




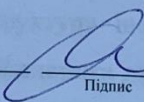
Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 3 курсу, група КНс-20-1  Т.В. Пасічник
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: к. фіз-мат. н., доцент кафедри КН  В.Д. Міхалевський
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: к. т. н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

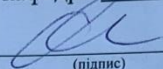
До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
05 06 2023 р. Підпис Ініціали, прізвище

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки
Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор О.В. Бармак

« 06 » 03 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом»
2. Завдання видано студенту Пасічнику Тимофію Віталійовичу
(прізвище, ім'я, по батькові)
3. Керівник роботи доцент кафедри КН Міхалевський Віталій Цезарійович
(посада, прізвище, ім'я, по батькові)
4. Затверджено наказом університету від « 01 » 03 2023 р. № 5
5. Дата видачі завдання студенту: « 03 » 03 2023 р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані: провести аналіз предметної області, огляд методів побудови рекомендаційних систем, існуючих програмних рішень та сформулювати постановку задачі. Розробити спосіб отримання рішення для швейного виробництва і умовах обмежень за мурашиним алгоритмом, спроектувати функціональну структуру інформаційної системи. Здійснити програмну реалізацію інформаційної системи для отримання рішення з використанням сучасних засобів створення програмного забезпечення.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	Виконано
3	Робота над розділом 1 – Характеристика предметної області і постановка задачі	січень 2023	Виконано
4	Робота над розділом 2 – Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом	березень 2023	Виконано
5	Робота над розділом 3 – Програмна реалізація інформаційної системи для отримання рішень з використанням запропонованого способу	квітень 2023	Виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	Виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	Виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	Виконано

Виконавець: студент 3 курсу, група КНс-20-1

Курс, група виконавця

Підпис

Т.В. Пасічник

Ініціали, прізвище

Керівник: к. фіз.-мат. н., доц. кафедри КН

Науковий ступінь, посада

Підпис

В.Ц. Михалевський

Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

Виконавець кваліфікаційної роботи бакалавра: студент групи КНс-20-1 Пасічник Тимофій Віталійович

Керівник кваліфікаційної роботи бакалавра: к.фіз.-мат.н., доцент кафедри КН Міхалевський Віталій Цезарійович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
67	23	12	28	3

Метою кваліфікаційної роботи бакалавра є розробка способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

Результатом виконання кваліфікаційної роботи бакалавра є створення інформаційної системи отримання рішень з використанням запропонованого способу за мурашиним алгоритмом.

Ключові слова: мурашиний алгоритм, прийняття рішень, умови обмежень, швейне виробництво.

Виконавець: студент 3 курсу, група КНс-20-1

Курс, група виконавця



Підпис

Т.В. Пасічник

Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Характеристика предметної області та постановка задачі	6
1.1 Аналіз предметної області	6
1.2 Огляд методів побудови отримання рішення	9
1.3 Аналіз існуючих рішень для подібних систем.....	11
1.4 Постановка задачі.....	16
Розділ 2 Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.....	18
2.1 Інформаційна модель отримання рішення для швейного виробництва в умовах обмежень.....	18
2.2 Метод мурашиного алгоритму	20
2.3 Схема способу отримання рішення за мурашиним алгоритмом	26
2.4 Функціональна структура інформаційної системи отримання рішення	29
2.5 Проектування структури інформаційної системи	33
Розділ 3 Програмна реалізація інформаційної системи отримання рішення з використанням запропонованого способу.....	41
3.1 Структура модулів системи, їх взаємозв'язок	41
3.2 Особливості реалізації способу отримання рішення для швейного виробництва за мурашиним алгоритмом.....	44
3.3 Опис функціональних можливостей інформаційної системи	47
3.4 Вимоги до розгортання системи.....	52
3.5 Проведення експериментів та дослідження роботи системи	56
Висновки	62
Перелік посилань.....	64
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
JS	JavaScript
SQL	Structured query language
IDE	Інтегроване середовище розробки
TS	TypeScript
MA	Мурашиний алгоритм
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
СКДБ	Система керування базами даних
ГБ	Гігабайт
ІТ	Інформаційні технології

Вступ

Кваліфікаційна робота бакалавра присвячена розробці способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

Актуальність теми. Проблематика отримання рішень у виробничій сфері на сьогодні набуває великого попиту. Особливо це гостро відчувається в умовах як стандартних економічних чинників і викликів, так і при наявності непереборних обмежень, таких як карантинні особливості, перебої із живленням, водою та матеріалами. Розвиток технологій дозволяє різного роду підприємствам та організаціям, зокрема, швейним підприємствам, залучати сучасне інформаційне та технічне забезпечення для оперативного прийняття рішення щодо функціонування виробництва та обрання оптимального ситуативного розподілу навантаження на окремих виробничих процесах.

Для отримання рішення можуть використовуватися різні способи, зокрема, мурашиний алгоритм, який дозволяє використовувати наявні множини параметрів виробничих процесів і обмежень для формування рекомендацій в інформаційній системі.

Мета кваліфікаційної роботи бакалавра полягає у розробці способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом та інформаційної системи отримання рішення, які необхідні для автоматизованого отримання рішення за відомостями множини параметрів виробництва та множини обмежень.

Об'єкт дослідження – процес отримання рішення для швейного виробництва в умовах обмежень.

Предмет дослідження – методи збору та аналізу інформації, моделі, способи та засоби автоматизації процесу отримання рішення для швейного виробництва в умовах обмежень.

Завдання кваліфікаційної роботи бакалавра:

- провести аналіз предметної області та відомих підходів до автоматизованого отримання рішення для швейного виробництва в умовах обмежень;
- розробити спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом;
- розробити інформаційну технологію автоматизованого отримання рішення в умовах обмеження;
- спроектувати функціональну та інформаційну структуру інформаційної системи отримання рішення;
- розробити програмну реалізацію інформаційної системи отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом;
- провести функціональне та прикладне дослідження розробленої інформаційної системи для запропонованого способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Швейне виробництво – це процес виготовлення швейних виробів, таких як одяг, взуття, домашні текстильні вироби тощо. Основними характеристиками швейного виробництва є якість виробів, швидкість виробництва, вартість та ефективність використання ресурсів.

Обмеження для швейного виробництва можуть бути різними, такі як обмеження по матеріалах, робочому часу, машинному обладнанню, кваліфікації працівників та інше. Види обмежень включають обмеження на виробництво певної кількості одиниць продукції за певний час, обмеження на розміщення інфраструктури, обмеження на кількість робочих місць тощо [1-4].

Перелік задач на швейному підприємстві, для яких потрібно приймати рішення, включає такі питання, як планування виробництва, розподіл ресурсів, контроль якості, управління витратами тощо [5]. Для автоматизації швейного виробництва потрібно збирати інформацію про виробничі процеси, якість виробів, витрати на матеріали та працю, використання машинного обладнання тощо. В умовах війни виникають нові види обмежень: відсутність (недостатність) людей, матеріалів, живлення, води та інших ресурсів [1-4, 6, 7].

Сутність предметної області швейного виробництва полягає в виробництві швейних виробів. Основними властивостями є тип виробів, якість, вартість, ефективність використання ресурсів та обмеження. Атрибути таблиці БД можуть включати інформацію про назву виробу, кількість виробів, якість, вартість, ресурси, використані для виробництва тощо [5, 7, 8].

Процеси предметної області швейного виробництва можуть включати планування виробництва, підготовку матеріалів, крою і шиття виробів, контроль якості та упакування. Кроки в процесі швейного виробництва можуть включати збір замовлень від клієнтів, планування виробництва на основі обмежень і

забезпечення необхідних ресурсів, розробку та виготовлення пробних виробів, масове виробництво, контроль якості і упакування для доставки клієнту [5, 8].

При автоматизації швейного виробництва можуть використовуватись різні способи подання інформації, такі як використання баз даних, створення програмного забезпечення для планування виробництва і контролю якості, використання сенсорів для відстеження виробничих процесів та інші технології. Для автоматизації швейного виробництва можуть збиратись інформація про кількість виробів, використання ресурсів, якість виробів та інші параметри, які необхідні для прийняття рішень [8-13].

У підсумку, швейне виробництво – це процес виготовлення швейних виробів, який має різні обмеження, такі як обмеження на ресурси, обладнання та робочий час. Для ефективного управління швейним виробництвом необхідно приймати рішення з планування виробництва, розподілу ресурсів, контролю якості та управління витратами.

При автоматизації швейного виробництва можуть використовуватись різні технології для збору та обробки інформації, яка необхідна для прийняття рішень. У швейному виробництві зазвичай використовуються різні цехи для виконання різних етапів виробничого процесу. Основними цехами, які можна знайти на швейних підприємствах, є [5, 7-9]:

- Цех розкрою. У цьому цеху здійснюється розкрій тканин та інших матеріалів перед пошиттям. Розкрійники використовують спеціальні ножиці або автоматизовані розкрійні машини для отримання деталей виробу за встановленими шаблонами.

- Цех пошиття. Це найбільший цех у швейному виробництві, де здійснюється безпосереднє пошиття швейних виробів. Тут розташовані ряди швейних машин, на яких працівники шиють шви, кріплять деталі, наносять застрочки та виконують інші операції.

- Цех обробки. Після пошиття швейні вироби можуть потребувати додаткової обробки, такої як прасування, прання, фарбування або нанесення

декоративних елементів. У цьому цеху працюють прасувальники, фарбувальники та інші спеціалісти для завершення виробу.

– Цех контролю якості. Цей цех відповідає за контроль якості готових швейних виробів. Спеціалісти проводять огляд, вимірювання та перевірку виробів, щоб переконатися, що вони відповідають встановленим стандартам якості та вимогам замовника.

– Цех упаковки. Після проходження контролю якості готові швейні вироби передаються в цех упаковки. Тут вони пакуються у відповідні контейнери або упаковки для зручності зберігання та транспортування.

В цеху розкрою використовуються спеціальні розкрійні столи або автоматизовані розкрійні машини. Розкрійні столи зазвичай мають плоску поверхню з маркуваннями для точного вимірювання та розміщення матеріалу перед розкроєм. Автоматизовані розкрійні машини використовуються для швидкого та точного розкрою матеріалу за допомогою комп'ютерного керування. Для розкрою матеріалу використовуються шаблони або креслення, які містять інформацію про форму та розміри деталей виробу. Ці шаблони або креслення можуть бути накреслені вручну або створені за допомогою комп'ютерних програм. У процесі розкрою виникають відходи матеріалу. Цех розкрою старається мінімізувати відходи, оптимізуючи розміщення деталей на матеріалі та використовуючи спеціальні методи розкрою, які дозволяють ефективно використовувати матеріали.

Цех пошиття є одним з найбільших та найважливіших цехів на швейному виробництві. У цьому цеху здійснюється безпосереднє пошиття швейних виробів, включаючи зшивання деталей, створення швів, кріплення елементів та завершення деталей виробу. В цеху пошиття використовуються різні матеріали, такі як тканини, нитки, гудзики, блискавки, тасьми та інші комплектуючі. Робітники цього цеху повинні мати доступ до різних видів матеріалів та комплектуючих, щоб виконувати різноманітні завдання пошиття. В цеху пошиття робочі місця організовані таким чином, щоб забезпечити оптимальну ефективність та потік роботи.

Цех обробки є важливим відділом на швейному виробництві, де проводяться різні операції для обробки та доведення швейних виробів до готового стану. Основна мета цього цеху - забезпечити чистоту, гладкість, естетичний вигляд та проведення додаткових функцій (прасування, контроль основних операцій тощо) із виробами перед їх відправкою на реалізацію. У цеху обробки виконуються різні декоративні операції для покращення вигляду та стилю виробів. Це може включати нанесення вишивки, аплікацій тощо.

Цех контролю якості є важливим відділом на швейному виробництві, в якому проводиться оцінка та перевірка якості швейних виробів згідно з встановленими стандартами та специфікаціями, забезпечується облік та документування продукції.

Цех упаковки є важливою частиною швейного виробництва, де проводяться операції з упаковки та підготовки швейних виробів до транспортування та реалізації. Головною метою цього цеху є забезпечення безпечного та ефективного упакування виробів для їхнього зберігання, транспортування та презентації.

1.2 Огляд методів побудови отримання рішення

Методи отримання рішень для швейного виробництва можуть бути різними, залежно від конкретних задач і обставин [7-9]. Одним з методів є математичне моделювання, коли використовуються математичні методи та алгоритми для розрахунку оптимальних рішень на основі певних обмежень і критеріїв. Наприклад, можна застосувати математичну модель для планування виробництва, яка враховує обмеження на ресурси, терміни виконання замовлень та інші фактори [5, 7].

Іншим методом є експертна оцінка, коли використовуються знання та досвід фахівців для прийняття рішень. Наприклад, при виборі матеріалів або розробці нових моделей виробів може бути корисним залучення дизайнерів і швачок, які мають великий досвід у даній галузі [9].

Для швейного виробництва характерні особливості, які варто враховувати при прийнятті рішень [7]. Насамперед, велика кількість різноманітних виробів, які можуть мати різні розміри, фасони, тканини та інші параметри, що може ускладнювати планування виробництва та контроль якості. Крім того, швейне виробництво часто має високу ступінь сезонності, коли попит на певні види виробів може дуже змінюватись в залежності від сезону або модних тенденцій

Обмеження для швейного виробництва можуть бути різними, від обмежень на ресурси, такі як тканини, нитки, замки, до обмежень на робочий час та обладнання. Крім того, можуть існувати обмеження на терміни виконання замовлень та якість виробів [8-13].

До інших обмежень для швейного виробництва можуть відноситись такі фактори, як обмеження на кількість працівників та їх кваліфікацію, обмеження на простір виробничих приміщень та зберігання готової продукції, а також обмеження на фінансові ресурси [14-16].

Оскільки швейне виробництво має складну структуру, існує багато задач, для яких потрібно приймати рішення [17]. До таких задач можуть відноситись:

- планування виробництва;
- контроль якості;
- оптимізація витрат;
- управління запасами;
- розробка нових моделей виробів.

Планування виробництва включає визначення обсягів виробництва, вибір методів виробництва та визначення ресурсів. У цьому процесі враховуються наявні ресурси, технічні можливості та попит на продукцію [18-20]. Задачею планування є забезпечення оптимального використання ресурсів та виробництва продукції відповідно до попиту [7, 16].

