1}^l$, , $y_i = y^*(x_i)$. Потрібно побудувати алгоритм $a: X \to Y$, який апроксимує цільову залежність на всьому просторі X.

Розглянемо задачу класифікації на два непересічних класи, в якій об'єкти описуються n -мірними векторами $X = R^n$, $Y = \{-1, +1\}$.

Побудуємо лінійний пороговий класифікатор:

$$a(x) = sign\left(\sum_{j=1}^{n} w_j x^j - w_0\right) = sign(\langle w, x \rangle - w_0), \qquad (2.1)$$

де $x = (x^1, x^2, \dots, x^n)$ – опис ознак об'єкта x;

$w = (w^1, w^2, \dots, w^n) \in R^n$ і $w_0 \in R$ – параметри алгоритму.

Рівняння $\langle w, x \rangle = w_0$ описує гіперплощину, що розділяє класи в просторі R^n.

Припустимо, що вибірка лінійно роздільна, тобто існують такі значення параметрів w, w_0, при яких функціонал числа помилок $Q(w, w_0) = \sum_{i=1}^{l}[y_i(\langle w, x_i\rangle - w_0) < 0]$ приймає нульове значення. Але тоді розділяюча гіперплощина не єдина, оскільки існують й інші положення розділяючої гіперплощини, що реалізують те ж саме розбиття вибірки.

Ідея методу полягає в тому, щоб розумним чином розпорядитися цією свободою вибору. Вимагатимемо, щоб розділяюча гіперплощина максимально далеко знаходилася від найближчих до неї точок обох класів. Метод опорних векторів вибирає ту гіперплощину, яка максимізує відступ між класами.

Відступ (margin) – характеристика, яка оцінює, наскільки об'єкт «занурений» у свій клас, наскільки типовим представником класу він є. Чим менше значення відступу, тим ближче об'єкт підходить до кордону класів і тим вище стає ймовірність помилки. Відступ негативний тоді і тільки тоді, коли алгоритм припускається помилки на об'єкті.

Параметри лінійного порогового класифікатора визначені з точністю до нормування: алгоритм $a(x)$ не зміниться, якщо w, w_0 одночасно помножити на одну і ту ж додатню константу. Зручно вибрати цю константу таким чином, щоб для всіх прикордонних (найближчих до розділяючої гіперплощини) об'єктів x_i з X^l виконувалися умови $\langle w, x_i\rangle - w_0 = y_i$.

Зробити це можливо, оскільки при оптимальному положенні розділяючої гіперплощини всі прикордонні об'єкти знаходяться від неї на однаковій відстані. Решта об'єктів знаходяться далі. Таким чином, для всіх $x_i \epsilon X^l$

$$\langle w, x_i\rangle - w_0 \begin{cases} \leq -1, & \text{якщо } y_i = -1, \\ \geq 1, & \text{якщо } y_i = +1. \end{cases} \qquad (2.2)$$

Умова $-1 < \langle w, x_i\rangle - w_0 < 1$ задає смугу, що розділяє класи. Жодна з точок навчальної вибірки не може лежати всередині цієї смуги. Межами смуги служать дві паралельні гіперплощини з направляючим вектором w. Точки, найближчі до розділяючої гіперплощини, лежать в точності на межах смуги. При цьому сама розділяюча гіперплощина проходить рівно по середині смуги.

Щоб розділяюча гіперплощина якнайдалі знаходилася від точок вибірки, ширина смуги повинна бути максимальною. Шириною площини є:

$$\langle (x_- - x_+), \frac{w}{\|w\|} \rangle = \frac{\langle w, x_+ \rangle - \langle w, x_- \rangle}{\|w\|} = \frac{(w_0+1)-(w_0-1)}{\|w\|} = \frac{2}{\|w\|},$$

де x_- і x_+ – дві довільні точки класів –1 і +1 відповідно, що лежать на кордоні смуги.

Ширина смуги максимальна, коли норма вектора w мінімальна.

У разі, коли вибірка лінійно роздільна, досить прості геометричні міркування приводять до наступної задачі: потрібно знайти такі значення параметрів w і w_0, при яких норма вектора w мінімальна за умови (2.2).

Лінійно роздільна вибірка. Побудова оптимальної розділяючої гіперплощини зводиться до мінімізації квадратичної форми при l обмеженнях-нерівностях виду (2.2) щодо $n+1$ змінних w, w_0:

$$\begin{cases} \langle w, w \rangle \to min, \\ y_i(\langle w, x_i \rangle - w_0) \geq 1, \quad i = 1, \dots, l \end{cases} \tag{2.3}$$

По теоремі Куна-Таккера ця задача еквівалентна двоїстій задачі пошуку сідлової точки функції Лагранжа:

$$\begin{cases} L(w, w_0; \lambda) = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{l} \lambda_i (y_i(\langle w, x_i \rangle - w_0) - 1) \to \min_{w, w_0} max_\lambda, \\ \lambda_i \geq 0, \quad i = 1, \dots, l, \\ \lambda_i = 0, \quad \text{або} \quad \langle w, x_i \rangle - w_0 = y_i, \quad i = 1, \dots, l, \end{cases}$$

де $\lambda = (\lambda_i, \dots, \lambda_l)$ – вектор двоїстих змінних.

Необхідною умовою сідлової точки є рівність нулю похідних лагранжіан. Звідси випливають два співвідношення:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} \lambda_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{l} \lambda_i y_i x_i \,, \tag{2.4}$$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{l} \lambda_i y_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{l} \lambda_i y_i = 0 \,. \tag{2.5}$$

З (2.4) випливає, що шуканий вектор ваг w є лінійною комбінацією векторів навчальної вибірки, причому тільки тих, для яких $\lambda_i \neq 0$. Згідно з умовою доповнюючої жорсткості на цих векторах x_i обмеження-нерівності

перетворюються в рівності: $\langle w, x_i \rangle - w_0 = y_i$, отже, ці вектори знаходяться на кордоні розділяючої смуги. Всі інші вектори знаходяться далі від кордону, для них $\lambda_i = 0$, і вони не беруть участі в сумі (2.4). Алгоритм (2.1) не змінився б, якби цих векторів взагалі не було в навчальній вибірці.

Якщо $\lambda_i > 0$ і $\langle w, x_i \rangle - w_0 = y_i$, то об'єкт навчальної вибірки x_i називається опорним вектором (support vector).

Підставляючи (2.4) і (2.5) назад в лагранжіан, отримаємо еквівалентну задачу квадратичного програмування, що містить тільки двоїсті змінні:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{l} \lambda_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \big(\langle x_i, x_j \rangle\big) \to min_{\lambda}, \\ \lambda_i \geq 0, \quad i = 1, \dots, l, \\ \sum_{i=1}^{l} \lambda_i y_i = 0. \end{cases} \tag{2.6}$$

Тут мінімізується квадратичний функціонал, що має невід'ємну визначену квадратичну форму, отже, опуклий. Область, яка визначається обмеженнями-нерівностями та однією рівністю, також опукла. Отже, дана задача має єдине рішення.

Вектор w обчислюється за формулою (2.4). Для визначення порогу w_0 досить взяти довільний опорний вектор x_i і виразити w_0 з рівності $w_0 = = \langle w, x_i \rangle - y_i$. На практиці для підвищення чисельної стійкості рекомендується брати в якості w_0 середнє по всім опорним векторах, або медіану:

$$w_0 = med\{\langle w, x_i \rangle - y_i : \ \lambda_i > 0, \ \ i = 1, \dots, l\}. \tag{2.7}$$

В результаті алгоритм класифікації може бути записаний в наступному вигляді:

$$a(x) = sign\big(\sum_{i=1}^{l} \lambda_i y_i \langle x_i, x \rangle - w_0\big). \tag{2.8}$$

Реально підсумовування йде не по всій вибірці, а тільки по опорних векторах, для яких $\lambda_i \neq 0$. Саме це властивість розрідженості (sparsity) відрізняє SVM від інших лінійних роздільників – дискримінанту Фішера, логістичної регресії і одношарового персептрона.

Лінійно нероздільна вибірка. Щоб узагальнити SVM на випадок лінійної нероздільності, дозволимо алгоритму допускати помилки на навчальних об’єктах, але при цьому постараємося, щоб помилок було менше. Введемо набір додаткових змінних $\xi_i \geq 0$, що характеризують величину помилки на об’єктах $x_i,\ i = 1, \dots, l$. Візьмемо за відправну точку задачу (2.3). Пом’якшимо в ній обмеження-нерівності і одночасно введемо в функціонал, що мінімізується штраф за сумарну помилку:

$$\begin{cases} \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{l}\xi_i \to min_{w,w_0,\xi}, \\ y_i(\langle w, x_i\rangle - w_0) \geq 1 - \xi_i, \quad i = 1, \dots, l, \\ \xi_i \geq 0, \quad i = 1, \dots, l. \end{cases} \tag{2.9}$$

У разі $Y = \{-1, +1\}$ відступом (margin) об’єкта x_i від кордону класів називається величина $m_i = y_i(\langle w, x_i\rangle - w_0)$.

Алгоритм припускається помилки на об’єкті x_i тоді і тільки тоді, коли відступ m_i негативний. Якщо $m_i \in (-1, +1)$, то об'єкт x_i потрапляє всередину розділяючої смуги. Якщо $m_i > 1$, то об’єкт x_i класифікується правильно, і знаходиться на деякому віддаленні від розділяючої смуги. Запишемо функціонал числа помилок алгоритму a на вибірці X^l в термінах відступів: $Q(a, X^l) = \sum_{i=1}^{l}[m_i < 0]$.