Контроль якості включає визначення критеріїв якості виробів, контроль відповідності виробів вимогам замовника та стандартам якості. Цей процес встановлює, чи вироби відповідають необхідним стандартам та вимогам якості,

забезпечуючи задоволення потреб замовника і дотримання обумовлених стандартів [20-22].

Оптимізація витрат передбачає визначення найбільш ефективних способів використання ресурсів та зниження витрат на виробництво. Цей процес може включати впровадження енергозберігаючих технологій, раціоналізацію виробничих процесів, управління витратами на матеріали та інші заходи з метою зниження загальних витрат підприємства.

Управління запасами охоплює визначення оптимальних запасів готової продукції та необхідних матеріалів для виробництва [8]. Цей процес передбачає раціональне планування, замовлення та контроль за запасами з метою забезпечення безперебійного виробництва і задоволення потреб замовників при мінімальних витратах на забезпечення виробничих технологій.

Розробка нових моделей виробів включає вибір дизайну, визначення технології виготовлення та розробку прототипу. Цей процес передбачає творчий підхід та інженерну діяльність для створення нових продуктів або модернізації існуючих. Включає в себе аналіз ринку, дослідження потреб споживачів, проектування та тестування нових моделей перед введенням їх в серійне виробництво [22].

При автоматизації процесів у швейному виробництві можна використовувати різні способи подання інформації, зокрема:

1. Текстові описи: короткий опис завдання або результатів аналізу.
2. Графічні зображення: діаграми та графіки для візуалізації даних.
3. Таблиці: таблиці з даними про виробництво, витрати, запаси, кількість працівників тощо.

1.3 Аналіз існуючих рішень для подібних систем

Класифікація існуючих рішень за напрямками розробки [16, 22-28]:

1. Системи автоматизованого проектування та керування виробництвом (CAD/CAM/PLM systems). Ці системи забезпечують повну

автоматизацію процесу від створення дизайну до виготовлення готового виробу. Вони використовуються для створення моделей, керування виробництвом, планування виробничих процесів та інших цілей.

2. Системи моніторингу та контролю якості (Quality Control Systems). Ці системи забезпечують моніторинг та контроль якості виробів на різних етапах виробництва. Вони використовуються для виявлення та виправлення дефектів, підвищення якості виробів та зниження витрат.

3. Системи планування та управління виробництвом (Manufacturing Execution Systems). Ці системи забезпечують планування та управління виробничими процесами на різних етапах виробництва. Вони використовуються для оптимізації виробництва, зниження витрат та підвищення ефективності виробництва.

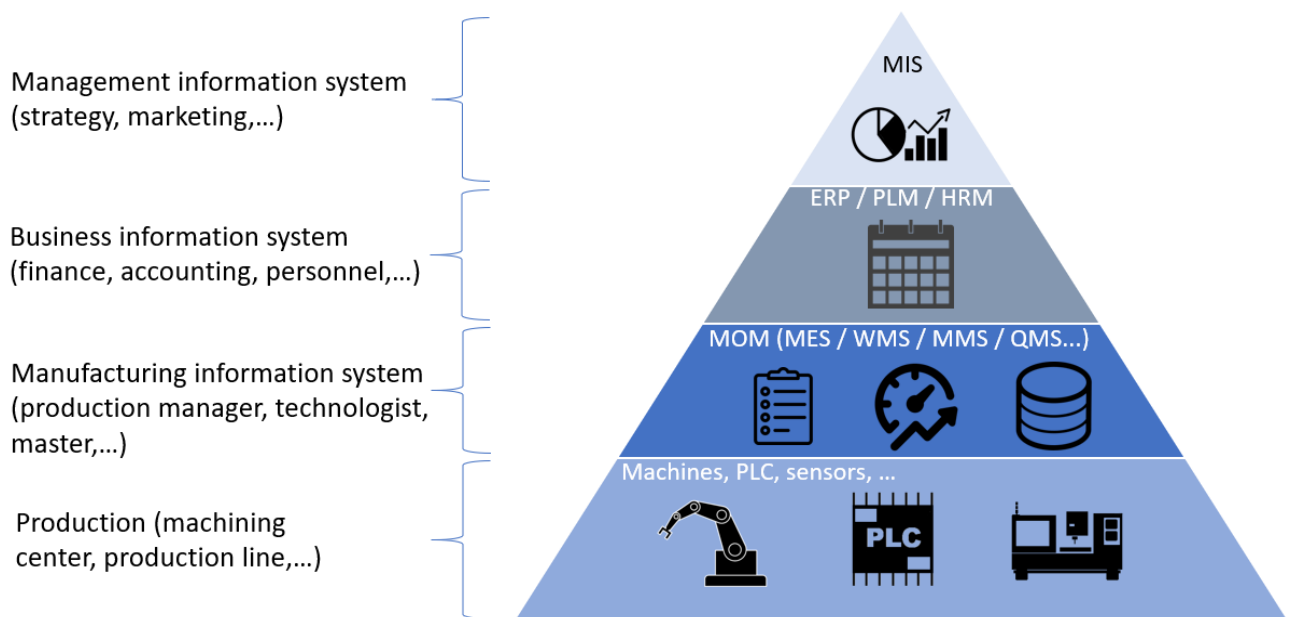


Рисунок 1.1 – Пірамідальна схема системи управління виробництвом

Опишемо характерних представників кожного класу:

1. CAD/CAM/PLM системи: AutoCAD, SolidWorks, CATIA, Siemens NX, PTC Creo, ANSYS.

2. Системи моніторингу та контролю якості: Cognex, Keyence, Omron, Hexagon.

3. Системи планування та управління виробництвом: SAP, Oracle, Microsoft Dynamics, Wonderware MES.

Із існуючих готових рішень для швейного виробництва виділимо систему Eqman – програму для обліку матеріальних цінностей, яка надає можливість запровадження аналізу швейного виробництва та автоматизації цього процесу.

Основною перевагою автоматизації в цій системі є прозора організація складського обліку у швейному виробництві.

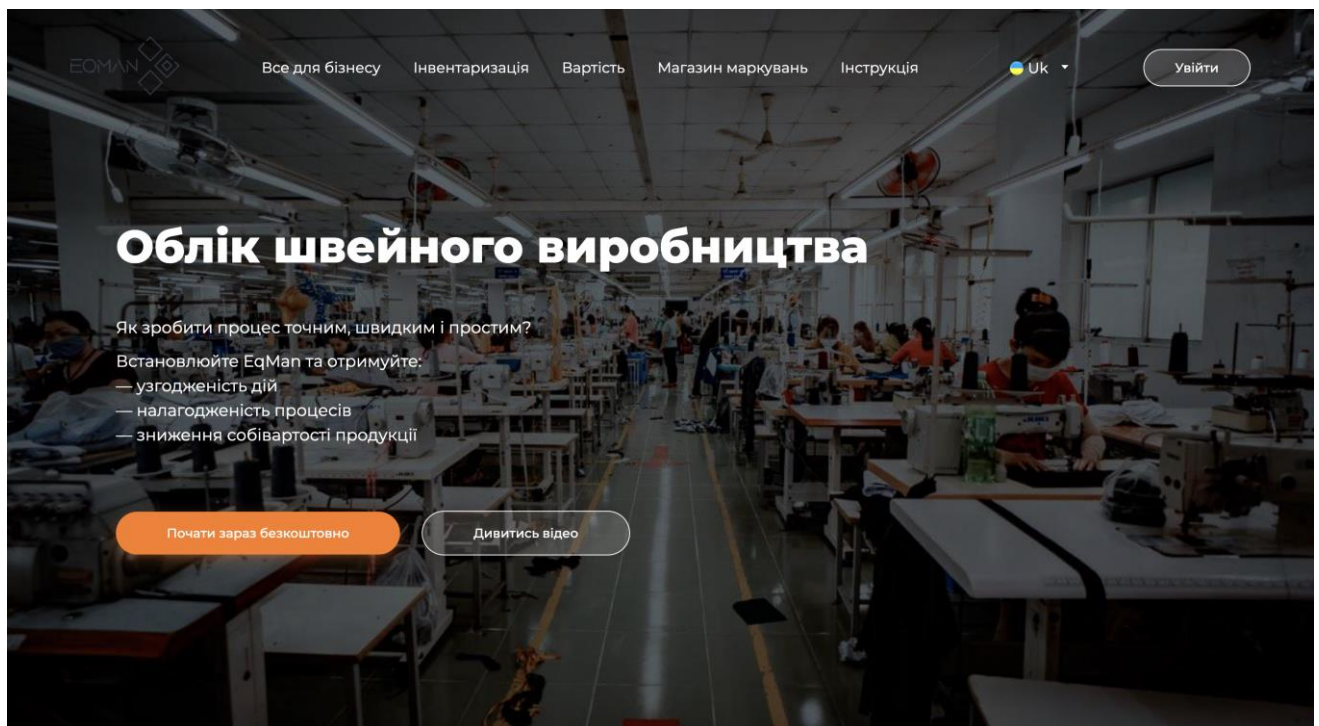


Рисунок 1.2 – Головний сайт Eqman обліку матеріалів швейного виробництва

Нижче наведено конкретні завдання, які розв'язує система Eqman:

1. Підвищення швидкості та точності приймання матеріалів та видачі товарів: Завдяки системі Eqman уникають пересортиці, нестачі та значних витрат часу на оформлення. Процес приймання та видачі стає більш швидким і точним.
2. Оптимізація штату: Завдяки автоматизації обліку швейного виробництва з використанням системи Eqman, необхідна кількість працівників для виконання інвентаризації, контролю та оформлення зменшується.
3. Зниження рівня втрат: У системі Eqman всі товарно-матеріальні цінності закріплюються за конкретними працівниками, завдяки подвійному

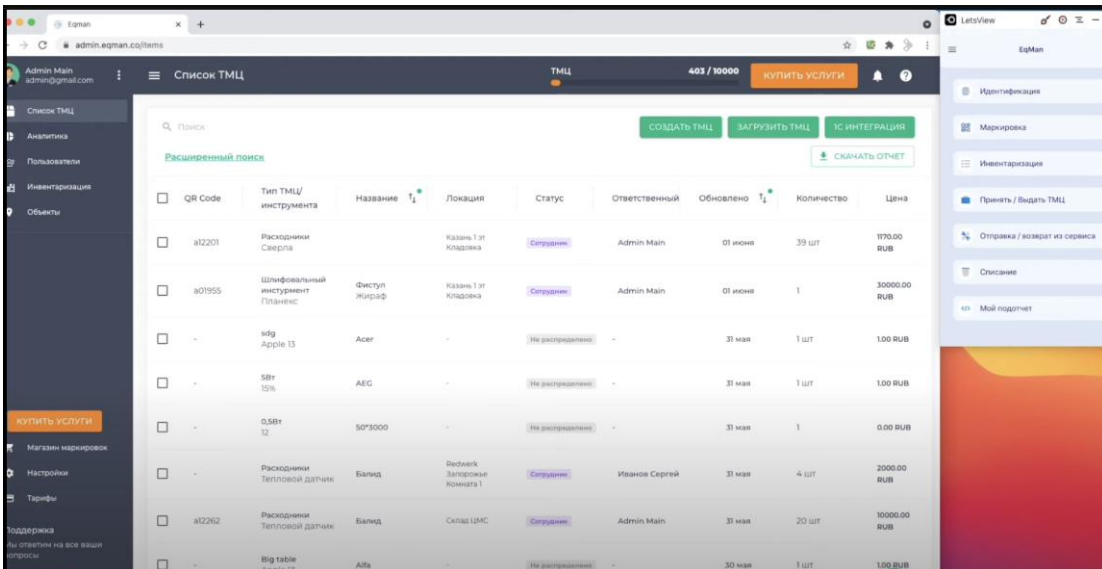
підтвердженню. Це ускладнює несанкціоноване списання товарів, а також підвищує увагу до збереження інструменту та обладнання через особисту відповідальність.

4. Відсутність перевищення бюджету: Завдяки чіткому відстеженню витрат у системі Eqman, ви можете купувати матеріали своєчасно та у необхідних кількостях, уникнувши непотрібних запасів, що допомагає знизити витрати.

5. Розміщення та пошук: Використання системи Eqman дозволяє створювати локації та розміщувати продукцію і заготовки в зручних місцях. Це допомагає уникнути втрати товару та зайвих зусиль при пошуку, оптимізуючи процес розміщення товарів та прискорюючи його.

6. Проведення інвентаризації: Застосування системи Eqman дозволяє зменшити витрати часу та зусиль під час інвентаризації. Завдяки скануванню позначок, дані миттєво вносяться до програми. У середньому, інвентаризація з використанням системи Eqman відбувається утричі швидше, ніж без цієї програми.

7. Просте масштабування бізнесу: Завдяки системі Eqman, можна відкривати нові ательє в будь-якій точці світу та додавати всю інформацію до загальної програми. Це спрощує процес масштабування бізнесу.



QR Code	Тип ТМЦ/інструмента	Назва	Локація	Статус	Відповідальний	Оновлено	Кількість	Ціна
42201	Расходники Сверла	Казань 1 эт. Кладовка	Сторонки	Admin Main	01 июня	39 шт	170.00 RUB	
401955	Специальный инструмент Планыкс	Фастул Жираф	Казань 1 эт. Кладовка	Сторонки	Admin Main	01 июня	1	30000.00 RUB
449	Apple 13	Acer	-	Не распределены	-	31 мая	1 шт	1.00 RUB
581	12A	AEG	-	Не распределены	-	31 мая	1 шт	1.00 RUB
0581	12	5073000	-	Не распределены	-	31 мая	1	0.00 RUB
-	Расходники Тепловой датчик	Балдз	Редовый Запасные Комната 1	Сторонки	Иванов Сергей	31 мая	4 шт	2000.00 RUB
42262	Расходники Тепловой датчик	Балдз	Склад LMS	Сторонки	Admin Main	31 мая	20 шт	10000.00 RUB
-	Big table Apple 13	Aix	-	Не распределены	-	30 мая	1 шт	1.00 RUB

Рисунок 1.3 – Вікно обліку товарно-матеріальних цінностей в Eqman

Таким чином, Еrman є однією із систем обліку та автоматизації у швейному виробництві, на яку варто орієнтуватися.