Замінимо в цьому функціоналі порогову функцію втрат кусочно-лінійною верхньою оцінкою: $[m_i < 0] \leq (1 - m_i)_+$. Сенс цієї заміни в тому, щоб зробити функцію втрат чутливою до величини помилки, а також ввести штраф за наближення об’єкта до кордону класів.

Крім того, додамо до функціоналу Q штрафний доданок $\tau\|w\|^2$. Відповідно до принципу регуляризації некоректно поставлених задач, така добавка означає, що серед усіх векторів w, що мінімізують функціонал Q, найкращими є вектори з мінімальною нормою. Регуляризація часто застосовується для налаштування лінійних моделей класифікації. При наявності шумових і (або) залежних ознак вона підвищує стійкість алгоритму по відношенню до складу вибірки і його узагальнюючу здатність.

З урахуванням обох модифікацій функціонал якості набуває вигляду:

$$(a, X^l) = \sum_{i=1}^{l}(1 - m_i)_+ + \tau\|w\|^2 \to min_{w,w_0}. \tag{2.10}$$

Задача мінімізації даного функціоналу еквівалентна оптимізаційній задачі з обмеженнями (2.9), якщо взяти параметр регуляризації $\tau = \frac{1}{2C}$. Таким чином, принцип оптимальної розділяючої гіперплощини (або максимізації ширини розділяючої смуги) збігається з принципом регуляризації по Тихонову.

Позитивна константа C (або τ) є керуючим параметром методу та дозволяє знаходити компроміс між максимізацією розділяючої смуги та мінімізацією сумарної помилки.

Запишемо функцію Лагранжа для задачі (2.9):

$$L(w, w_0, \xi; \lambda, \eta) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \lambda_i (y_i(\langle w, x_i\rangle - w_0) - 1) - \sum_{i=1}^{l} \xi_i(\lambda_i + \eta_i - C),$$

де $\eta = (\eta_1, \dots, \eta_l)$ – вектор змінних, двоїстих до змінних $\xi = (\xi_1, \dots, \xi_l)$.

Умови Куна-Таккера зводять задачу до пошуку сідлової точки функції Лагранжа:

$$\begin{cases} L(w, w_0, \xi; \lambda, \eta) \to \min\limits_{w, w_0, \xi} max_{\lambda, \eta}, \\ \xi_i \geq 0, \lambda_i \geq 0, \eta_i \geq 0, \quad i = 1, \dots, l, \\ \lambda_i = 0, \ \text{або} \ \ y_i(\langle w, x_i\rangle - w_0) = 1 - \xi_i, \quad i = 1, \dots, l, \\ \eta_i = 0 \ \ \text{або} \ \ \xi_i = 0, \ \ i = 1, \dots, l. \end{cases}$$

В останніх двох рядках записані умови доповнюваної нежорсткості. Необхідною умовою сідлової точки є рівність нулю похідних Лагражіана. Звідси виходять три співвідношення:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} \lambda_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{l} \lambda_i y_i x_{i,} \tag{2.11}$$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{l} \lambda_i y_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{l} \lambda_i y_i = 0, \tag{2.12}$$

$$\frac{\partial L}{\partial \xi_i} = -\lambda_i - \eta_i + C = 0 \quad \Rightarrow \quad \lambda_i + \eta_i = C, \quad i = 1, \dots, l. \tag{2.13}$$

Перші два співвідношення такі ж, як і в лінійно роздільному випадку. Із третього співвідношення та нерівності $\eta_i > 0$ з'являється обмеження $\lambda_i \leq C$.

Звідси, та з умов доповнюваної нежорсткості випливає, що можливі тільки три допустимих поєднання значень змінних ξ_i і відступів m_i.

Відповідно, всі об'єкти x_i, $i = 1, \dots, l$ діляться на наступні три типи:

а) $\lambda_i = 0;\ \eta_i = C;\ \xi_i = 0;\ m_i > 0$. Об'єкт x_i класифікується правильно і знаходиться далеко від розділяючої смуги. Такі об'єкти будемо називати периферійними;

б) $0 < \lambda_i < C;\ 0 < \eta_i < C;\ \xi_i = 0;\ m_i = 0$. Об'єкт x_i класифікується правильно і лежить в точності на кордоні розділяючої смуги. Такі об'єкти будемо називати опорними;

в) $\lambda_i = \mathrm{C};\ \eta_i = 0;\ \xi_i > 0;\ m_i < 1$. Об'єкт x_i або лежить всередині розділяючої смуги, але класифікується правильно ($0 < \xi_i < 1;\ 0 < m_i < 1$), або потрапляє на кордон класів ($\xi_i = 1;\ m_i = 0$), або взагалі відноситься до чужого класу ($\xi_i > 1;\ m_i < 0$). У всіх цих випадках об'єкт x_i будемо називати порушником.

У силу співвідношення (2.13) в лагранжіані обнуляються всі члени, що містять змінні ξ_i і η_i, і він приймає той же вид, що і в разі лінійної роздільності. Параметри розділяє поверхні w і w_0, відповідно до формул (2.11) і (2.12), також виражаються тільки через подвійні змінні λ_i. Таким чином, задача знову зводиться до квадратичного програмування щодо подвійних змінних λ_i. Єдина відмінність від лінійно розділимого випадку полягає в появі обмеження зверху $\lambda_i \leq \mathrm{C}$:

$$\begin{gathered} -L(\lambda) = -\sum_{i=1}^{l} \lambda_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \langle x_i, x_j \rangle \to min_{\lambda}, \\ 0 \leq \lambda_i \leq \mathrm{C}, \quad i = 1, \dots, l, \\ \sum_{i=1}^{l} \lambda_i y_i = 0. \end{gathered} \tag{2.14}$$

На практиці для побудови SVM вирішують саме цю задачу, а не (2.6), так як гарантувати лінійну роздільність вибірки в загальному випадку не представляється можливим. Цей варіант алгоритму називають SVM з м'яким зазором (soft-margin SVM), тоді як в лінійно роздільному випадку говорять про SVM з жорстким зазором (hard-margin SVM).

Для алгоритму класифікації зберігається формула (2.8), з тією лише різницею, що тепер ненульовими λ_i володіють не тільки опорні об'єкти, але і об'єкти порушники. У певному сенсі це недолік SVM, оскільки порушниками

часто виявляються шумові викиди, і побудоване на них вирішальне правило, по суті, спирається на шум.

Константу C зазвичай вибирають за критерієм змінного контролю. Це трудомісткий спосіб, так як задачу доводиться вирішувати заново при кожному значенні C.

Якщо є підстави вважати, що вибірка майже лінійно роздільна, і лише об'єкти-викиди класифікуються невірно, то можна застосувати фільтрацію викидів. Спочатку завдання вирішується при деякому C, і з вибірки видаляється невелика частка об'єктів, що мають найбільшу величину помилки ξ_i. Після цього завдання вирішується заново з усіченою вибіркою.

Існує ще один підхід для вирішення проблеми лінійної нероздільності. Це перехід від початкового простору ознакових описів об'єктів X до нового простору H за допомогою деякого перетворення $\psi: X \to H$. Якщо простір H має досить високу розмірність, то можна сподіватися, що в ньому вибірка виявиться лінійно нероздільною. Простір H називають спрямним.

Якщо припустити, що ознаковими описами об'єктів є вектори $\psi(x_i)$, а не вектори x_i, то побудова SVM проводиться так само, як і раніше. Єдина відмінність полягає в тому, що скалярний добуток $\langle x_i, x' \rangle$ в просторі X всюди замінюється скалярним добутком $\langle \psi(x), \psi(x^1) \rangle$ у просторі H.

Звідси випливає вимога: простір H має бути наділений скалярним добутком, зокрема, підійде будь-який простір.

Функція $K: X \times X \to R$ називається ядром (kernel function), якщо вона подана у вигляді $K(x, x') = \langle \psi(x), \psi(x^1) \rangle$ при деякому відображенні $\psi: X \to H$, де H – простір зі скалярним добутком.

Постановка задачі (2.14), як і сам алгоритм класифікації (2.8), залежать тільки від скалярних добутків об'єктів, але не від самих ознакових описів. Це означає, що скалярний добуток $\langle x, x' \rangle$ можна формально замінити ядром $K(x, x')$. Оскільки ядро в загальному випадку нелінійне, така заміна приводить до істотного розширення множини реалізованих алгоритмів $a: X \to Y$.

Більш того, можна взагалі не будувати спрямний простір H в явному вигляді, і замість підбору відображення ψ займатися безпосередньо підбором ядра.

Можна відмовитися від ознакових описів об'єктів. У багатьох практичних завданнях об'єкти спочатку задаються інформацією про їх попарні

взаємовідносини, наприклад, щодо схожості. Якщо ця інформація допускає подання у вигляді двомісної функції $K(x, x')$, що задовольняє аксіомам скалярного добутку, то задача може вирішуватися шляхом SVM.

Функція $K(x, x')$ є ядром тоді і тільки тоді, коли вона симетрична, $K(x, x') = K(x', x)$ і невід'ємно визначена: $\iint K(x, x')g(x)g(x')dxdx' \geq 0$ для будь-якої функції $g: X \to R$.