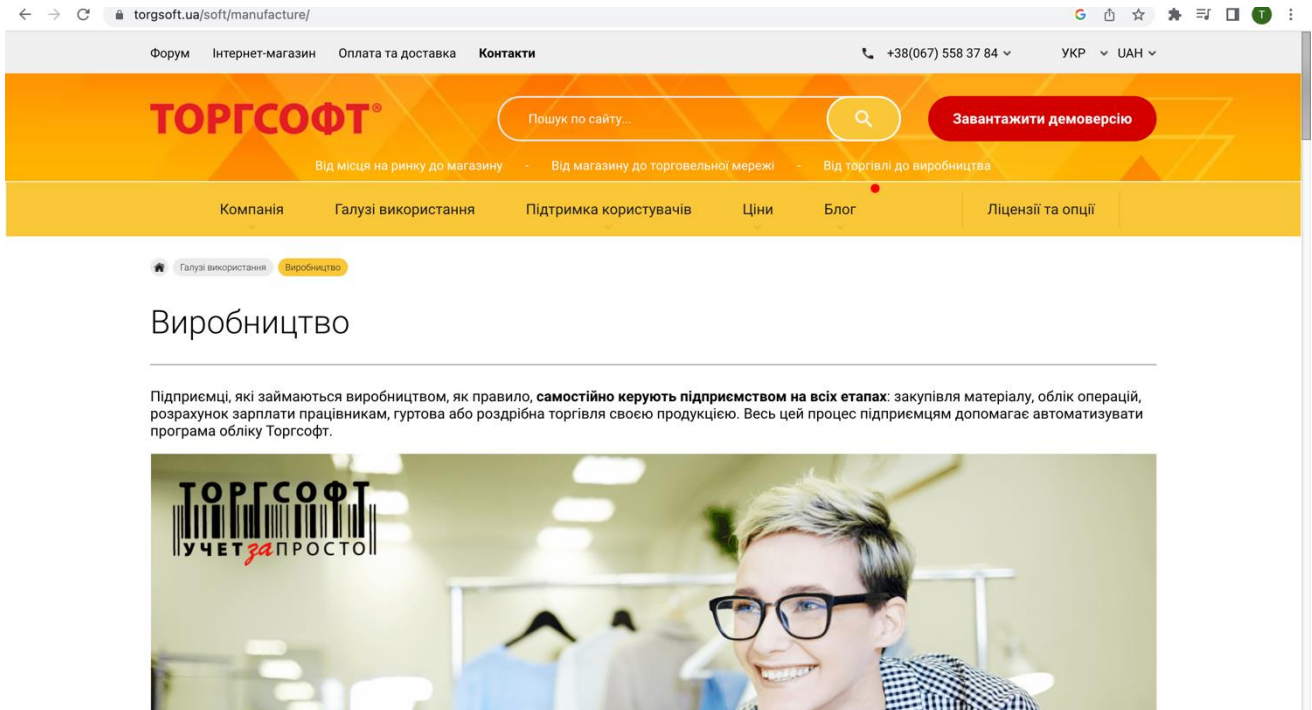


Рисунок 1.4 — Головний сайт сервісу Торгсофт

Також є ще рішення від сервісу Торгсофт, яке надає можливість користування програмою Виробництво. Торгсофт обіцяє такі функції відстеження програмою:

- Облік закуплених та фактично використаних матеріалів, фурнітури, пакування, відходів, браку.
- Облік витрат на кожну виробничу операцію, трудовитрат на виготовлення конкретної партії товару. Підрахунок собівартості партії та кожної одиниці товару.
- Стан складу матеріалів: дані по закупівлях, залишках, витратах на кожну партію, брак, відходи (тканини, нитки, фурнітура, пакувальний матеріал та інше).
- Друк маршрутних листів і виробничих актів з переліком операцій, в яких автоматично фіксуються виконані роботи та їх виконавці.

- Планування виробництва по кожному виробу згідно з технологічними картами, розрахунок необхідних матеріалів.
- Автоматична фіксація виконаних робіт та її виконавців в маршрутному листі.
- Собівартість виробів, розрахована з урахуванням витрачених на партію матеріалів, трудовитрат, браку.
- Стан складу готових виробів, що надходять з виробництва в продаж: кількість за моделями та розмірами, націнка, ціна гуртова та роздрібна.
- Ціна товару для роздрібної торгівлі та гуртових покупців, в залежності від партії.
- Замовлення від клієнта моделей товару для виробництва.
- Контроль та облік матеріалів, зарезервованих для виробництва.
- Автоматичний облік зарплати для співробітників та ведення історії виплат (розрахункових листів).

1.4 Постановка задачі

Метою кваліфікаційної роботи бакалавра є розробка способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом та інформаційної системи отримання рішення, які необхідні для автоматизованого отримання рішення за відомостями множини параметрів виробництва та множини обмежень.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- *провести аналіз предметної області та відомих підходів до автоматизованого отримання рішення для швейного виробництва в умовах обмежень;*
- *розробити спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом;*

- розробити інформаційну технологію автоматизованого отримання рішення в умовах обмеження;*
- спроектувати функціональну та інформаційну структуру інформаційної системи отримання рішення;*
- розробити програмну реалізацію інформаційної системи отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом;*
- провести функціональне та прикладне дослідження розробленої інформаційної системи для запропонованого способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.*

Розділ 2 Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом

2.1 Інформаційна модель отримання рішення для швейного виробництва в умовах обмежень

Інформаційна модель отримання рішення для швейного виробництва в умовах обмежень – це розробка, яка дозволяє автоматизувати процес отримання рішення щодо оптимальних параметрів виробництва для швейної компанії в умовах обмежень.

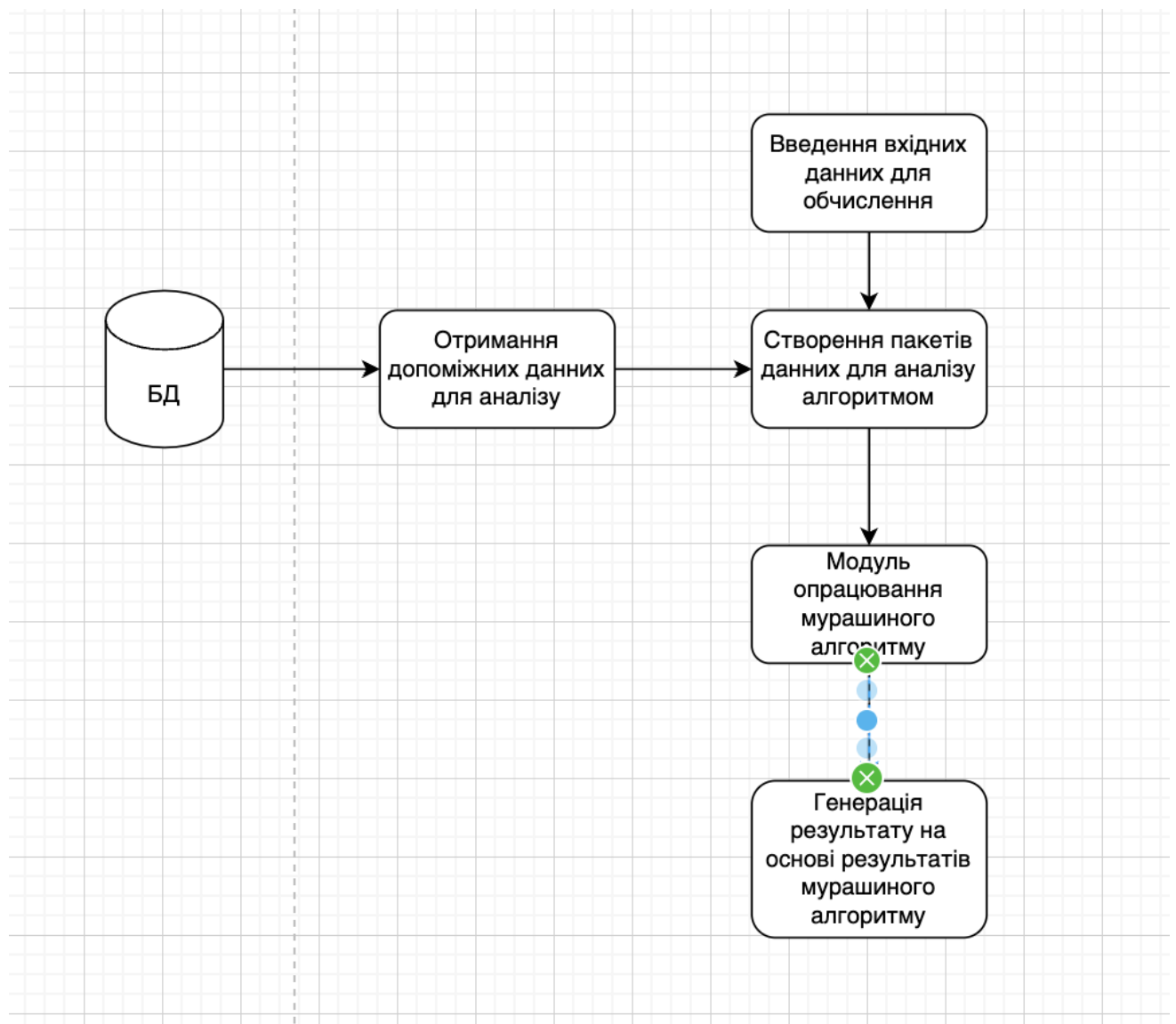


Рисунок 2.1 — Інформаційна схема моделі отримання рішення

Інформаційна модель швейного виробництва в умовах обмежень включає в себе наступні модулі:

- виконавці;
- розклад робіт;
- інструменти;
- ресурси;
- матеріали;
- товари;
- процеси.

Основні компоненти інформаційної моделі:

1. База даних – містить інформацію про параметри виробництва, обмеження та поточний стан процесу виробництва.
2. Аналітичний модуль – використовує алгоритм Мурашиного алгоритму для знаходження оптимальних параметрів виробництва.
3. Модуль візуалізації – дозволяє відображати результати аналізу та визначення оптимальних параметрів виробництва у зручному для користувача форматі, такому як графіки, діаграми тощо.

Переваги інформаційної моделі отримання рішення для швейного виробництва в умовах обмежень:

1. Автоматизація процесу – інформаційна модель дозволяє автоматизувати процес отримання рішення, що зменшує час і зусилля, необхідні для знаходження оптимальних параметрів виробництва.
2. Підвищення точності – за допомогою аналітичного модуля на основі алгоритму Мурашиного алгоритму можна знайти оптимальні параметри виробництва з високою точністю.
3. Інтеграція з базою даних – інформаційна модель підключена до бази даних, що дозволяє автоматично оновлювати інформацію про стан виробництва та обмеження.

2.2 Метод мурашиного алгоритму

Мурашиний алгоритм – це один з ефективних поліноміальних алгоритмів для знаходження наближених розв'язків задачі оптимізації (комівояжера), а також аналогічних завдань пошуку маршрутів на графах [23, 24, 27, 28].

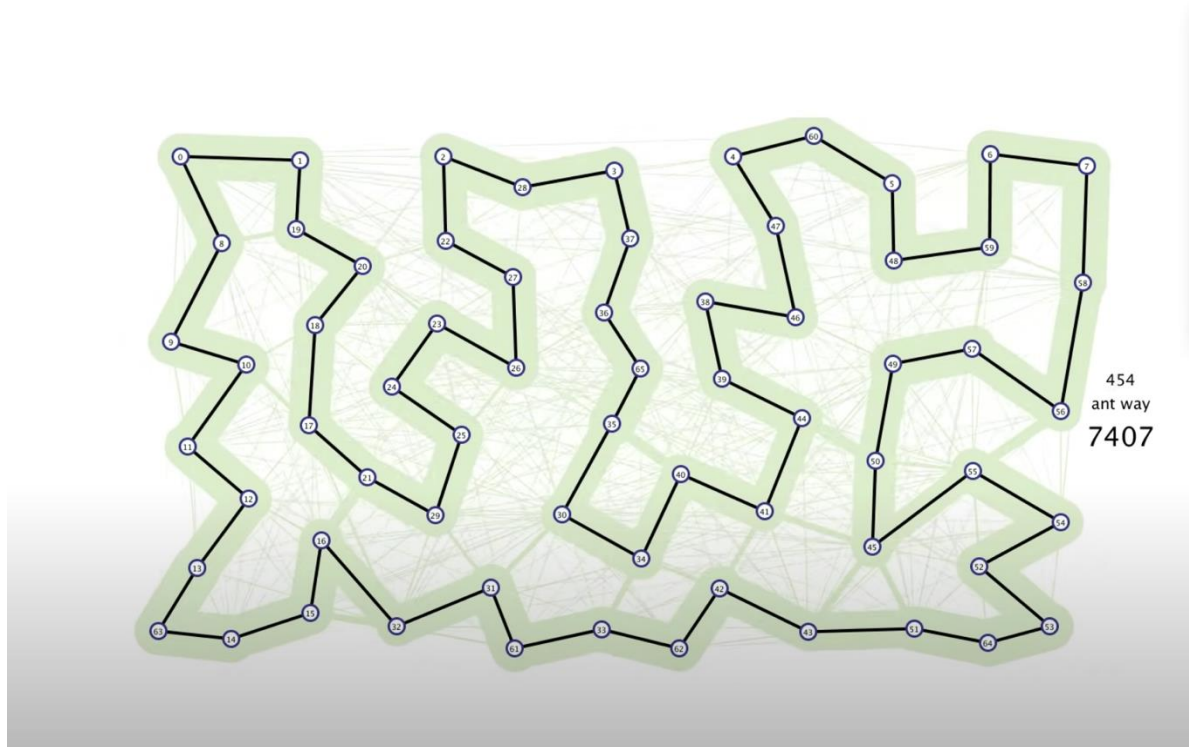


Рисунок 2.2 – Використання методу мурашиного алгоритму для задачі оптимізації

Ідея алгоритму базується на поведінці колонії мурах, які залишають феромонні сліди на своєму шляху, щоб покращити свій шлях до джерела їжі. Алгоритм використовує цей механізм для пошуку найкращого шляху в графі.

Алгоритм складається з декількох етапів:

- Ініціалізація: створюється колонія мурах, яка випускається в граф для пошуку найкращого шляху. Кожна мураха починає з деякої вершини і рухається випадковим чином по графу.
- Вибір шляху: мураха вибирає наступну вершину, керуючись ймовірністю, яка залежить від кількості феромонів на ребрі та його довжини.

Чим більше феромонів залишили інші мурахи на ребрі, тим більша ймовірність того, що мураха піде по цьому ребру:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum \tau_{im}^{\alpha} \eta_{im}^{\beta}} \quad (2.1)$$

де P_{ij} — ймовірність переходу мурахи з вершини i в j , τ — кількість феромонів на шляху з i вершини до j , η — видимість, довжина шляху переходу з i вершини в j , $\alpha + \beta = 1$, m - всі інші доступні вершини для переходу

– Оновлення феромонів: коли мураха дійде до кінцевої вершини, вона оцінює якість свого шляху. Якщо шлях короткий, мураха залишає більше феромонів на всіх ребрах свого шляху. Якщо шлях довгий, то мураха залишає менше феромонів. Цей механізм допомагає забезпечити збалансованість між інтенсивністю феромонів і довжиною шляху.

– Евапорація феромонів: феромони поступово випаровуються з поверхонь, на яких вони розповсюджуються. Це відбувається через природні процеси, такі як вплив температури і повітряної вологості. Чим більше часу пройшло з моменту виділення феромонів, тим більша ймовірність, що вони випаруються повністю. Евапорація феромонів має важливе значення, оскільки вона дозволяє системі мурашиного алгоритму підтримувати актуальну інформацію про шляхи.

Варіації мурашиного алгоритму включають в себе:

– Анти-колонія або колонія мурашиних систем (Ant Colony System, ACS): Ця варіація використовує концепцію колоній мурах, де кілька мурах працюють разом для знаходження оптимального шляху. Вона включає в себе розвиток феромонів, використання евристичних правил для вибору маршруту та оновлення рівнів феромонів. ACS зазвичай досягає швидшої збіжності до оптимального рішення.

– Мурашиний системний алгоритм (Max–Min Ant System, MMAS): У MMAS використовуються три основні концепції – максимізація та мінімізація феромонів та обмеження феромону. Він контролює рівні феромонів на ребрах

графу, використовуючи верхні та нижні межі феромону. Це дозволяє уникнути пасток локального мінімуму та підвищити різноманітність розв'язків.

– Мурашиний алгоритм на основі розподілу (Rank-based Ant System, ASrank): Ця варіація алгоритму використовує розподіл Леві для вибору наступного маршруту мурахи. Замість використання ймовірностей, як у класичному мурашиному алгоритмі, ASrank використовує значення рангу для вибору маршруту. Це допомагає збільшити різноманітність пошукового простору та сприяє збільшенню збіжності.

– Мурашиний алгоритм на основі дифузії феромонів (Ant Colony Optimization with Diffusion, ACO-D): Ця варіація алгоритму поєднує ідеї мурашиного алгоритму з дифузиею феромонів. Вона дозволяє феромонам поширюватись від одного ребра до інших, навіть якщо мураха не проходить через ці ребра. Це поліпшує здатність алгоритму до виявлення та експлуатації потенційно можливих розв'язків.

Ці варіації мурашиного алгоритму мають різні переваги та підходять для різних типів задач комбінаторної оптимізації. Вибір конкретної варіації залежить від характеру задачі та особливостей пошукового простору.

Колонія мурашиних систем (Ant Colony System, ACS) – це одна з варіацій мурашиного алгоритму, яка використовує концепцію колоній мурах для вирішення задач оптимізації маршруту. ACS є покращеною версією класичного мурашиного алгоритму і дозволяє досягати швидшої збіжності до оптимального рішення.

Основна ідея ACS полягає в наслідуванні поведінки мурах у природі, де мурахи працюють разом для знаходження найкоротшого шляху до джерела їжі. Алгоритм використовує феромони, які мурахи залишають на своєму шляху, для спілкування та сприяння вибору оптимального маршруту.

Основні кроки ACS:

1. Ініціалізація феромонів: Початкові рівні феромонів встановлюються на всіх ребрах графу. Зазвичай ці значення ініціалізуються невеликими випадковими значеннями.

2. Пошук розв'язку: Кожна мураха розпочинає свій пошук з початкової точки і рухається по ребрах графу. Мурахи вибирають наступний вузол згідно з ймовірнісною формулою, яка залежить від рівнів феромонів та евристичних інформаційних значень. Чим більші феромони на ребрах та чим ближче вузол до потрібного розв'язку, тим більша ймовірність його вибору.

3. Оновлення феромонів: Після того, як всі мурахи завершили свій шлях, відбувається оновлення рівнів феромонів. Зазвичай, феромони випаровуються (зменшуються) на всіх ребрах, а потім мурахи додають свої феромони на вибраних ними шляхах. Додані значення феромонів можуть бути залежними від якості розв'язку, отриманого мурахою.

4. Вдосконалення розв'язку: Кращі розв'язки, знайдені протягом ітерації, можуть бути використані для вдосконалення феромонних слідів. Це допомагає прискорити збіжність алгоритму та збільшити ймовірність вибору оптимальних маршрутів у майбутньому.

5. Повторення: Кроки 2–4 повторюються до досягнення заданої умови зупинки, такої як задана кількість ітерацій або досягнення заданої якості розв'язку.

ACS дозволяє мурахам обмінюватися інформацією про найдешевший маршрут, що призводить до підвищення ймовірності вибору цього маршруту іншими мурахами. Таким чином, використовуючи механізми феромонів та евристичні інформаційні значення, ACS поступово знаходить оптимальний шлях у пошуковому просторі.

Варіація ACS може включати додаткові елементи, такі як локальні оновлення феромонів, додаткові евристичні правила або механізми відбору лідерів мурах. Ці додаткові елементи можуть варіюватися в залежності від конкретної реалізації алгоритму та вимог задачі оптимізації.

Мурашиний системний алгоритм (Max–Min Ant System, MMAS) є однією з варіацій мурашиного алгоритму, яка використовує механізми контролю феромонів для покращення збіжності та уникнення пасток локального мінімуму.

MMAS є ефективним і стійким до параметрів алгоритмом та здатним досягати оптимальних розв'язків у задачах комбінаторної оптимізації.

Основні особливості MMAS:

- Верхня та нижня межі феромонів: MMAS використовує верхню (Max) та нижню (Min) межі для обмеження значень феромонів на ребрах графу. Це допомагає контролювати розподіл феромонів та уникати їх занадто швидкого зростання або спадання.

- Оновлення феромонів: Рівні феромонів оновлюються за допомогою формули, яка обмежує їх значення між верхньою та нижньою межами. Мурахи вносять феромони лише на найкращі ребра відповідно до критерію обрання найкращого маршруту.

- Інтенсифікація та диверсифікація: MMAS використовує механізми інтенсифікації та диверсифікації для забезпечення ефективного пошуку. Інтенсифікація відбувається шляхом засилля феромонів на найкращих ребрах, що сприяє покращенню якості розв'язку. Диверсифікація здійснюється шляхом зниження феромонів на всіх ребрах, що допомагає уникнути локального мінімуму та збільшити різноманітність розв'язків.

- Ітераційний процес: Алгоритм повторюється протягом певної кількості ітерацій або до досягнення заданої умови зупинки. На кожній ітерації мурахи шукають нові розв'язки, оновлюють рівні феромонів та здійснюють інтенсифікацію та диверсифікацію.

MMAS є ефективним алгоритмом завдяки контролю феромонів та механізмам інтенсифікації та диверсифікації. Він дозволяє швидко знаходити оптимальні розв'язки, а також має стійкість до варіацій параметрів та хорошу здатність до уникнення локальних мінімумів.

Мурашиний алгоритм на основі розподілу (Ant Colony Optimization with Rank-Based Ant System) є варіацією мурашиного алгоритму, яка використовує розподіл ймовірностей для вибору наступного кроку мурахи. Цей підхід дозволяє збільшити різноманітність пошуку та уникнути пасток локального мінімуму.

Основні особливості мурашиного алгоритму на основі розподілу:

– Ранжування ребер: У цій варіації алгоритму ребра ранжуються залежно від значень феромонів та евристичних інформаційних значень. Ранжування допомагає встановити порядок перевірки ребер та впливає на розподіл ймовірностей вибору.

– Розподіл ймовірностей: За допомогою ранжування ребер визначаються ймовірності вибору кожного з них. Більш важливим ребрам, які мають високі значення феромонів або евристичних інформаційних значень, надається більша ймовірність бути обраним мурахою.

– Розподіл на основі рангів: Ймовірності вибору ребер визначаються за допомогою функції розподілу на основі рангів. Ця функція прив'язує ранги ребер до ймовірностей вибору, де більш важливим ребрам присвоюються вищі ранги та вищі ймовірності.

– Збереження різноманітності: Розподіл ймовірностей на основі рангів допомагає зберігати різноманітність пошуку. Це дозволяє мурахам розглянути більше ребер та зробити більш різноманітні вибори, що допомагає уникнути пасток локального мінімуму та забезпечити кращі розв'язки.

– Оновлення феромонів: Після кожного кроку мураха оновлює рівні феромонів на вибраних ребрах. Оновлення може відбуватись за допомогою формул, які залежать від якості розв'язку та рівнів феромонів.

Ця варіація мурашиного алгоритму на основі розподілу дозволяє збільшити різноманітність та гнучкість пошуку шляхів у задачах оптимізації. Використання розподілу ймовірностей на основі рангів допомагає мурахам вибирати шляхи з більшою об'єктивністю та зменшує ймовірність застрягання у локальних мінімумах.

Мурашиний алгоритм на основі дифузії феромонів (Ant Colony Optimization with Pheromone Diffusion) є варіацією мурашиного алгоритму, в якій феромони поширюються на ребрах графу шляхом дифузії. Цей підхід сприяє більш глобальному пошуку та розподілу феромонів у задачах оптимізації.

Основні особливості мурашиного алгоритму на основі дифузії феромонів:

– Дифузія феромонів: В цій варіації алгоритму феромони розповсюджуються на ребрах графу шляхом дифузії. Кожне ребро передає своє значення феромону в навколишні ребра залежно від встановленої функції дифузії.

– Функція дифузії: Функція дифузії визначає, як швидко та в якій мірі феромони розповсюджуються від одного ребра до іншого. Ця функція може залежати від властивостей ребер, таких як відстані, ваги, евристичних інформаційних значень тощо.

– Розподіл феромонів: Мурахи обирають наступний крок залежно від розподілу феромонів на доступних ребрах. Ймовірність вибору ребра може бути визначена на основі значень феромонів та евристичних інформаційних значень, які можуть бути модифіковані під час дифузії.

– Оновлення феромонів: Після кожного кроку мураха оновлює рівні феромонів на ребрах, по яких вона пройшла. Оновлення може залежати від якості розв'язку та рівнів феромонів на кожному ребрі.

Мурашиний алгоритм на основі дифузії феромонів сприяє більш глобальному пошуку, оскільки феромони розповсюджуються на ребрах, впливаючи на вибір мурахи у подальшому. Цей підхід дозволяє розподілити феромони ширше та ефективніше у просторі розв'язків, що сприяє знаходженню кращих розв'язків задач оптимізації.

2.3 Схема способу отримання рішення за мурашиним алгоритмом

Спосіб отримання рішення за мурашиним алгоритмом складається з наступних етапів:

1. Ініціалізація: на початку алгоритму випадково генерується початковий шлях із випадковими значеннями параметрів. Кожен мураха починає свій шлях випадковим чином.

2. Вибір наступного кроку: кожен мураха обирає наступний крок на основі ймовірності переходу від поточного стану до кожного з доступних станів.

Ймовірність переходу визначається на основі розрахунку феромонних слідів на ребрах графа.

3. Оновлення феромонних слідів: після проходження кожним мурахою шляху, феромонні сліди оновлюються на основі якості пройденого шляху. Якщо якість шляху висока, то феромонні сліди на його ребрах збільшуються, а якщо якість шляху низька, то феромонні сліди зменшуються.

4. Перевірка критеріїв закінчення: після кожного проходження всіма мурахами, проводиться перевірка критеріїв закінчення. Якщо будь-який з критеріїв виконується, то алгоритм закінчує свою роботу.

5. Вибір оптимального шляху: після закінчення роботи алгоритму, вибирається найкращий знайдений шлях як оптимальне рішення.

Мурашиний алгоритм використовується для розв'язання проблем комбінаторної оптимізації, таких як проблема швейного виробництва з обмеженнями. У цьому контексті, граф використовується для представлення рішень і їх взаємодії.

Опишемо вершини графа для мурашиного алгоритму, що можуть бути використані для проблеми швейного виробництва з обмеженнями:

Вершини-завдання (Task Nodes): Кожна вершина-завдання відповідає конкретному завданню виробництва, наприклад, пошиття певного виробу або виконання певної операції на пошиття. Кожна вершина-завдання має свої властивості, такі як тривалість виконання, пріоритетність, обмеження ресурсів і обмеження часу.

Вершини-ресурси (Resource Nodes): Кожна вершина-ресурс відповідає ресурсам, які можуть бути використані для виконання завдань. Це можуть бути робочі машини, робочі місця або матеріали. Кожна вершина-ресурс має свої властивості, такі як час доступності, кількість одиниць ресурсу, обмеження кількості використань тощо.

Вершини-вибір (Choice Nodes): Деякі завдання можуть мати декілька варіантів виконання або вибору ресурсів. Вершини-вибір використовуються для представлення цих альтернатив і визначення рішень, що приймаються.

Вершина–старт (Start Node): Це початкова вершина графа, з якої починається виконання роботи. У проблемі швейного виробництва вона може представляти початок робочого дня або початок конкретного процесу.

Вершина–фініш (Finish Node): Це кінцева вершина графа, що вказує на завершення виконання всіх завдань і виробництва. Вона може відображати кінець робочого дня або досягнення встановленої цілі.

Ці вершини графа пов'язані між собою ребрами, які відображають послідовність виконання завдань, обмеження доступності ресурсів та інші взаємодії між ними.

У мурашиному алгоритмі мурахи проходять по графу, вибираючи оптимальний шлях на основі феромонів, які вони залишають на ребрах. За допомогою цього алгоритму можна знайти найкращий розподіл завдань і ресурсів для вирішення проблеми швейного виробництва з обмеженнями.