Перевірка невід'ємності визначеності функції в практичних ситуаціях може виявитися справою нетривіальною. Часто обмежуються перебором кінцевого числа функцій, про які відомо, що вони є ядрами. Серед них вибирається найкраща, як правило, за критерієм змінного контролю.

Переваги SVM:

– завдання опуклого квадратичного програмування добре вивчена і має єдине рішення;

– принцип оптимальної розділяючої гіперплощини призводить до максимізації ширини розділяючої смуги, а отже, до більш впевненої класифікації.

Недоліки SVM:

– нестійкість до шуму: викиди у вихідних даних стають опорними об'єктами-порушниками і безпосередньо впливають на побудову розділяючої гіперплощини.

– не описані загальні методи побудови ядер і спрямних просторів, найбільш придатних для конкретного завдання;

– відсутній відбір ознак.

2.3.2 Алгоритм k-найближчих сусідів (k-NN)

В алгоритмі k-найближчих сусідів (k-nearest neighbors method, k-NN) [36] передбачається, що вже є певна кількість об'єктів з точною класифікацією (тобто для кожного з них точно відомо якому класу він належить). Потрібно виробити правило, що дозволяє віднести новий об'єкт до одного з можливих класів (самі класи відомі заздалегідь).

В основі k-NN лежить таке правило: об'єкт належить до того класу, до якого належить більшість його найближчих сусідів. Під «сусідами» тут розуміються об'єкти, близькі до досліджуваного в тому чи іншому сенсі.

На першому кроці алгоритму слід задати число k – кількість найближчих сусідів. Якщо прийняти $k = 1$, то алгоритм втратить узагальнюючу здатність (здатність видавати правильний результат для даних, що раніше в алгоритмі не зустрічалися) так як новому об'єкту буде присвоєно клас близький до нього. Якщо встановити занадто велике значення, то багато локальних особливостей буде не виявлено.

На другому кроці знаходяться k об'єктів з мінімальним відстанню до вектора ознак нового об'єкта (пошук сусідів). Нехай x, y, z – вектори ознак порівнюваних об'єктів, тоді функція для розрахунку відстані повинна відповідати наступним правилам:

а) $d\,(x, y) \geq 0, d\,(x, y) = 0$ тоді і тільки тоді, коли $x = y$;

б) $d\,(x, y) = d\,(y, x)$;

в) $d\,(x, z) \leq d\,(x, y) + d\,(y, z)$ за умови, що точки x, y, z не лежать на одній прямій.

Для впорядкованих значень атрибутів знаходиться евклідова відстань:

$$D_E = \sqrt{\sum_{i}^{n} (x_i - y_i)^2},$$

де n – кількість атрибутів.

Для строкових змінних, які не можуть бути впорядковані, може бути застосована функція відмінності, яка задається наступним чином:

$$dd(x, y) = \begin{cases} 0, & x = y, \\ 1, & x \neq y. \end{cases}$$

Часто перед розрахунком відстані виникає необхідність у нормалізації даних. Різні атрибути можуть мати різний діапазон представлених значень у вибірці. У такому випадку значення дистанції можуть сильно залежати від атрибутів з великими діапазонами. Саме тому необхідно використовувати нормалізацію. При кластерному аналізі є два основних способи нормалізації даних: мінімаксна і нормалізація за допомогою стандартного відхилення.

Мінімаксна нормалізація: $x^* = \frac{x - \min(x)}{\max(x) - \min(x)}$.

Нормалізація за допомогою стандартного відхилення: $x^* = \frac{x - x_{cp}}{\sigma_x}$, де σ_x – стандартне відхилення; x_{cp} – середнє значення.

При знаходженні відстані іноді виникає необхідність врахувати значимість атрибутів. Значимість визначається експертом або аналітиком суб'єктивно, покладаючись на власний досвід. У такому випадку при знаходженні відстані кожен i-ий квадрат різниці в сумі множиться на коефіцієнт z_i.

Після того, як знайдені об'єкти, найбільш схожі на новий, необхідно вирішити, як вони впливають на клас нового об'єкту. Для цього використовується функція поєднання (combination function). Одним з основних варіантів такої функції є просте незважене голосування (simple unweighted voting).

Просте незважене голосування. Спочатку, задавши число k, визначається скільки об'єктів буде мати право голосу при визначенні класу. Потім виявляються об'єкти, відстань від яких до нової виявилося мінімальним. Далі необхідно приступити до голосування.

Відстань від кожного об'єкту при голосуванні більше не грає ролі. Всі мають рівні права у визначенні класу. Кожна вільна позиція голосує за клас, до якого належить. Нового об'єкту присвоюється клас, який набрав найбільшу кількість голосів:

$$y_a(a, X, k) = arg\ max_{y \in Y} \sum_{i=1}^{k} y_a^{(i)} = y,$$

де a – новий об'єкт;

X – навчальна вибірка;

Y – множина класів;

$y_a^{(i)}$ – клас i-го сусіда a;

k – кількість сусідів.

При використанні простого незваженого голосування виникає проблема, коли декілька класів набрали рівну кількість голосів. Цю проблему дозволяє вирішити зважене голосування (weighted voting).

Зважене голосування. При зваженому голосуванні враховується також і відстань до нового об'єкту. Чим менше відстань, тим більш значущий внесок вносить голос. Голоси за клас знаходяться за наступною формулою:

$$y_a(a, X, k) = arg\ max_{y \in Y} \sum_{i=1}^{k} \frac{1}{d^2(a, b)} (y_a^{(i)} = y),$$

де a – новий об’єкт;

X – навчальна вибірка;

Y – множина класів;

$y_a^{(i)}$ – клас i-го сусіда a;

$d(a, b)$ – відстань від відомого об’єкта b до нового a;

k – кількість сусідів.

Новий запис співвідноситься з класом, який набрали найбільшу кількість голосів. При цьому ймовірність того, що кілька класів наберуть однакові голоси, набагато нижче. При $k = 1$, новому запису присвоюється клас найближчого сусіда.

Робота алгоритму здійснюється у декілька кроків.

а) вибрати значення k, k – будь-яке ціле число;

б) для кожної точки в тестових даних необхідно:

1) розрахувати відстань між даними випробувань і кожним рядком даних тренування.

2) відсортувати відстані в порядку їх зростання.

3) обрати верхні k рядків з відсортованого масиву.

4) призначити клас контрольній точці на основі класу рядків, що найбільш часто зустрічаються.

Переваги методу k-NN:

– програмна реалізація алгоритму відносно проста;

– можливість модифікації алгоритму;

– алгоритм стійкий до аномальних викидів;

– можливість інтерпретації результатів роботи алгоритму.

Недоліки методу k-NN:

– набір даних, який використовується для алгоритму, повинен бути репрезентативним;

– необхідність зберігати навчальну вибірку цілком;

– великі затрати продуктивності, оскільки необхідно обчислити відстані між кожним екземпляром і всіма пробними екземплярами.

2.4 Знаходження оптимальної структури технологічного процесу

Загальну схему алгоритму можна представити наступним чином (рис.2.7).

Вихідними даними є множина технологічних процесів, які розбиваються на кластери з використанням алгоритмів кластеризації (k-means, DBSCAN, OPTICS, Affinity propagation) за принципом схожості. Для кожного кластера знаходиться типовий технологічний процес, який представляє собою центр мас кластера. Коли отримуємо новий ТП, то відносимо його до одного з кластерів за допомогою алгоритмів класифікації (Support Vector Machines, K-Nearest Neighbors).

Після того, як ТП було віднесено до якогось з кластерів виконується наступна процедура: для кожного етапу обирається тільки те обладнання, яке відповідає заданому типу та має продуктивність не нижче необхідної. При перегляді всіх структур обладнання, які відносяться до кластеру, вибирається те, що має мінімальну вартість. Таким чином формується ефективне рішення.

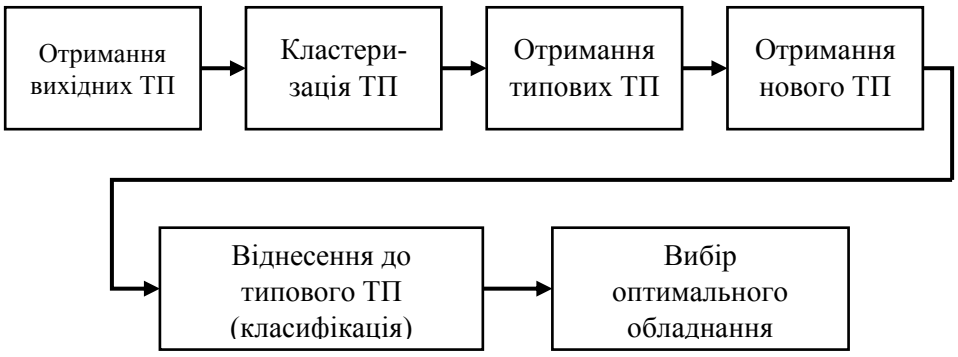

Рисунок 2.7 – Загальна схема методу оптимізації виробничого ТП

Розглянемо приклад, в якому за базовий визначено ТП, який включає 2 фази (операції). На першій фазі потрібно використовувати обладнання типу 1, продуктивність якого повинна бути не менше ніж 5.1 (од./год). На другій фазі потрібно використовувати обладнання типу 3, продуктивність якого повинна бути не менше ніж 6.1 (од./год).

При класифікації ТП був віднесений до кластеру 3, частина його елементів приведена у табл. 2.2. У таблицю включені лише ті елементи які мають перший

тип обладнання, бо для демонстрації роботи не має сенсу вказувати всі елементи кластеру. Те обладнання, яке відповідає вимогам виділено сірим кольором.