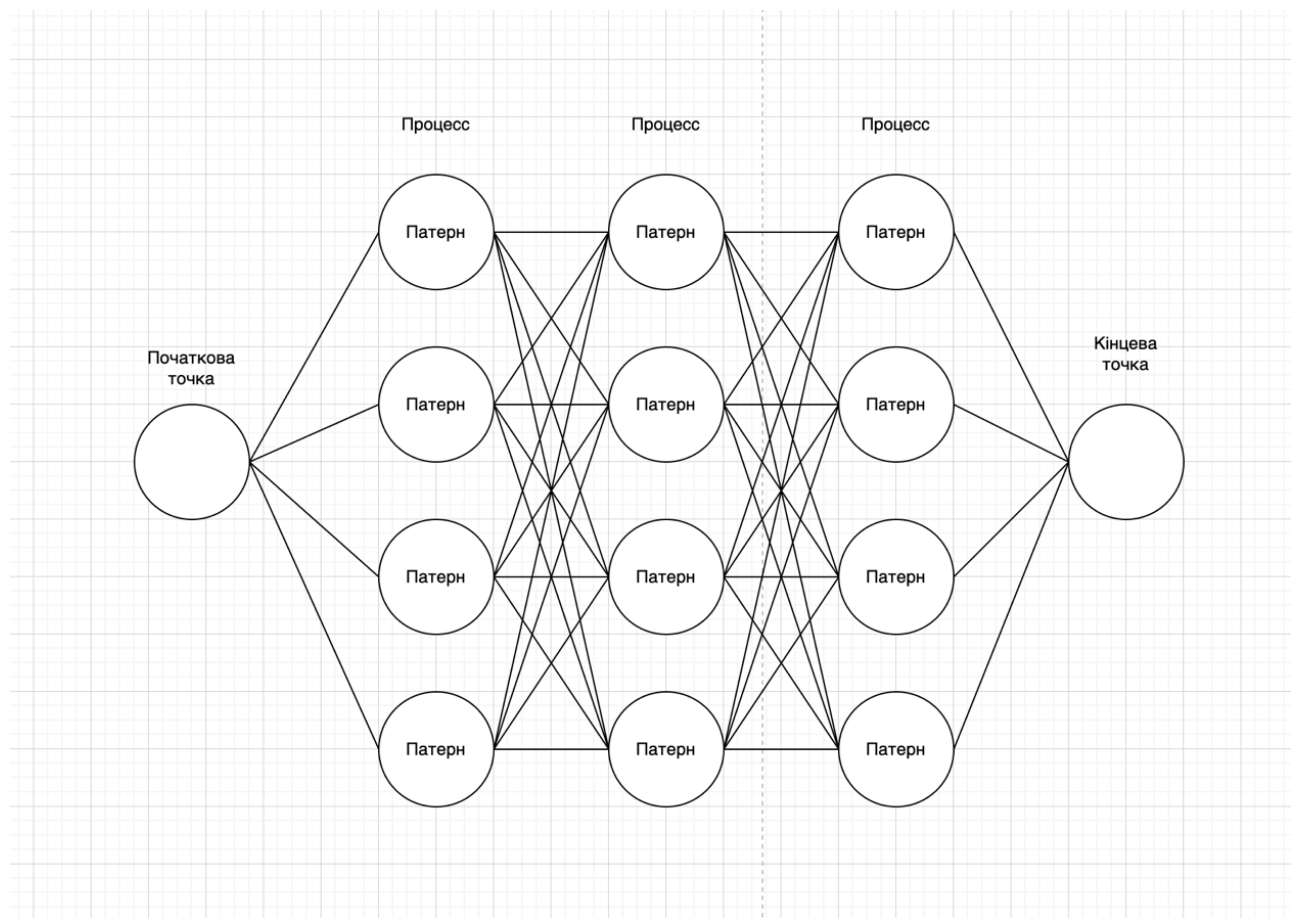


Рисунок 2.3 Граф, який використовується у мурашиному алгоритмі

Отже, схема способу отримання рішення за мурашиним алгоритмом передбачає використання випадкових значень для створення початкового шляху, вибір наступного кроку на основі розрахунку феромонних слідів, оновлення феромонних слідів та перевірку критеріїв закінчення.

В нашому випадку, ми будемо використовувати граф з типами вершин: "початок", "вибір" та "фінальний результат". Цей граф буде містити інформацію про процес, який включає в себе інструменти, матеріали, виконавця та продукт, який буде результатом роботи. Кожна вершина графа буде мати атрибути, що описують необхідну кількість матеріалів, годин виконавця, кількість інструментів та кількість продукту, який буде виготовлено на цьому етапі.

2.4 Функціональна структура інформаційної системи отримання рішення

Динаміка системи – це процес вивчення та аналізу процесу виробництва з метою отримання оптимального результату, який задовольняє виробництво з максимальною продуктивністю та ефективністю. Основна мета динаміки системи полягає в забезпеченні максимальної продуктивності та вигідності виробництва.

Для досягнення цієї мети, необхідно ретельно проаналізувати процес виробництва та визначити його найбільш оптимальний режим роботи. Для цього використовуються різні методи та інструменти, такі як математичні моделі, алгоритми оптимізації та інші.

Одним з ключових аспектів динаміки системи є максимальна продуктивність, яка означає виробництво максимальної кількості продукції в максимально короткий проміжок часу. Це досягається шляхом використання ефективних технологій та оптимального використання ресурсів.

Крім того, динаміка системи також пов'язана з максимальним прибутком, який можна отримати від виробництва продукції. Для досягнення цієї мети

необхідно використовувати стратегії маркетингу та продажів, щоб забезпечити максимальний обсяг продажів та прибуток від них.

Отже, динаміка системи полягає у забезпеченні максимальної продуктивності та ефективності виробництва з максимальним прибутком. Для досягнення цієї мети використовуються різні методи та інструменти, такі як математичні моделі, алгоритми оптимізації, технології виробництва та стратегії маркетингу та продажів.

Інтерфейс користувача, побудований за допомогою бібліотеки Swagger, є важливою складовою системи, яка дозволяє спрощено взаємодіяти користувачу з програмним забезпеченням. Swagger надає основу для документації та визначення функціональних можливостей системи.

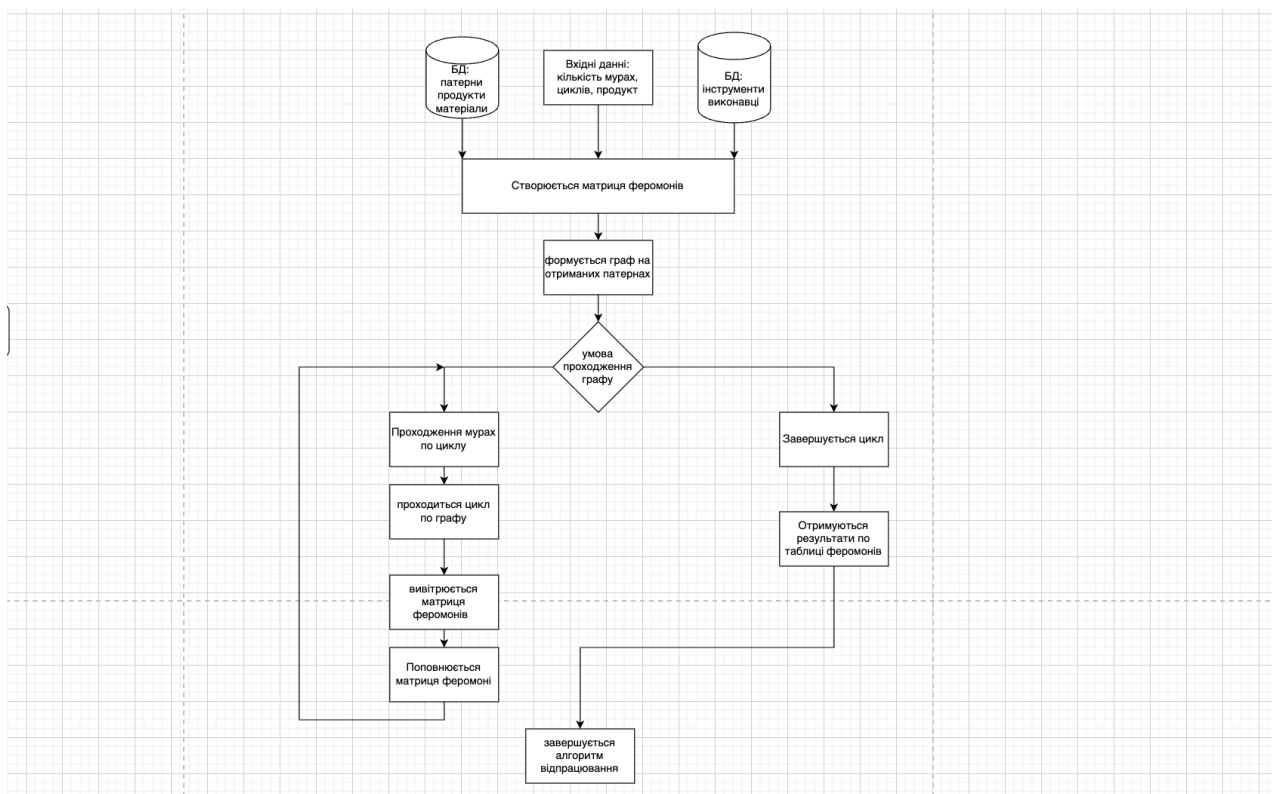


Рисунок 2.4 — Схема методу рішення для швейного виробництва за допомогою мурашиного алгоритму

Завдяки Swagger, користувач може легко взаємодіяти з системою, створюючи запити на основі потрібних даних або дій. Це означає, що користувач

може легко вказати необхідні параметри, вибрати методи і виконати дії, які він бажає здійснити.

Після відправки запиту, користувач може отримати результат своїх дій у інтерфейсі Swagger. Це дозволяє швидко отримувати зворотний зв'язок і переглядати результати виконаних запитів. Інтерфейс Swagger також може надавати корисну документацію, що допомагає користувачам розуміти можливості системи та використовувати їх ефективно.

В цілому, інтерфейс користувача, заснований на бібліотеці Swagger, створює зручну та просту у використанні платформу для взаємодії користувача з системою, що генерує запити на основі потрібних даних або дій користувача.

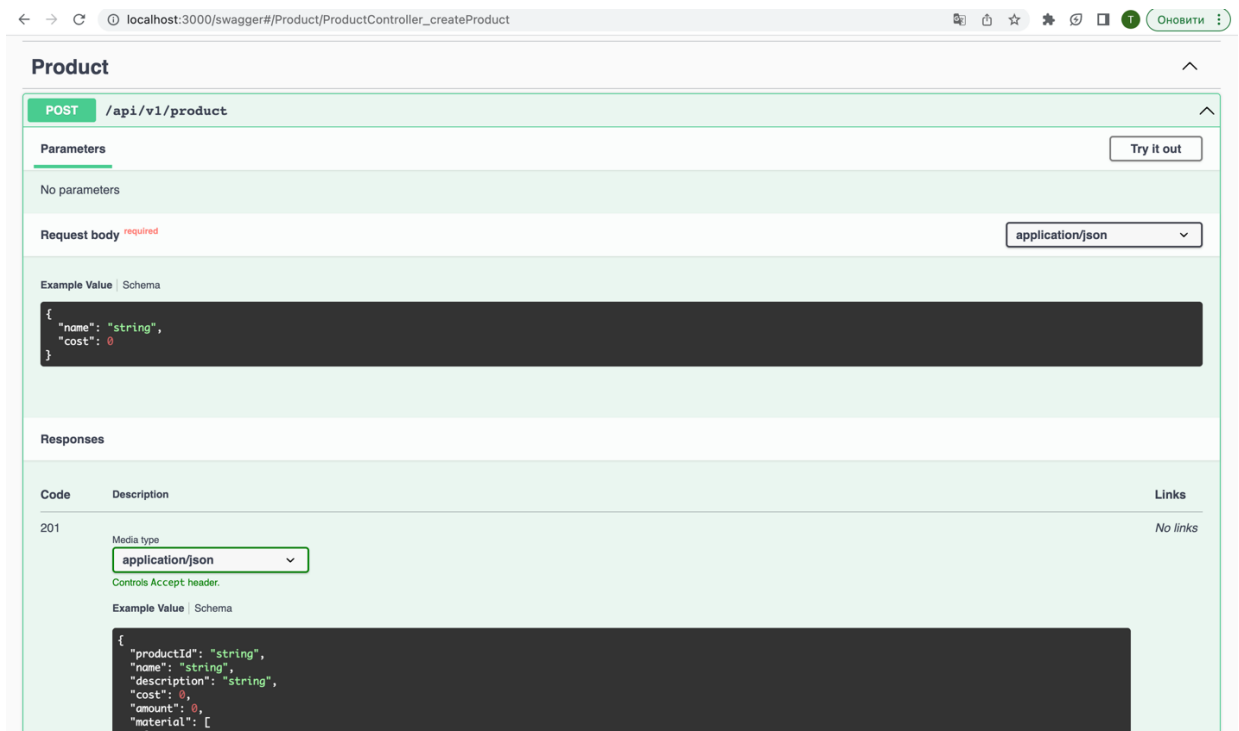


Рисунок 2.5 – Розгорнутий інтерфейс запиту створення продукту

Щоб згенерувати запит для створення продукту потрібно натиснути на функціональну кнопку “Try it out”, після цього надається можливість вписати дані у тіло запиту і запустити процес створення продукту або іншого об’єкту модулів за допомогою кнопки execute. Після цього буде наданий результат роботи.