Таблиця 2.2 – Елементи кластеру 3

Процес	ТП4	ТП5	ТП9	ТП13	ТП18	ТП21	ТП46	ТП57	ТП81
Тип обладнання	1	1	1	1	1	1	1	1	1
Продуктивність, од./год.	3,6	7,2	7,1	3,6	5,8	8,7	3,7	6,8	10,2
Вартість, тис. у.о.	12,4	10,6	48,8	44,5	56,9	12,4	27,1	54,4	72,2

Частина елементів 3 кластеру, яка необхідна для вибору обладнання для другої фази, приведена у табл. 2.3. Те обладнання, яке відповідає вимогам виділено сірим кольором.

Таблиця 2.3 – Частина елементів кластеру 3

Процес	ТП20	ТП27	ТП35	ТП41	ТП47	ТП51	ТП52	ТП71	ТП82
Тип обладнання	3	3	3	3	3	3	3	3	3
Продуктивність, од./год.	4,6	2,2	5,1	8,6	5,4	7,1	4,7	5,1	6,3
Вартість, тис. у.о.	74,4	53,6	24,7	41,7	36,9	27,1	22,2	87,4	41,1

Отже, за результатами прикладу, новий технологічний процес було віднесено до 3 кластеру, тому пошук обладнання здійснювався в його рамках. Найменша вартість обладнання для першої фази становить 10.6 (тис. у.о.), а сама мінімальна вартість для другої фази становить 27.1 (тис. у.о.). Сумарна наведена вартість обладнання двох фаз складає 37.7 (тис. у.о.).

3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Вибір середовища розробки

Для вирішення обраної задачі можуть використовуватися різні середовища розробки. У роботі було вирішено використовувати середовище розробки Spyder, оскільки вона найкраще підходить для наукових розрахунків на мові Python. Spyder забезпечує легкість використання функціональних можливостей та легковажність програмної частини.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника та читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій [37].

Python підтримує структурне, об’єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване програмування. Основні архітектурні риси – динамічна типізація, автоматичне керування пам’яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об’єднуватися в пакети.

Відмінні характеристики мови Python:

– низький поріг входження;

– мінімалістична мова, з невеликою кількістю конструкцій;

– лаконічний код;

– підходить для створення програм-обгорток, підтримується імпорт С-бібліотек;

– існує велика кількість реалізацій: CPython (основна реалізація); Jython (реалізація для JVM); IronPython (CLR); PyPy;

– хороша підтримка математичних обчислень (бібліотеки NumPy, SciPy);

– використовується для обробки природних мов (NLTK);

– велика кількість розвинених web-фреймворків (Django, TurboGear, CherryPy, Flask).

Для багатьох основна перевага мови Python полягає в легкості читання, ясності та більш високій якості, що відрізняють його від інших інструментів у

мовах програмування. Програмний код на мові Python читається легше, а значить, багаторазове його використання і обслуговування виконується набагато простіше, ніж використання програмного коду на інших мовах сценаріїв. Одноманітність оформлення програмного коду на мові Python полегшує його розуміння навіть для тих, хто не брав участі в його створенні. Крім того, Python підтримує найсучасніші механізми багаторазового використання програмного коду, яким є об'єктно-орієнтоване програмування.

Python у багато разів підвищує продуктивність праці розробника. Обсяг програмного коду на мові Python зазвичай становить третину або навіть п'яту частину еквівалентного програмного коду на мові C ++ або Java. Це означає менший обсяг введення з клавіатури, менша кількість часу на налагодження і менший обсяг трудовитрат на супровід. Крім того, програми на мові Python запускаються відразу ж, минаючи тривалі етапи компіляції і зв'язування, необхідні в деяких інших мовах програмування, що ще більше збільшує продуктивність праці програміста.

Велика частина програм на мові Python виконується без змін на всіх основних платформах. Перенесення програмного коду з операційної системи Linux в Windows зазвичай полягає в простому копіюванні файлів програм з однієї машини на іншу. Більш того, Python надає масу можливостей по створенню переносяться графічних інтерфейсів, програм доступу до баз даних, веб-додатків і багатьох інших типів програм.

У складі Python поставляється велика кількість функціональних можливостей, відомих як стандартна бібліотека. Крім того, Python допускає розширення як за рахунок ваших власних бібліотек, так і за рахунок бібліотек, створених сторонніми розробниками. З числа сторонніх розробок можна назвати інструменти створення веб-сайтів, програмування математичних обчислень, доступ до послідовного порту, розробку ігрових програм і багато іншого. Наприклад, розширення NumPy позиціонується як вільний і більш потужний еквівалент системи програмування математичних обчислень Mathlab.

Сценарії Python легко можуть взаємодіяти з іншими частинами програми завдяки різним механізмам інтеграції. Ця інтеграція дозволяє використовувати Python для настройки і розширення функціональних можливостей програмних продуктів.

З точки зору функціональних можливостей Python можна назвати гібридом. Його інструментальні засоби вкладаються в діапазон між

традиційними мовами сценаріїв (такими як Tcl, Scheme і Perl) і мовами розробки програмних систем (такими як C, C ++ і Java).

Python сам стежить за типами об’єктів, що використовуються в програмі. Умові Python взагалі відсутні поняття типу і необхідність оголошення змінних. Так як програмний код на мові Python не обмежений рамками типів даних, він автоматично може обробляти цілий діапазон об’єктів.

Python автоматично розподіляє пам’ять під об’єкти і звільняє її, коли об'єкти стають непотрібними.

Для створення великих систем Python надає такі можливості, як модулі, класи і виключення. Вони дозволяють розбити систему на складові, застосовувати ООП для створення програмного коду багаторазового користування, обробляти події та помилки.

Python надає найбільш типові структури даних, такі як списки, словники і рядки, у вигляді особливостей, властивих самій мові програмування. Ці типи відрізняються високою гнучкістю та зручністю. Наприклад, вбудовані об’єкти можуть розширюватися і стискуватися по необхідності, можуть комбінуватися один з одним для представлення даних зі складною структурою.

Для роботи з усіма цими типами об’єктів в складі Python є потужні і стандартні засоби, включаючи такі операції, як конкатенація (об’єднання колекцій), отримання зрізів (витяг частини колекції), сортування, відображення і багато іншого.

Для виконання більш вузьких завдань до складу Python також входить велика колекція бібліотечних інструментів, які підтримують практично все, що може знадобитися – від пошуку з використанням регулярних виразів до роботи в мережі.

Python – це відкритий програмний продукт і тому розробники можуть створювати свої попередньо скомпільовані інструменти підтримки завдань, вирішити які внутрішніми засобами неможливо.

Коли запускається програма, Python спочатку компілює вихідний текст в формат, відомий під назвою байт-код. Трансляція в байт-код проводиться для підвищення швидкості – байт-код виконується набагато швидше, ніж вихідні інструкції в текстовому файлі.

Інтерпретатор зберігає байт-код для прискорення запуску програм. При наступному запуску програми, Python завантажить файл .pyc і мине етап – за умови, що вихідний текст програми не змінювався з моменту останньої

компіляції. Щоб визначити, чи потрібно виконувати перекомпіляцію, Python автоматично порівняє час останньої зміни файлу з вихідним текстом і файлу з байт-кодом. Якщо вихідний текст зберігався на диск після компіляції, наступного разу його запуску інтерпретатор автоматично виконає повторну компіляцію програми.

Якщо інтерпретатор виявиться не в змозі записати файл з байт-кодом на диск, програма від цього не постраждає, байт-код буде згенеровано в пам’яті і зникне після закінчення програми.

Як тільки програма буде скомпільована в байт-код, він передається механізму під назвою віртуальна машина Python (PVM). PVM – це механізм часу виконання, вона завжди присутня в складі системи Python і це програмний компонент, який виконує сценарії.

Оригінальний текст, який вводиться програмістом, транслюється в байт-код, який потім виповнюється віртуальною машиною Python. Оригінальний текст автоматично компілюється та потім інтерпретується.

Завдяки простоті і гнучкості мови Python, він є дуже популярним серед користувачів. Програми на Python розробляються швидше, ніж на компільованих мовах. Мова може представляти інтерес для розробки програм, що використовують складні структури даних.