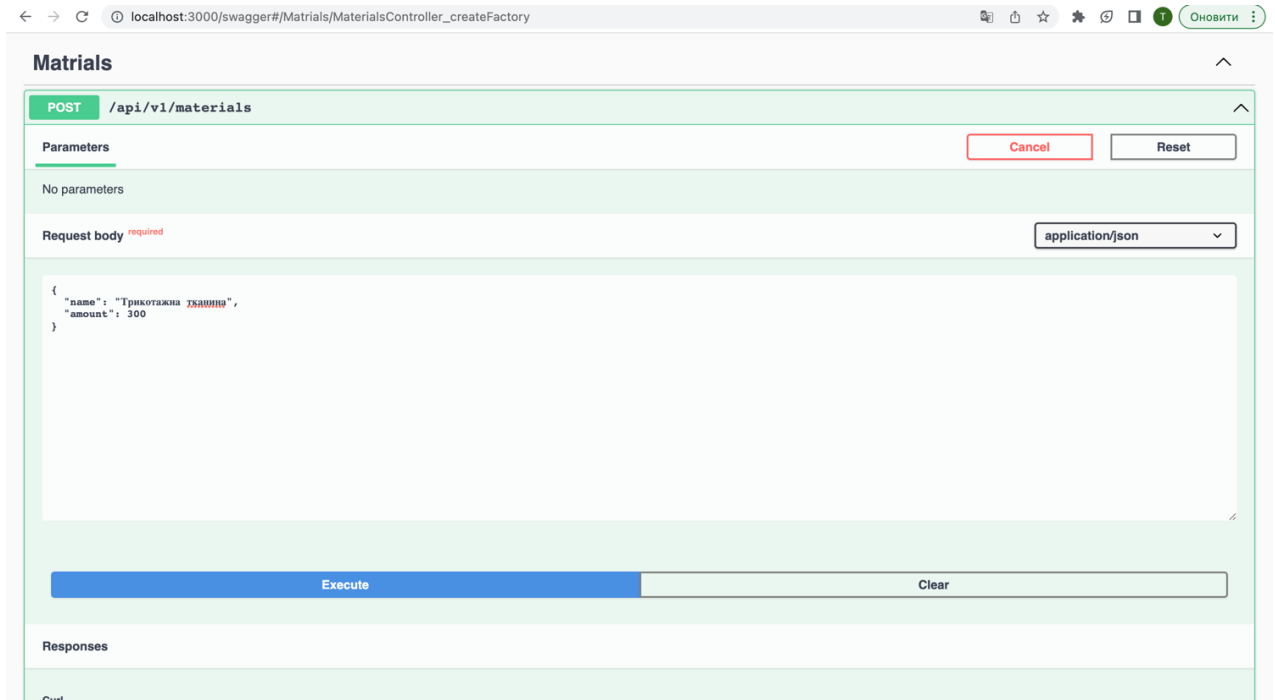


Рисунок 2.6 – Інтерфейс, готовий для надсилання запиту

Щоб згенерувати запит для створення продукту або іншого об'єкту модулів у інтерфейсі Swagger, спочатку потрібно натиснути на функціональну кнопку "Try it out". Це відкриє можливість для введення необхідних даних у тіло запиту.

Після натискання кнопки "Try it out", з'явиться можливість вписати необхідні дані у відповідні поля або параметри запиту. Наприклад, можна вказати назву продукту, його опис, ціну та інші характеристики, які необхідно передати для створення об'єкту.

Після введення необхідних даних у тіло запиту, користувач може натиснути кнопку "Execute", щоб запустити процес створення продукту або об'єкту модулів. Після виконання запиту система обробить надані дані та здійснить відповідні дії.

Після виконання запиту, результат роботи буде наданий у відповідному розділі інтерфейсу Swagger. Це може бути відображено як підтвердження успішного створення об'єкту або повідомлення про можливі проблеми, якщо такі виникнуть.

2.5 Проектування структури інформаційної системи

Реляційна база даних (РБД) є однією з найпоширеніших і доведених часом технологій для збереження і управління даними. Ось кілька причин, чому реляційна БД може бути вибором за замовчуванням:

- Структурованість даних: РБД заснована на принципах реляційної алгебри і теорії множин, що дозволяє структурувати дані у вигляді таблиць з рядками і стовпцями. Цей структурований підхід полегшує організацію, доступ і аналіз даних.

- Надійність: РБД зазвичай має вбудовану підтримку для транзакцій, забезпечуючи атомарність, консистентність, ізолюваність та стійкість (ACID-властивості). Це дозволяє гарантувати цілісність даних і уникнути проблем, таких як втрата або порушення цілісності даних.

- Запити та аналітика: Реляційні БД надають потужні механізми запитів, такі як мова структурованих запитів SQL. Це дозволяє зручно виконувати різноманітні запити до бази даних і проводити аналітику, використовуючи різні функції та операції.

- Масштабованість: Реляційні БД можуть бути масштабовані, починаючи з невеликих проектів із низьким обсягом даних до великих підприємств з великим обсягом даних. Можна розширювати обсяг даних шляхом додавання нових таблиць, індексів, використання шарування даних (шардингу).

- Нормалізація даних: Реляційна модель даних дозволяє застосовувати принципи нормалізації, що сприяє уникненню дублювання даних і забезпеченню ефективного використання простору. Це допомагає зменшити ризик втрати даних та покращити продуктивність операцій з даними.

- Екосистема та підтримка: Реляційні БД, такі як MySQL, PostgreSQL, Oracle та Microsoft SQL Server, мають широко розповсюджену екосистему, включаючи засоби розробки, інструменти адміністрування, форуми спільноти та документацію. Це забезпечує доступність підтримки та допомоги при роботі з РБД.

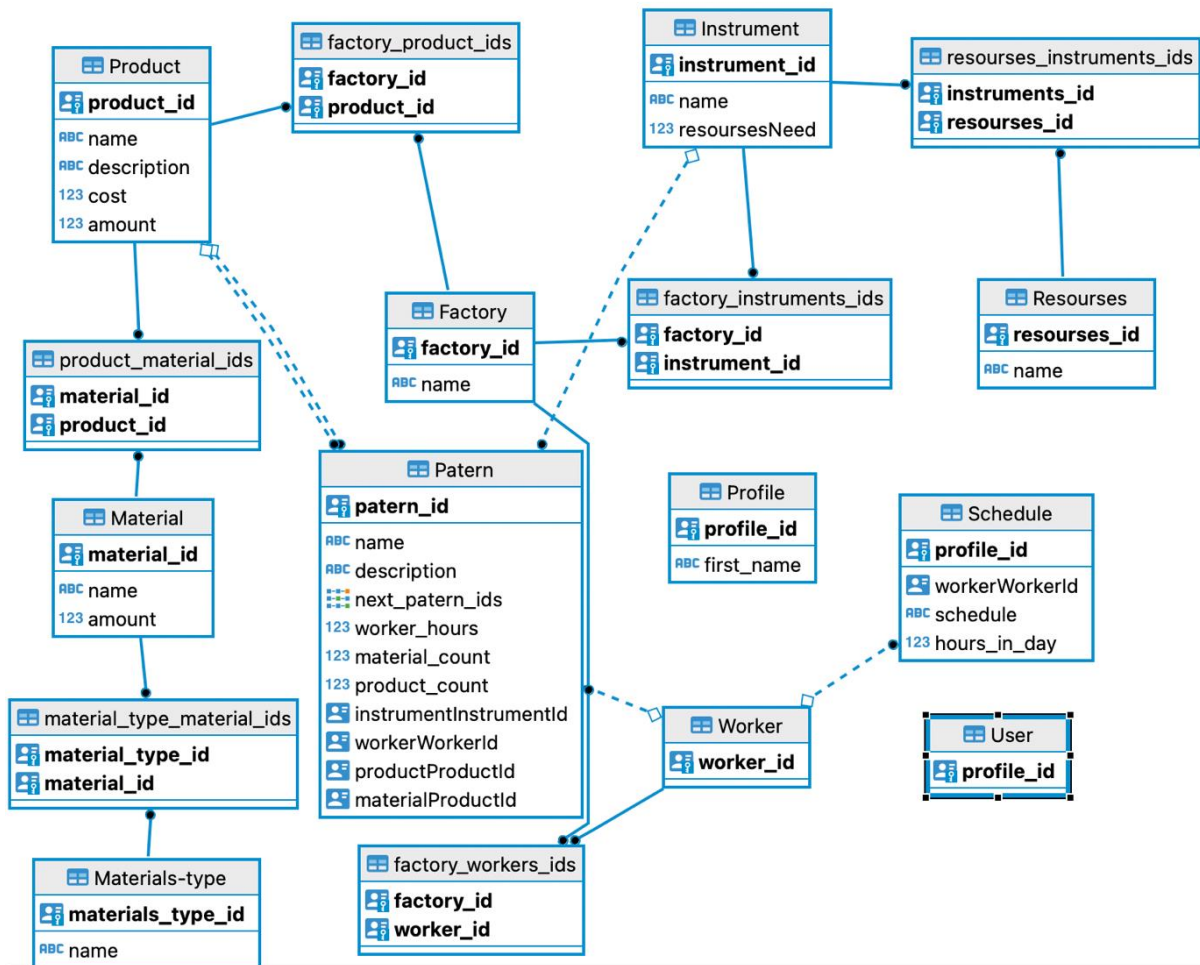


Рисунок 2.7 – Схема зв'язків БД

Таблиця Product використовується для збереження детальної інформації про товари, які можуть бути виготовлені на швейному виробництві. Кожен запис у таблиці містить назву продукту, опис продукту та зв'язок з таблицею Material, що містить інформацію про матеріали, необхідні для виробництва цього товару.

Таблиця 2.1 – Таблиця Product

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва товару
cost	Int4	Ціна товару
description	Int4	Опис товару

Таблиця Material використовується для детального опису матеріалів, які використовуються у виробництві товарів на швейному заводі. Кожен запис у таблиці містить інформацію про назву матеріалу та його потрібну кількість для виготовлення товару.

Таблиця 2.2 – Таблиця Material

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва матеріалу

Таблиця MaterialType служить для позначення наявних матеріальних запасів на складі. У цій таблиці зберігається інформація про назву матеріалу та кількість, яка доступна на складі для використання у виробництві.

Таблиця 2.3 – Таблиця MaterialType

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва матеріалу
amount	Int4	Кількість матеріалів

Таблиця Instrument містить детальний опис інструментів, які використовуються при виготовленні товарів на швейному виробництві. Кожен запис у таблиці містить назву інструменту, зв'язок з фабрикою, яка виготовляє цей інструмент, а також зв'язок з ресурсом, що використовується інструментом.

Таблиця 2.4 – Таблиця Instrument

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва інструменту

Таблиця Resources призначена для збереження інформації про ресурси, необхідні для використання різних інструментів. Кожен запис у таблиці містить назву ресурсу та значення кількості, доступної цього ресурсу щодня.

Таблиця 2.5 – Таблиця Resources

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва ресурсу

Таблиця Factory використовується для збереження детальної інформації про саме виробництво на швейній фабриці. Кожен запис у таблиці містить назву фабрики та зв'язки з таблицями Instrument, Product і Worker, які використовуються під час виробництва. Ця таблиця дозволяє встановити зв'язки між фабрикою, виробами, працівниками та інструментами.

Таблиця 2.6 – Таблиця Factory

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва підприємства

Таблиця Worker використовується для збереження інформації про виконавців, які займаються виробництвом. Кожен запис у таблиці містить назву професії працівника, який виконує роботу з використанням певного інструменту та матеріалів для створення продукту.

Таблиця 2.7 – Таблиця Worker

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
name	varchar	Назва професії

Таблиця Schedule використовується для збереження інформації про робочий графік для ефективного планування часу. Кожен запис у таблиці містить час, зв'язок з профілем робітника та зв'язок з назвою професії, що вказує на тип робітника, який має бути присутнім у даному часовому інтервалі.

Таблиця 2.8 – Таблиця Schedule

Атрибут	Тип даних	Опис
id	uuid	Ідентифікатор запису
workerId	uuid	Зв'язок з працівником

Таблиця Pattern використовується для збереження інформації про патерни, які використовуються при виготовленні продукту, містять інформацію про необхідні інструменти, матеріали для виготовлення. Кожен запис у таблиці містить інформацію про кількість потрібного матеріалу, про кількість продукції, інформацію про інструмент, а також зв'язки між матеріалами, продукцією, інструментами.

Таблиця 2.9 – Таблиця Patternory

Атрибут	Тип даних	Опис
paternId	uuid	Ідентифікатор запису
workerHours	Int4	кількість годин працівника
name	varchar	Назва патерну
description	varchar	Опис патерну
instrument	uuid	Зв'язок із інструментом
material	uuid	Зв'язок із матеріалом
worker	uuid	Зв'язок із робітником
materialCount	Int4	Кількість матеріалу
product	uuid	Зв'язок із продуктом
productCount	Int4	Кількість продукту
nextPaternIds	varchar	Зв'язок із наступними патернами

Множина функцій інформаційної технології для створення інформаційної системи автоматизованого отримання рішень для швейного виробництва в умовах обмежень за мурашиним алгоритмом включає наступні компоненти:

- Збір та обробка даних: Інформаційна система здатна збирати та обробляти різноманітні дані, пов'язані з швейним виробництвом, такі як дані про товари, матеріали, інструменти, фабрики та робітників. Ці дані можуть бути отримані з різних джерел, включаючи бази даних, датчики або зовнішні системи.

- Моделювання та оптимізація: Застосування мурашиного алгоритму для моделювання та оптимізації процесів швейного виробництва. Цей алгоритм базується на принципах колективного інтелекту мурашок, які співпрацюють для знаходження оптимальних рішень. Система може використовувати мурашиний алгоритм для вирішення завдань, таких як планування виробництва, розподіл ресурсів, оптимізація поставок матеріалів тощо.

- Розрахунок ефективності: Інформаційна система може проводити розрахунки ефективності рішень, отриманих за допомогою мурашиного алгоритму. Це включає оцінку різних показників продуктивності, таких як час виробництва, використання ресурсів, якість продукції, вартість виробництва тощо.

- Візуалізація результатів: Система може надавати візуальне відображення отриманих рішень та результатів оптимізації. Це дозволяє користувачам легко сприймати та аналізувати дані, а також зробити обґрунтовані вирішення щодо виробничих процесів.

- Інтеграція з існуючими системами: Інформаційна система може бути інтегрована з існуючими системами у швейному виробництві, такими як системи управління запасами, системи планування виробництва або системи керування якістю. Це дозволяє обмінюватися даними та координувати роботу між різними компонентами системи.

- Керування обмеженнями: Система може враховувати обмеження, які є характерними для швейного виробництва, наприклад, обмеження на

доступність матеріалів, ресурсів чи робочої сили. Вона може аналізувати та оптимізувати рішення з урахуванням цих обмежень, що допомагає досягти ефективності та оптимальності виробничих процесів.

Загальна структура інформаційної системи для автоматизованого отримання рішень для швейного виробництва може включати такі компоненти: модуль збору та обробки даних, модуль моделювання та оптимізації за допомогою мурашиного алгоритму, модуль розрахунку ефективності, модуль візуалізації результатів та модуль керування обмеженнями. Ці компоненти взаємодіють між собою для забезпечення автоматизованого та оптимального процесу прийняття рішень у швейному виробництві.

Інтерфейс користувача представляє собою бібліотеку Swagger, яка надає графічне представлення запитів для REST-API застосунку. Ця бібліотека дозволяє розробникам зручно взаємодіяти із застосунком, використовуючи документацію і готові інструменти для формування запитів і отримання відповідей.

За допомогою інтерфейсу Swagger можна отримати доступ до функціоналу, який дозволяє формувати абстрактне підприємство і знаходити максимальний прибуток. Застосунок, який побудований на основі цього інтерфейсу, дозволяє користувачеві встановлювати параметри підприємства, такі як витрати на рекламу, ціну товарів, кількість працівників тощо, і аналізувати їх вплив на прибуток.

Крім того, інтерфейс Swagger надає можливість переглядати документацію з запитом, описувати параметри запитів, перевіряти правильність введених даних та отримувати відповіді у зручному графічному вигляді. Це спрощує процес розробки, тестування та взаємодії з REST-API застосунком.

Таким чином, інтерфейс користувача на базі бібліотеки Swagger допомагає зрозуміти і використовувати можливості REST-API застосунку для формування абстрактного підприємства і досягнення максимального прибутку. Він спрощує процес взаємодії із застосунком і надає зручні інструменти для розробки, тестування та налагодження функціоналу.

Загальна структура бази даних побудована за допомогою TypeORM entity, які представляють собою таблиці і містять всі необхідні дані, що вказуються для заповнення параметрів цих таблиць. TypeORM – це розширена ORM (Object–Relational Mapping) бібліотека для TypeScript та JavaScript, яка дозволяє легко взаємодіяти з реляційними базами даних.

Кожен TypeORM entity відповідає одній таблиці в базі даних і включає в себе властивості (properties), які представляють колонки таблиці. Після визначення TypeORM entity, ми можемо використовувати його для створення, отримання, оновлення та видалення даних відповідної таблиці в базі даних. TypeORM автоматично генерує SQL–запити на основі взаємодії з TypeORM entity та виконує їх у базі даних.

Така структура бази даних, побудована з використанням TypeORM entity, дозволяє легко і зручно працювати з даними і забезпечує високий рівень абстракції між кодом і базою даних.

Розділ 3 Програмна реалізація інформаційної системи отримання рішення з використанням запропонованого способу

3.1 Структура модулів системи, їх взаємозв'язок

Розглянемо та проаналізуємо файлову структуру проекту.

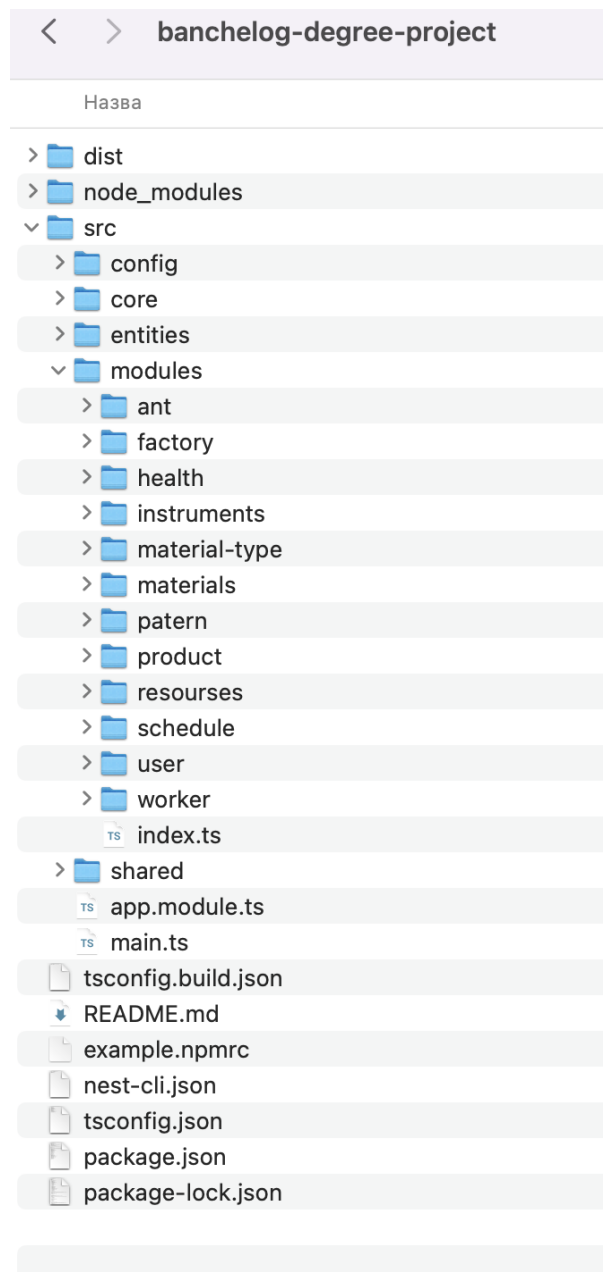


Рисунок 3.1 – Загальний вигляд модулів та файлова структура проекту

Головний модуль AppModule в NestJS є центральним модулем, який визначає основну структуру та налаштування програми. Він використовується

для імпорту, налаштування та підключення інших модулів, провайдерів та контролерів у застосунку.

Модуль Factory є ключовим компонентом, відповідальним за забезпечення роботи підприємства та координацію інших модулів, таких як Instruments, Product, Materials і Workers, Patern. Він виконує роль основного майданчика, де з'єднуються всі елементи виробничого процесу.

По–перше, модуль Factory забезпечує ефективну комунікацію між модулями. Він встановлює зв'язки та взаємодію між Модулем Instruments, що відповідає за створення необхідних інструментів, Модулем Product, який відповідає за створення продуктів, а також Модулем Materials, який відповідає за забезпечення поставок матеріалів для виготовлення продуктів. Це дозволяє забезпечити гладку роботу виробництва, де кожен модуль виконує свої функції відповідно до встановлених вимог.

По–друге, модуль Factory відіграє важливу роль у використанні робочої сили, представленої Модулем Workers. Цей модуль відповідає за створення виконавців, які використовують інструменти та матеріали для виготовлення продукту. Factory забезпечує правильну координацію роботи робітників, розподіл завдань і надання ресурсів, що дозволяє забезпечити оптимальну продуктивність і якість виробництва.

Окрім того, модуль Factory використовує мурашиний алгоритм для прорахунку процесів та вирішення оптимізаційних задач. Цей алгоритм натхненний природною поведінкою мурах, які комунікують і координують свої дії, щоб знайти найкоротший шлях до їжі. Таке застосування алгоритму допомагає Factory в оптимізації розкладу виробничих процесів, управлінні запасами та плануванні робочої сили.

Модуль Factory має головний метод алгоритму, який забезпечує обчислення максимального прибутку підприємства, використовуючи та комунікуючи із різними модулями, застосовуючи їхні методи, які потрібні для прорахування цього прибутку.

Модуль Product: Цей модуль відповідає за створення продуктів, які будуть виготовлятися на виробництві. Він включає в себе процеси проектування, розробки і виробництва продукту. Модуль Product забезпечує відповідність продукції встановленим стандартам якості, визначає технічні характеристики та вимоги до виробу, а також виконує контроль якості під час виробничого процесу.

Модуль Materials: Цей модуль відповідає за забезпечення роботи створення матеріалів, які будуть надходити для виготовлення продуктів на виробництві. Він включає у себе процеси пошуку та закупівлі необхідних матеріалів, контроль якості поставок та складського управління. Модуль Materials забезпечує наявність правильних матеріалів у потрібній кількості та якості для безперебійного виробничого процесу.

Модуль Instruments: Цей модуль відповідає за забезпечення роботи створення інструментів, за допомогою яких будуть створюватись товари із наданих продуктів. Він включає в себе розробку, виготовлення та обслуговування необхідних інструментів та обладнання. Модуль Instruments гарантує наявність функціональних інструментів, що дозволяють ефективно виконувати роботу з виготовлення продукту.

Модуль Pattern: Цей модуль відповідає за забезпечення роботи створення патернів, який являє собою шаблон, який являє собою процес в який входять найменування інструментів, скільки потрібно використати людей, яка кількість годин потрібно витратити, також які види матеріалів, а також, можливо, продукт, який потрібно обробити і з нього отримати інший продукт.

Модуль Workers: Цей модуль відповідає за забезпечення роботи створення виконавців, які будуть використовувати інструменти та матеріали для створення продукту. Він включає процеси найму, навчання, управління та мотивації робочої сили. Модуль Workers забезпечує наявність кваліфікованих та мотивованих працівників, які здатні ефективно виконувати свої обов'язки у виробничому середовищі.

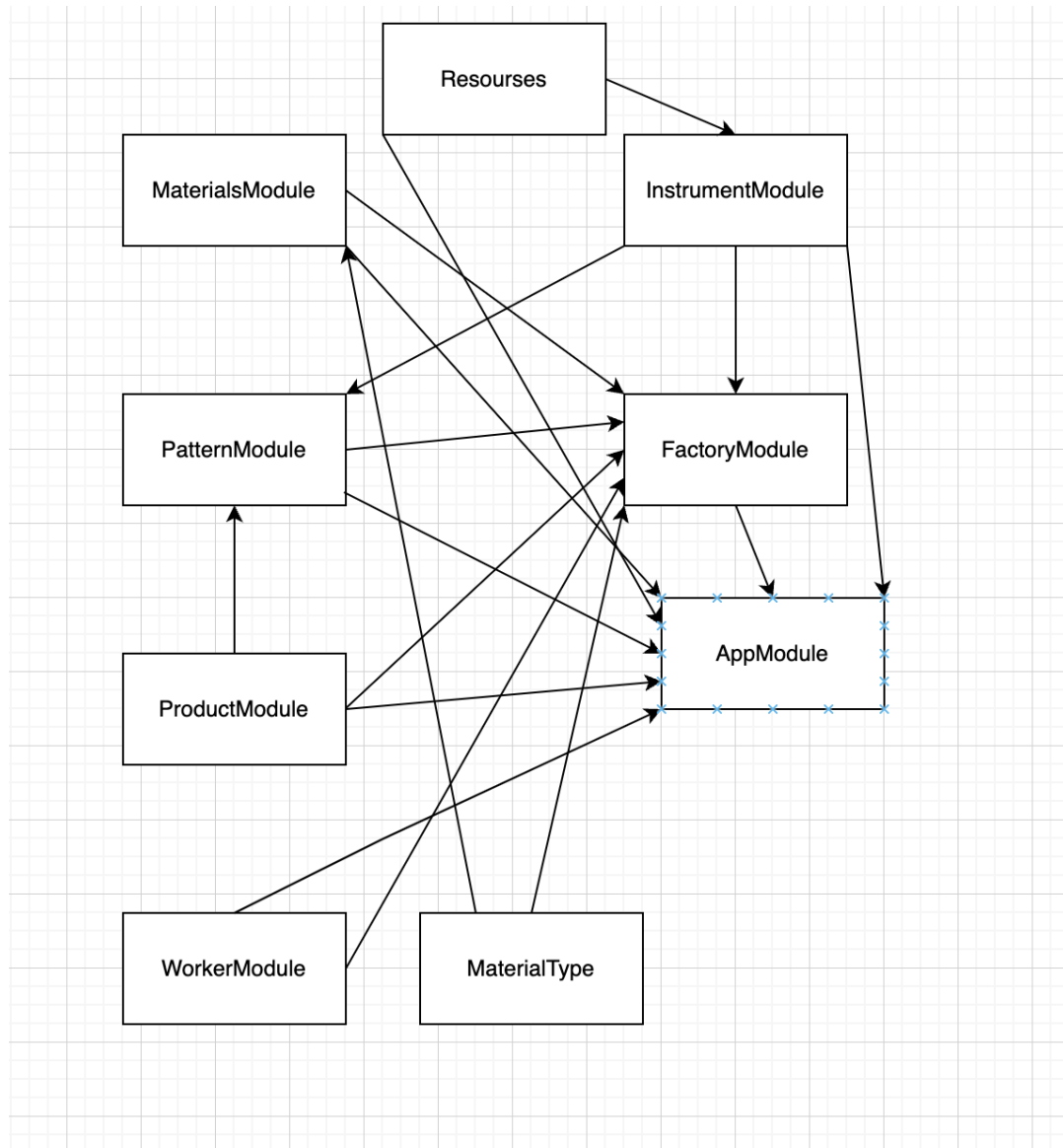


Рисунок 3.2 – Залежність модулів у програмі

Кожен з цих модулів виконує важливу функцію у виробничому процесі і співпрацює з іншими модулями, щоб забезпечити ефективну роботу модулів підприємства та створення якісних продуктів.

3.2 Особливості реалізації способу отримання рішення для швейного виробництва за мурашиним алгоритмом

Для рішення проблеми швейного виробництва в умовах обмежень за допомогою мурашиного алгоритму буде використано платформу Node.js та фреймворк Nest.js. Node.js є ефективним середовищем для розробки серверних

застосунків, а Nest.js надає потужні інструменти для створення масштабованих та модульних застосунків.

NestJS – це модульний фреймворк для побудови масштабованих та ефективних серверних додатків на Node.js. Він базується на ідеях та патернах з Angular, що робить його добре організованим, легким для розуміння та підтримки. Ось кілька плюсів використання NestJS:

Модульність: NestJS сприяє структуруванню застосунку на модулі, сервіси, контролери та постачальники. Це дозволяє легко організувати код і забезпечує принципи високої зв'язності та низької залежності. Модульність також сприяє перевикористанню компонентів та полегшує тестування.

TypeScript: NestJS написаний на TypeScript, що робить його потужним та розширюваним. TypeScript надає статичну типізацію, автодоповнення та переваги, пов'язані з сильною типізацією. Це допомагає зменшити кількість помилок під час розробки та полегшує рефакторинг.

Dependency Injection: NestJS підтримує ін'єкцію залежностей, що дозволяє легко керувати залежностями та підтримує принцип єдиного об'єкту. Це полегшує тестування та розширення застосунків.

HTTP та WebSockets: NestJS надає потужну підтримку для створення API з використанням HTTP та WebSockets. Вбудований маршрутизатор дозволяє визначати маршрути та контролери з легкістю, а декоратори роблять код більш експресивним та зрозумілим.