3.2 Основні функції програмного засобу

Повний текст програмного засобу наведено у додатку Б. Серед реалізованих функцій слід зазначити:

– функція «Пошук оптимальної структури обладнання» (min_value), параметри, які приймаються: labels – приналежність до кластеру; test_item – новий ТП; items – всі ТП; n_cluster – номер кластера; n_stage – кількість етапів

```
def min_value(labels, test_item, items, n_cluster, n_stage):
    min = [[-1] for i in range(2)]
    index = [[-1] for i in range(2)]
    for i in range(n_stage):
        type = test_item.at[0, 'Type'][i]
        power = test_item.at[0, 'Power'][i]
        for j in range(items.index.size):
            if labels[j] == n_cluster[0]:
```

```
                if items.at[j, 'Type'] == type and items.at[j,
    'Power'] >= power:
                    if min[i] != -1 and items.at[j, 'Value'] >
    min[i]:
                        min[i] = items.at[j, 'Value']
                        index[i] = j
                    elif min[i] == -1:
                        min[i] = items.at[j, 'Value']
                        index[i] = j
        return min, index
```

– функція «Повний перебір» (exhaustive_search), параметри, які приймаються: test_item – новий ТП; items – всі ТП; n_stage – кількість етапів

```
def exhaustive_search(test_item, items, n_stage):
    min = [[-1] for i in range(2)]
    index = [[-1] for i in range(2)]
    for i in range(n_stage):
        type = test_item.at[0, 'Type'][i]
        power = test_item.at[0, 'Power'][i]
        for j in range(items.index.size):
            if items.at[j, 'Type'] == type and items.at[j,
    'Power'] >= power:
                if min[i] != -1 and items.at[j, 'Value'] >
    min[i]:
                    min[i] = items.at[j, 'Value']
                    index[i] = j
                elif min[i] == -1:
                    min[i] = items.at[j, 'Value']
                    index[i] = j
    return min, index
```

– функція «Визначення типових ТП» (get_centroids), параметри, які приймаються: labels – приналежність до кластеру; items – всі ТП; n_centroids – кількість центроїдів

```
def get_centroids(labels, items, n_centroids):
    centroids = pd.DataFrame(columns=["Mi1", "Mi2", "Mi3"])
    for i in range(n_centroids + 1):
```

```
            mi = [[0] for i in range(7)]
            count = 0
            for j in range(len(labels)):
                if labels[j] == i:
                    for k in range(7):
                        mi[k] += items.iat[j, k]
                    count += 1
            centroids.loc[i] = {"Mi1": mi[0] / count, "Mi2": mi[1]
      / count, "Mi3": mi[2] / count
    }
        return centroids
```

– функція «Класифікація k-NN» (k_neighbors_classifier), параметри, які приймаються: list_centroids – список типових ТП; test – новий ТП

```
def k_neighbors_classifier(list_centroids, test):
    classifier = KNeighborsClassifier(n_neighbors=3)
    x = list_centroids.iloc[:, :7].values
    y = list_centroids.index.values
    classifier.fit(x, y)
    KNN_prediction = classifier.predict(test)
    return KNN_prediction
```

– функція «Класифікація SVM» (min_value), параметри, які приймаються: list_centroids – список типових ТП; test – новий ТП

```
def svm_classifier(list_centroids, test):
    x = list_centroids.iloc[:, :7].values
    y = list_centroids.index.values
    SVC_model = svm.SVC()
    SVC_model.fit(x, y)
    SVC_prediction = SVC_model.predict(test)
    return SVC_prediction
```

3.3 Аналіз результатів експериментів

Розглянемо множину зі 100 елементів, де кожен елемент – це виробничий ТП. Проведемо кластеризацію алгоритмами: OPTICS, DBSCAN, k-means, AP. Моделювання було проведено 10 разів з різними вихідними даними, усередненні результати кластеризації наведено у табл. 3.1.

Таблиця 3.1 – Результат кластеризації

Алгоритм	OPTICS	DBSCAN	k-means	AP
Час роботи, с	0.16	0,01	0.09	0.04
Кількість кластерів	5	9	5	6

Як видно з табл. 3.1:

– алгоритм OPTICS – отримав саму малу кількість кластерів, але витратив найбільшу кількість часу;

– алгоритм DBSCAN – має дуже малий час роботи, отримав найбільшу кількість кластерів;

– алгоритм k-means – має середній час роботи, отримав саму малу кількість кластерів;

– алгоритм AP – час роботи менше середнього, формує не велику кількість кластерів.

У табл. 3.2 приведено результат моделювання процесу вибору оптимальної структури обладнання. Для кожного алгоритму кластеризації було застосовано алгоритми класифікації k-NN та SVM, також у табл. 3.2 приведені показники повного перебору. Показник «Час роботи» включає час, який витрачається на відповідну класифікацію, та час, якій необхідно витратити на вибір ліпшого рішення.

З табл. 3.2 видно, що алгоритм «Повний перебір» дав оптимальний варіант обладнання ТП загальною вартістю 2036,3 тис. у.о., затративши на це 0,0184 с. Ці показники є умовним еталоном. Самий малий час роботи у алгоритму DBSCAN k-NN. Також слід виділити AP k-NN та SVM. Самі нижчі показники вартості у OPTICS SVM, також слід виділити AP SVM та k-means k-NN, які дали схожі значення.

Таблиця 3.2 – Результат моделювання

Алгоритм кластеризації	Повний перебір	OPTICS		DBSCAN		k-means		AP	
Алгоритм класифікації		k-NN	SVM	k-NN	SVM	k-NN	SVM	k-NN	SVM
Час роботи, с	0,0184	0,0151	0,0178	0,0080	0,0109	0,0171	0,0170	0,0093	0,0099
Загальна вартість, тис. у.о.	2036,3	2791,9	2225,1	2819,3	2819,3	2387,8	2683,9	2402,0	2398,3
Фаза 1									
Код облад-ня	55	35	81	72	72	55	72	72	50
Вартість, тис. у.о.	337,1	458,7	353,1	605,7	605,7	337,1	605,7	605,7	363,4
Фаза 2									
Код облад-ня	62	34	84	31	31	84	77	98	7
Вартість, тис. у.о.	723,5	837,9	793,0	961,2	961,2	793,0	957,5	727,9	791,8
Фаза 3									
Код облад-ня	24	29	80	79	79	41	23	23	5
Вартість, тис. у.о.	442,7	489,6	507,6	667,3	667,3	520,7	497,0	497,0	489,5
Фаза 4									
Код облад-ня	74	30	93	15	15	15	39	93	60
Вартість, тис. у.о.	130,1	585,2	143,8	130,8	130,8	130,8	196,1	143,8	299,3
Фаза 5									
Код облад-ня	67	28	92	52	52	75	92	92	52
Вартість, тис. у.о.	402,9	420,5	427,6	454,3	454,3	606,2	427,6	427,6	454,3

Таблиця 3.3 – Час розв’язання задачі для ТП з різною кількістю фаз

Алгоритм 1	Повний перебір	OPTICS		DBSCAN		k-means		AP	
Алгоритм 2		k-NN	SVM	k-NN	SVM	k-NN	SVM	k-NN	SVM
1 фаза									
Час роботи, с	0,0036	0,0074	0,0114	0,0056	0,0081	0,0066	0,0072	0,0058	0,0063
2 фази									
Час роботи, с	0,0065	0,0089	0,0115	0,0058	0,0082	0,0108	0,0107	0,0068	0,0075
3 фази									
Час роботи, с	0,0121	0,0108	0,0144	0,0068	0,0085	0,0113	0,0110	0,0076	0,0088
4 фази									
Час роботи, с	0,0148	0,0131	0,0169	0,0073	0,0091	0,0132	0,0135	0,0080	0,0092
5 фази									
Час роботи, с	0,0184	0,0151	0,0178	0,0080	0,0109	0,0171	0,0170	0,0093	0,0099

Проведемо аналіз того, як кількість фаз впливає на час роботи алгоритмів. Результати проведення моделювання для різної кількості фаз наведені у табл. 3.3, на її основі побудуємо графіки залежності часу роботи від кількості фаз для повного перебору та різних методів кластеризації та класифікації (рис. 3.1-3.11).

Данні в табл. 3.3 наведені з усередненням, виходячи з 10 експериментів.

Рисунок 3.1 – Графік часової складності для алгоритму «Повного перебору»

Часова складність алгоритму «Повного перебору» складає:

$$t(N) = -0.0002 \cdot N^3 + 0.0012 \cdot N^2 + 0.0009 \cdot N + 0.0015.$$

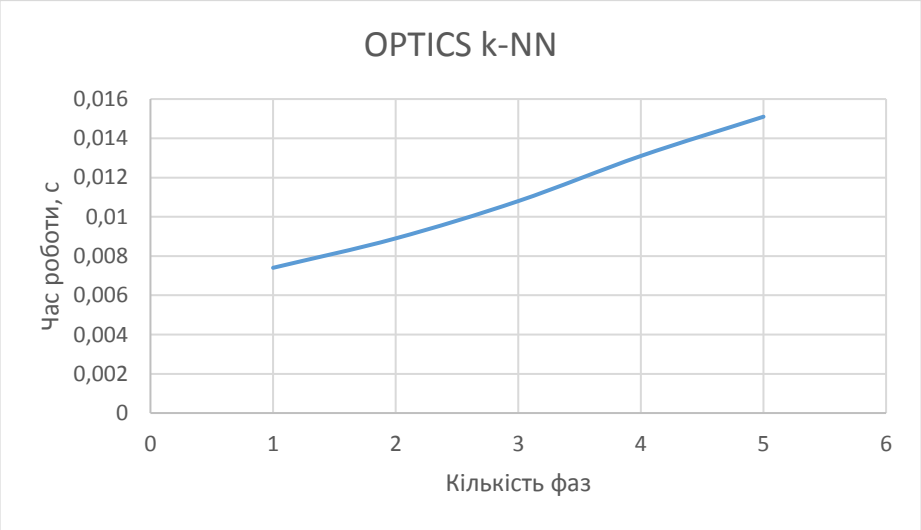

Рисунок 3.2 – Графік часової складності для алгоритму «OPTICS» k-NN

Часова складність алгоритму «OPTICS» k-NN складає:

$$t(N) = 0.002 \cdot N + 0.0052.$$

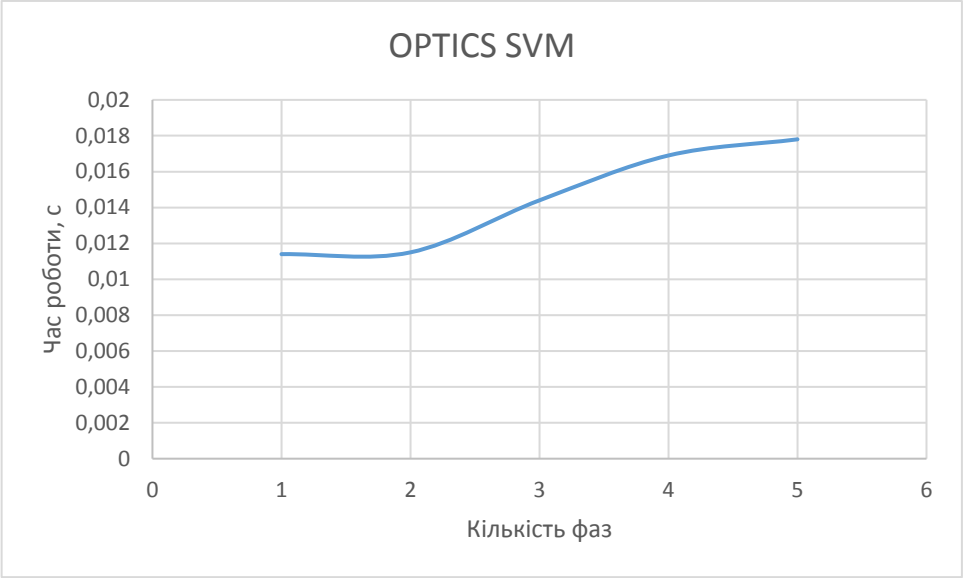

Рисунок 3.3 – Графік часової складності для алгоритму «OPTICS» SVM

Часова складність алгоритму «OPTICS» SVM складає:

$$t(N) = -0.0004 \cdot N^3 + 0.0034 \cdot N^2 - 0.0073 \cdot N + 0.0157.$$

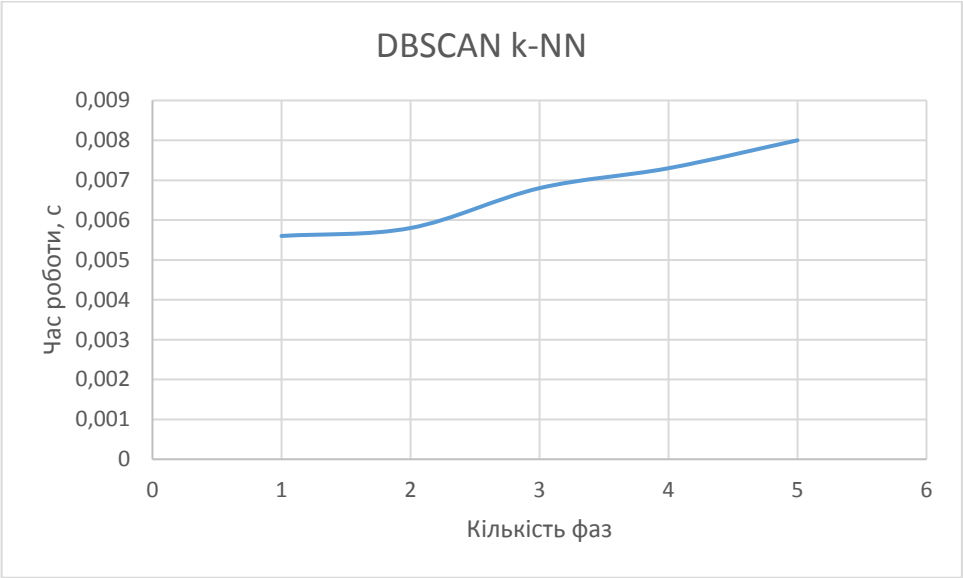

Рисунок 3.4 – Графік часової складності для алгоритму «DBSCAN» k-NN

Часова складність алгоритму «DBSCAN» k-NN складає:

$$t(N) = -5 \cdot 10^{-5} \cdot N^3 + 0.0005 \cdot N^2 - 0.0008 \cdot N + 0.0059.$$

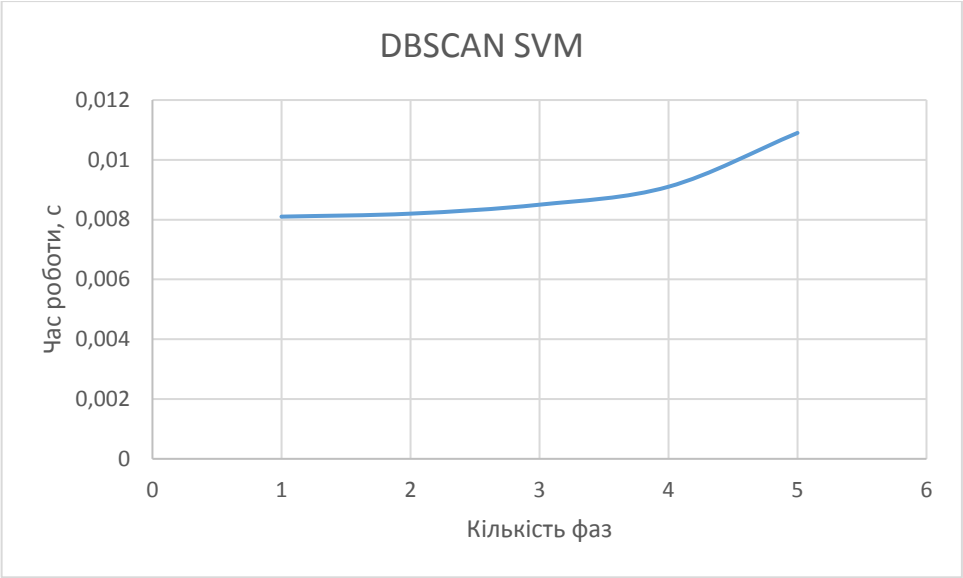

Рисунок 3.5 – Графік часової складності для алгоритму «DBSCAN» SVM

Часова складність алгоритму «DBSCAN» SVM складає:

$$t(N) = 0.0003 \cdot N^2 - 0.0009 \cdot N + 0.0089.$$

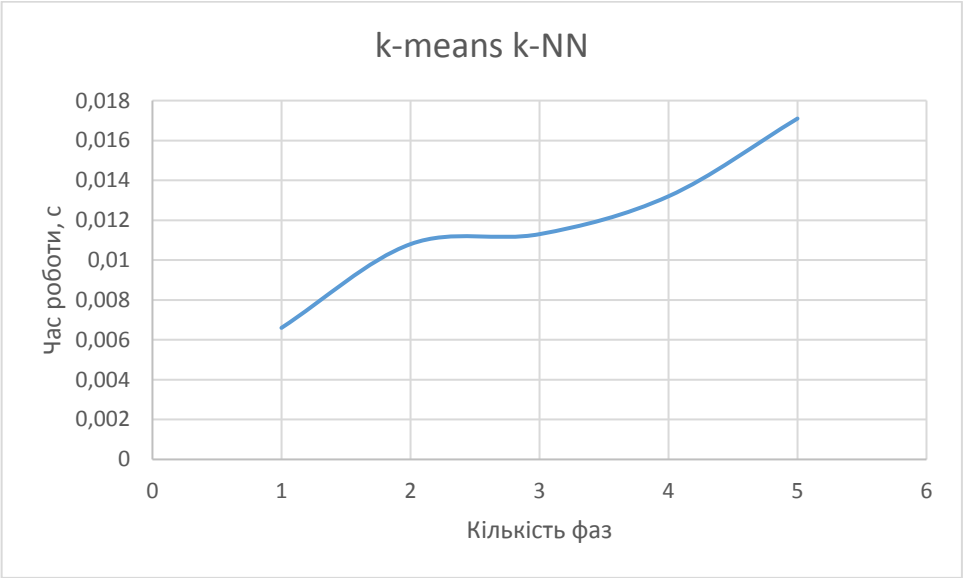

Рисунок 3.6 – Графік часової складності для алгоритму «k-means» k-NN

Часова складність алгоритму «k-means» k-NN складає:

$$t(N) = 0.0005 \cdot N^3 - 0.0042 \cdot N^2 + 0.0132 \cdot N - 0.0028.$$

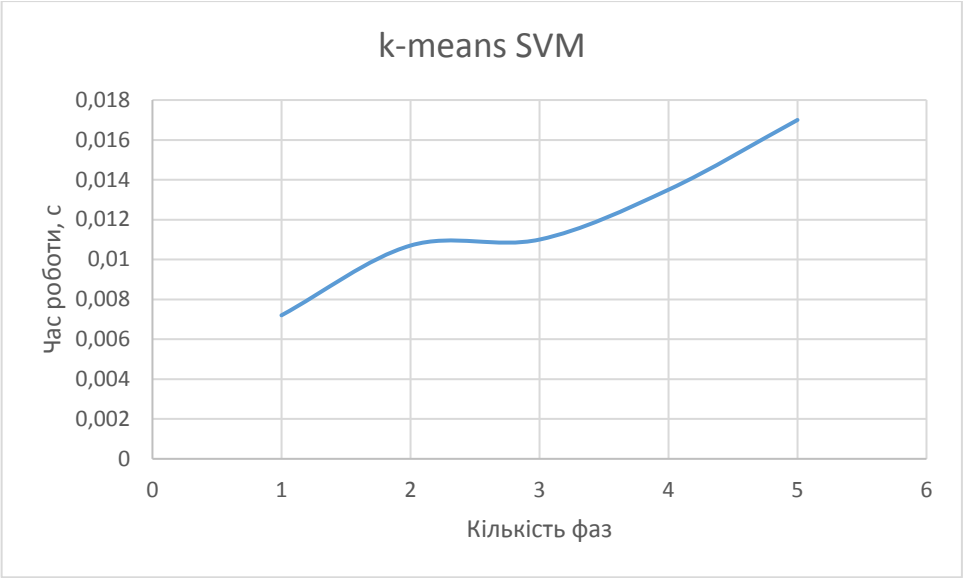

Рисунок 3.7 – Графік часової складності для алгоритму «k-means» SVM

Часова складність алгоритму «k-means» SVM складає:

$$t(N) = 0.0004 \cdot N^3 - 0.003 \cdot N^2 + 0.0096 \cdot N + 0.0004.$$

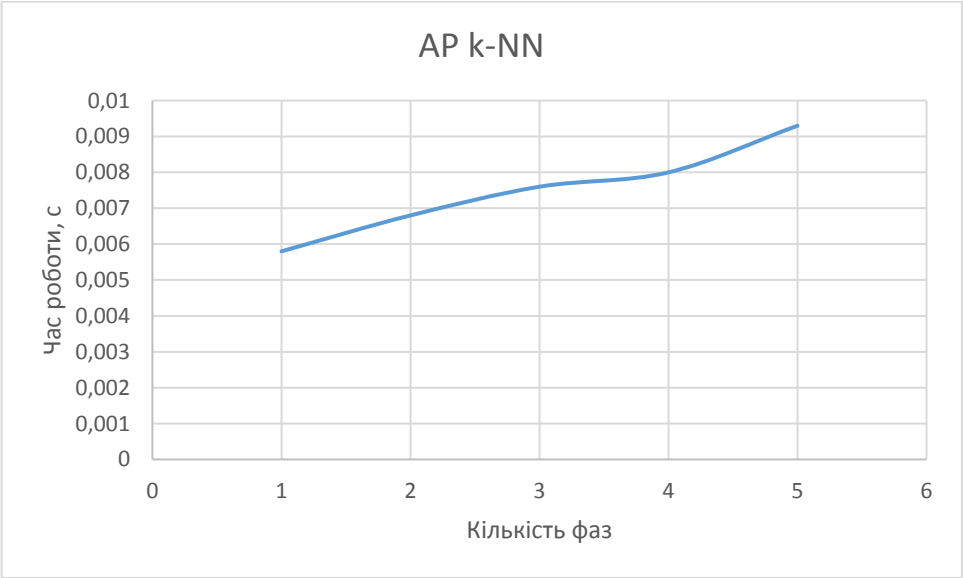

Рисунок 3.8 – Графік часової складності для алгоритму «AP» k-NN

Часова складність алгоритму «AP» k-NN складає:

$$t(N) = 10^{-5} \cdot N^2 + 0.0007 \cdot N + 0.0051.$$

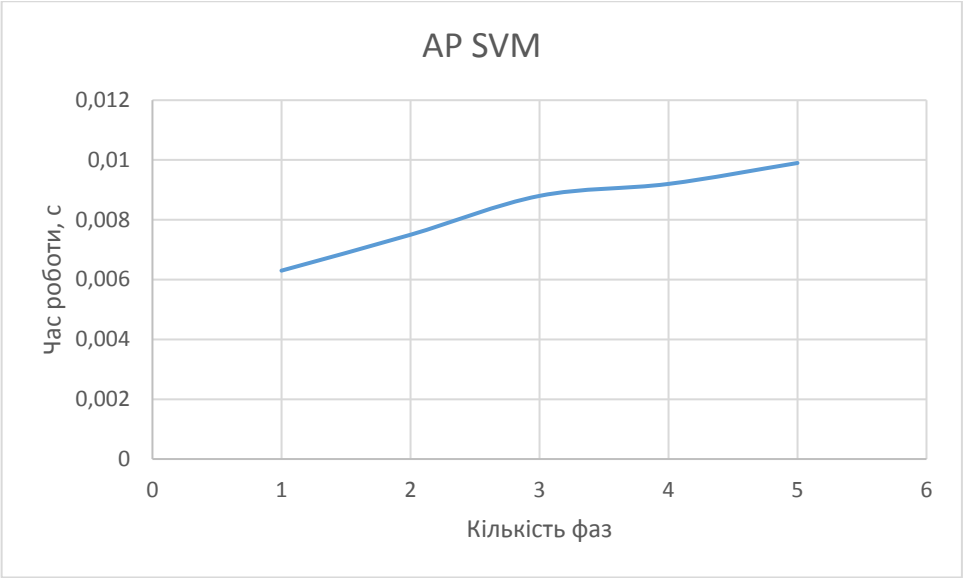

Рисунок 3.9 – Графік часової складності для алгоритму «AP» SVM

Часова складність алгоритму «AP» SVM складає:

$$t(N) = -0.0001 \cdot N^2 + 0.0017 \cdot N + 0.0047.$$

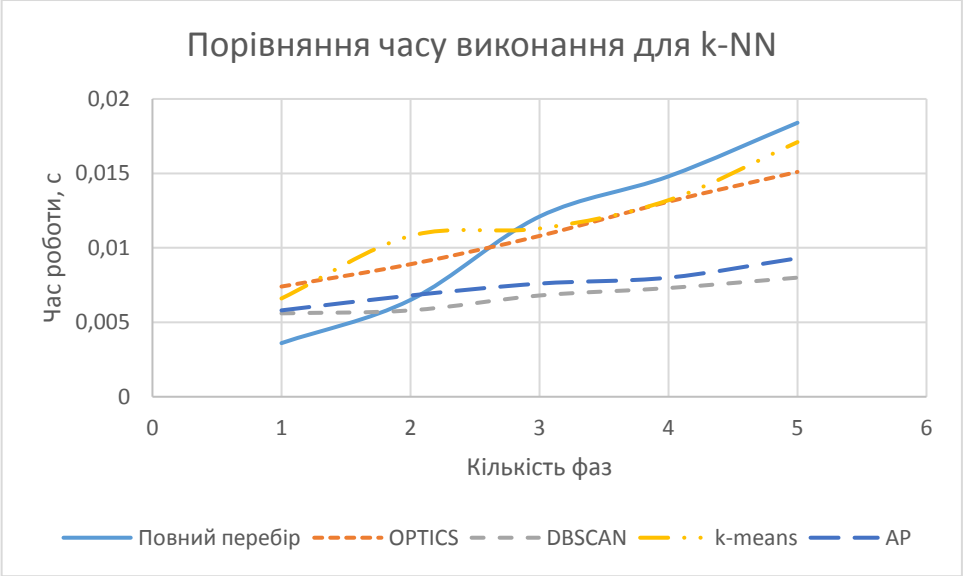

Рисунок 3.10 – Порівняння часу виконання для алгоритму k-NN

Як видно з графіку порівняння часу виконання, як для k-NN, так і для SVM, час роботи повного перебору при малій кількості фаз менший, це пов’язано з малою кількісті можливого обладнання, яке може використовуватися, якщо технологічний процес простий (одна-дві фази), то використовувати додаткові методи не має сенсу, бо повний перебір, дає найліпше значення вартості, ще й працює швидше. Але час роботи повного перебору росте швидше, ніж у

розглянутих методів. Тому для складних ТП гнучкого виробництва з великою кількісті варіантів обладнання використання методів, що розглянуті у роботі виправдано.

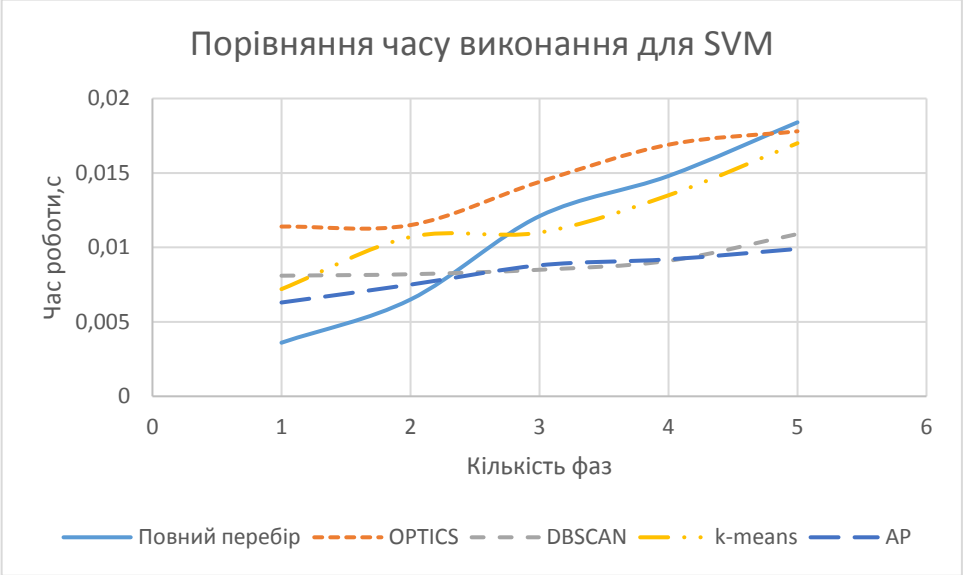

Рисунок 3.11 – Порівняння часу виконання для алгоритму SVM

З рис. 3.10 та рис. 3.11 можна зробити загальний висновок, що алгоритми DBSCAN та AP працюють значно швидше та мають меншу швидкість зростання часу розв’язання задачі зі зростання складності ТП.

ВИСНОВКИ

У результаті виконання дослідження виконано огляд і аналіз сучасного стану проблеми проектування гнучких виробничих технологічних процесів, проаналізовано гнучкі виробничі технологічні процеси як об’єкти проектування, технології їх автоматизованого проектування, методи їх структурно-параметричної оптимізації та сучасні системи автоматизованого проектування технологічних процесів.

За результатами аналізу для подальшого розвитку обрано метод оптимізації ТП на базі процесу-аналога. З метою його удосконалення проаналізовано методи й алгоритми кластеризації та отримання типових технологічних рішень k-means, DBSCAN, OPTICS, AP, опорних векторів (SVM), k-найближчих сусідів (k-NN).

На основі методів кластеризації та спрямованого перебору здійснено удосконалення методу структурно-параметричної оптимізації дискретних виробничих технологічних процесів. З їх використанням розроблено алгоритми та програмне забезпечення розв’язання задачі структурно-параметричної оптимізації технологічних процесів, проведено експериментальне дослідження ефективності базового й удосконаленого методів оптимізації технологічних процесів.

Отримані результати дозволяють підвищувати ефективність виробничих технологічних процесів за рахунок підбору кращих за комплексним показником «продуктивність-вартість» варіантів обладнання. Практичне використання розробленого математичного і програмного забезпечення для технології проектування ТП дозволить зменшити витрати, збільшити конкурентоспроможність, та забезпечити гнучкість виробництва.

Результати досліджень відповідають напрямкам наукових досліджень, що проводяться на кафедрах і в лабораторіях Харківського національного університету радіоелектроніки та присвячені автоматизації проектування виробничих технологічних процесів приладо- та машинобудування, територіально розподілених об’єктів. Вони можуть бути використані в навчальному процесі при вивченні дисциплін з автоматизації проектування і управління дискретними виробничими процесами.

Результати дослідження можуть бути використані в проектних компаніях, що займаються розробкою та реінжинірингом технологічних процесів, і на виробничих підприємствах з гнучкими технологічними ланцюгами.

За результатами дослідження підготовлено доповіді на: міжнародну науково-практичну конференцію «Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі» (м. Київ) [1]; 24-й Міжнародний молодіжний форум «Радіоелектроніка і молодь в XXI столітті» (м. Харків) [38]; звітну наукову конференцію викладачів, докторантів, аспірантів та студентів Прикарпатського національного університету ім. В. Стефаника (м. Івано-Франківськ) [39].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЯ

1. Сідорчук Є.І, Безкоровайний В.В. Інформаційна технологія оптимізації виробничих технологічних процесів // Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі : матеріали міжнар. наук.-практ. конф. (м. Київ, 21-22 квітня 2020 р.). Київ, 2020. (Прийнято до друку).

2. Гибкие производственные процессы. URL: http://chemanalytica.com/book/novyy_spravochnik_khimika_i_tekhnologa/12_obshchie_svedeniya/6308 (дата звернення 11.04.2020).

3. Hoda A. Flexible and reconfigurable manufacturing systems paradigms // International Journal of Flexible Manufacturing Systems. 2005. №17. P. 261-276.

4. Yash P.Gupta, Mahesh C.Gupta. Flexibility and availability of flexible manufacturing systems: An information theory approach // Computers in Industry. 1991. №4. P. 391-406

5. Особенности разработки технологических процессов для гибких производств. URL: https://studopedia.org/8-8502.html (дата звернення 11.04.2020)

6. Diaconu M. Technological Innovation: Concept, Process, Typology and Implications in the Economy // Theoretical and Applied Economics. 2018. №10. P. 127-144.

7. Ятчев А.Г., Бадамшина О.Р., Мурадымов А.М. Технологический процесс и его составляющие. виды технологических процессов // Инновационная наука. 2020. №4. С. 73-75.

8. Онищенко Е.Д., Шацких И.И. Методика разработки модульной технологии механообработки типовой детали машины // Современные материалы, техника и технологии. 2018. №2. С. 101-105.

9. Туровец О.Г., Родионов В.Б., Бухалков М.И. Организация производства и управление предприятием. М., 2010. 217 с.

10. Непомнящий Е.Г. Экономика и управление предприятием : конспект лекций. Таганрог : Изд-во ТРТУ, 2003. 374 с.

11. Петухов, А.В. Системы автоматизированного проектирования технологических процессов: учебн. пособие. Гомель, 2011. 144 с.

12. Ребрин Ю.И. Основы экономики и управления производством : конспект лекций. Таганрог: Изд-во ТРТУ, 2000. 145 с.

13. William D. Engelke. How to Integrate CAD/CAM Systems: Management and Technology. CRC Press, 1987. 400 p.

14. Computer-aided process planning. Wikipedia. URL: https://en.wikipedia.org/wiki/Computer-aided_process_planning#cite_note-1 (дата звернення 15.04.2020).

15. Jain N., Jain V. Computer Aided Process Planning (CAPP) Approach for Advanced Machining Processes // International Journal of Production Research. 1999. №27. P. 231-242.

16. Костюк В.И. Ямпольский Л.С. Гибкие робототехнические системы: общий подход. К.: Выща шк. Головное изд-во, 1988. 72 с.

17. Илюшина С. В. Методы оптимизации ТП // Вестник Казанского технологического университета. 2014. Т. 17. № 8. С. 323-327.

18. Безкоровайний В. В., Шевченко О. Ю. Модель системної оптимізації технологічних об'єктів. Інформаційні технології та комп'ютерне моделювання: матеріали статей Міжнародної науково-практичної конференції, м. Івано-Франківськ, 14-19 травня 2018 року. Івано-Франківськ: п. Голіней О.М., 2018. С. 327-330.

19. Свирский Д.Н., Климентьев А.Л. Автоматизация принятия технологических решений в компактном производстве машиностроительной продукции // Вестник полоцкого государственного университета. Витебск. 2010. № 15. С. 17-20.

20. Цветков, В.Д. Системно-структурное моделирование и автоматизация проектирования ТП. Минск : Наука и техника, 1979. 264 с.

21. Капустин Н.М. Автоматизация проектирования технологических процессов в машиностроении. М. : Машиностроение, 1985. 304с.

22. Фролова И. Н., Кутилова О. И. Анализ современных систем автоматизированного проектирования технологических процессов (САПР ТП) // Труды НГТУ им. Р.Е. Алексеева. 2010. №1. С. 12-15.

23. TECHCARD. URL: https://old.intermech.ru/techcard.htm#mark1 (дата звернення 15.04.2020).

24. T-FLEX PLM. URL: https://www.tflex.ru/ (дата звернення 15.04.2020).

25. ООО НПО «Вертикаль». URL: https://npo-vertical.com.ua/ (дата звернення 15.04.2020).

26. TechnologiCS. URL: https://www.technologics.ru/ (дата звернення 15.04.2020).

27. Илюшина С. В. Методы оптимизации технологических процессов // Вестник Казанского тех. университета. 2014. Т. 17. № 8. С. 323-327.

28. Суслов С.А. Кластерный анализ: сущность, преимущества и недостатки // Вестник НГИЭИ. 2010. №1. С. 19-21.

29. Кутуков Д.С. Применение методов кластеризации для обработки новостного потока. Технические науки: проблемы и перспективы: материалы междунар. науч. конф. (г. СПб, март 2011 г.). СПб.: Реноме, 2011. С. 77–83.

30. Youguo Li, Haiyan Wu. A Clustering Method Based on K-Means Algorithm. International Conference on Solid State Devices. Macao, 2012. 231 p.

31. Ram A., Sunita J., Manoj K. A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases // International Journal of Computer Applications. 2010. №3. P. 7-11.

32. Pranjal Dubey, Anand Rajavat. Comparative study between density based clustering – DBSCAN and OPTICS // International Journal of Advanced Computational Engineering and Networking. 2016. №12. P. 34-37.

33. Yasuhiro Fujiwara, Go Irie, Tomoe Kitahara. Fast Algorithm for Affinity Propagation. Proceedingsof the Twenty-Second International Joint Conference on Artificial Intelligence. Barcelona, 2011. 243 p.

34. Туровец О.Г., Родионов В.Б., Бухалков М.И. Организация производства и управление предприятием : учебник / под ред. О.Г. Турговца. Москва : ИНФА-М, 2009. 544 с.

35.Mariette Awad, Rahul Khanna. Efficient Learning Machines. Apress, 2016. 263 p.

36. Debo Cheng, Shichao Zhang, Zhenyun Deng. kNN Algorithm with Data-Driven k Value. 10th International Conference on Advanced Data Mining and Applications. Guilin, 2014. 741 p.

37. Python. Wikipedia. URL: https://ru.wikipedia.org/wiki/Python (дата звернення 20.04.2020).

38. Сідорчук Є.І., Безкоровайний В.В. Реінжиніринг виробничих технологічних процесів // Радіоелектроніка та молодь у XXI столітті : матеріали 24-го Міжнародного молодіжного форуму. 2020. Т. 6. С. 270-271.

39. Сідорчук Є.І., Безкоровайний В.В. Моделювання задач оптимізації виробничих технологічних процесів // Еврика XXI : матеріали звітної наук. конф. (м. Івано-Франківськ, 9 квітня 2020 р.). м. Івано-Франківськ, 2020. (Прийнято до друку).