Патерни проектування: NestJS включає в себе декілька популярних патернів проектування, таких як Singleton, Factory та Middleware. Ці патерни допомагають забезпечити правильну організацію коду, полегшують розширення та сприяють збереженню чистоти архітектури застосунку.

Підтримка стандартів: NestJS підтримує різні стандарти, такі як OpenAPI (Swagger) для автоматичної генерації документації API, WebSocket для реального часу. Це дозволяє швидко використовувати різні інструменти та стандарти в застосунках.

Загалом, NestJS – це потужний фреймворк, який сприяє структуруванню, розширенню та ефективності розробки серверних застосунків на Node.js. Він поєднує в собі переваги TypeScript, модульності, ін'єкції залежностей та популярних патернів проектування, роблячи його привабливим вибором для будь-якого проекту.

Крім вищезгаданих технологій, також будуть використані деякі додаткові інструменти та методи для оптимізації та покращення ефективності швейного виробництва.

Один з таких інструментів – це використання асинхронного програмування. Використання асинхронного коду дозволяє ефективно обробляти багатопотокові запити та оптимізувати використання ресурсів. Node.js має вбудовану підтримку асинхронного програмування, що дозволяє створювати швидкодіючі та масштабовані застосунки.

Для полегшення взаємодії з базою даних та забезпечення зручної обробки запитів буде використана ORM (Object–Relational Mapping). ORM дозволяє працювати з базою даних, використовуючи об'єктно–орієнтований підхід, що спрощує створення, зміну та взаємодію з моделями даних. Для Nest.js доступні різноманітні ORM, такі як TypeORM, які допоможуть зручно взаємодіяти з базою даних PostgreSQL.

Однією з ключових функцій буде розподілена обробка задач. Мурашиний алгоритм дозволяє емулювати колективну поведінку мурах для вирішення оптимізаційних задач. Це означає, що можна розподілити завдання між різними робочими вузлами або процесами, що прискорить обробку даних та забезпечить більш швидкий збір результатів.

Усі ці інструменти та методи допоможуть покращити ефективність та надійність швейного виробництва, зменшити час обробки та збільшити точність результатів. Завдяки інтеграції вищезгаданих технологій та додаткових заходів, процес швейного виробництва стане більш автоматизованим, ефективним та прозорим.

Для зберігання даних про швейне виробництво буде використана реляційна база даних PostgreSQL. PostgreSQL є потужною та надійною системою керування базами даних, яка надає розширені можливості для роботи зі складними структурами даних.

Одним з важливих аспектів є валідація вхідних даних. Для цього використовується бібліотека `express-validator`, яка надає зручні інструменти для перевірки та фільтрації даних, що надходять до сервера. Завдяки цій бібліотеці можна легко контролювати та перевіряти коректність вхідних даних перед їх обробкою.

Щоб забезпечити чіткість і зрозумілість документації проекту, використовується бібліотека `swagger`. Swagger дозволяє описати API застосунку за допомогою структурованого формату, що дозволяє зрозуміти його функціональність та взаємодію з іншими системами. Завдяки документації, створеній з використанням `swagger`, розробники та інші зацікавлені сторони зможуть легко розуміти, як взаємодіяти з рішенням та використовувати його функціонал у своїх проектах.

Використання платформи Node.js, фреймворку Nest.js, реляційної бази даних PostgreSQL, бібліотеки `express-validator` та бібліотеки `swagger` дозволить ефективно вирішити проблему швейного виробництва в умовах обмежень, забезпечуючи надійну та модульну архітектуру застосунку, точну валідацію даних та зрозумілу документацію.

3.3 Опис функціональних можливостей інформаційної системи

Інформаційна система надає широкі можливості для створення абстрактної копії підприємства і використання мурашиного алгоритму для оптимізації його прибутковості. Завдяки графічному інтерфейсу програми, користувач може створити абстрактну модель підприємства, що включає в себе всі необхідні складові.

Перш за все, система дозволяє створити функціонал робітників і встановити їх характеристики, такі як навички, кваліфікація і доступні ресурси. Можна встановити розклад роботи для кожного робітника, враховуючи їхню доступність і оптимальні години роботи. Це можна зробити, використовуючи групу методів під заголовком Worker.

Далі, потрібно визначити патерни процесів обробки та перетворення матеріалів у готові товари. Це дозволяє створити ефективні та оптимізовані шляхи виробництва. Кожен процес може бути розроблений і візуалізований у графічному редакторі, що допомагає легко розуміти послідовність операцій та виявляти можливі покращення.

Система також дозволяє створити товари, які будуть виготовлятися вашим підприємством. Можна визначити характеристики товарів, їх склад і необхідні ресурси для виробництва. Це дозволяє планувати виробництво і оцінювати витрати на ресурси перед початком роботи.

Крім того, можна додавати необхідні інструменти і ресурси для виконання виробничих процесів. Це можуть бути машини, обладнання, матеріали або будь-які інші ресурси, необхідні для виробництва товарів. Потрібно вказати їхні характеристики, доступність та використовувати цю інформацію при плануванні та оптимізації виробничих процесів.

Загалом, користувач має повний контроль над створенням і керуванням різноманітними елементами підприємства. Це включає створення матеріалів, товарів, патернів, профілів робітників, робочих місць, розкладів роботи, ресурсів та інструментів. Всі ці елементи будуть використовуватися при застосуванні мурашиного алгоритму для пошуку оптимальних шляхів та максимізації прибутковості вашого підприємства.

Для початку запуску процесу мурашиного алгоритму потрібно створити та наповнити базу даних інформацією, яка буде оброблена мурашиним алгоритмом з метою отримання максимального прибутку від програми.

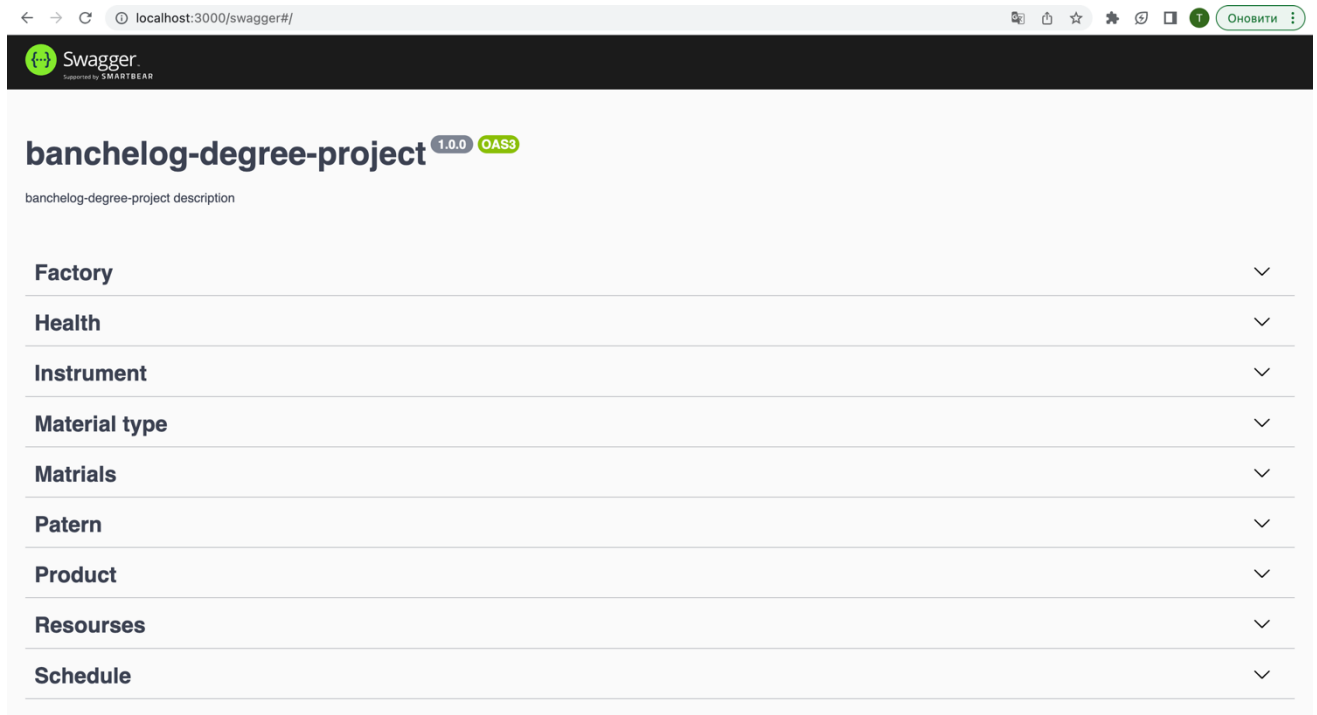


Рисунок 3.3 – Загальний вигляд модулів у графічному поданні

Першими даними, які потрібно створити, є матеріали, з яких будуть виготовлятися продукти. Для цього потрібно скористатися методом для створення об'єкту типу "material type". Таким чином, заповнюємо інформацію про сировину, яка використовується у виробництві товарів.

Щоб уточнити деталі для виготовлення конкретного товару, необхідно створити об'єкт типу "material", який вказує, скільки матеріалу потрібно на виробництво одиниці продукту. Для цього можна використовувати метод створення матеріалу ("material").

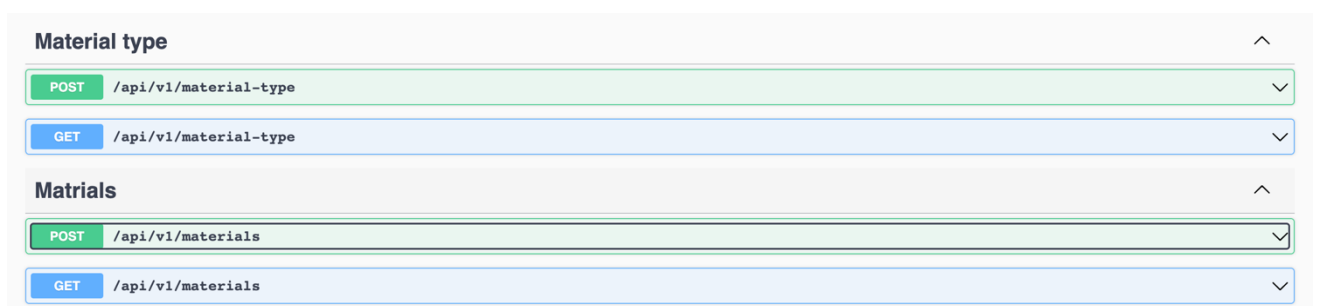


Рисунок 3.4 – Модулі Material і Material type та їх методи

Наступним кроком буде створення продукту, в якому вказуємо використаний матеріал, який був створений раніше для визначення витрат на виготовлення одиниці продукції. Для цього можна скористатися методом створення продукту у модулі "product". Таким чином, отримуємо логічний ланцюжок, що починається з матеріалів на складах, вказує, скільки матеріалу потрібно для виробництва товару, і нарешті, сам продукт, який буде виготовлений з цієї сировини.

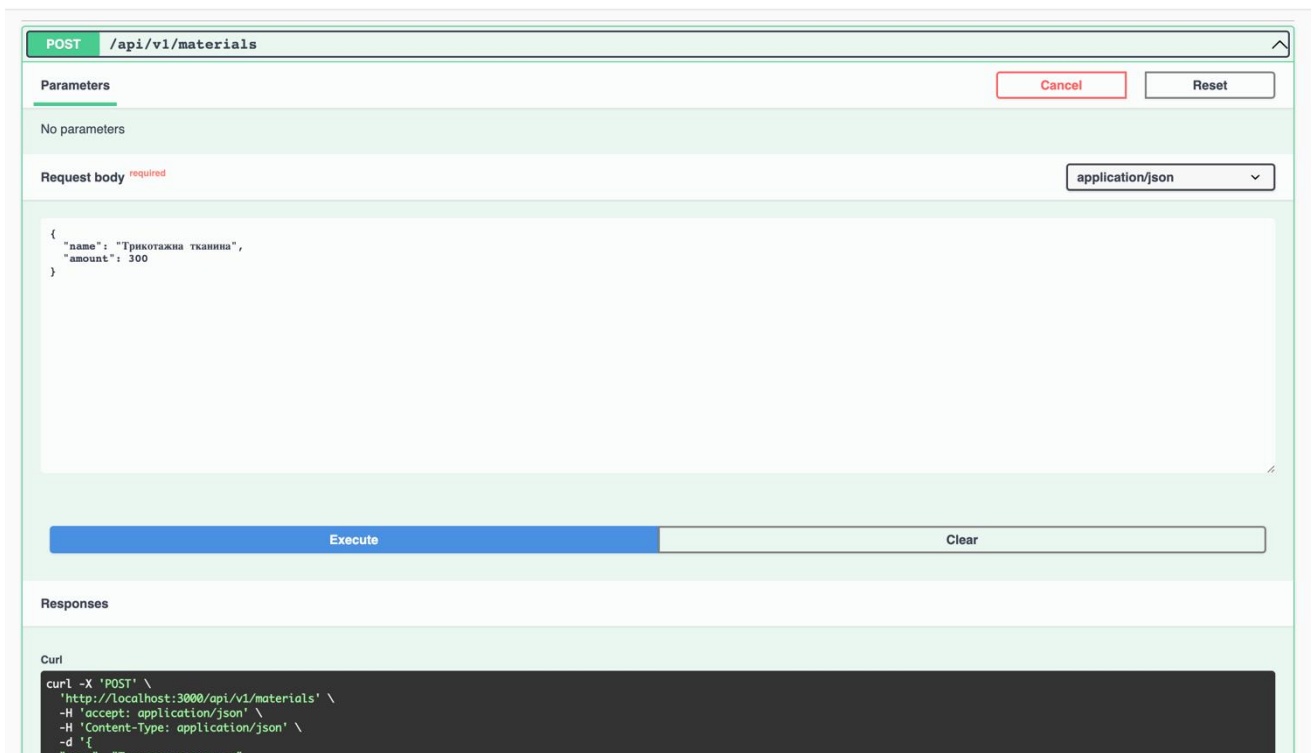


Рисунок 3.5 – Введення даних для створення матеріалу

Після цього переходимо до наступного кроку – створення інструментів та ресурсів, які використовуються цими інструментами. Для цього використовуємо модулі "Instrument" і "Resources".

Модуль "Instrument" відповідає за створення інструментів. Використовуючи відповідні методи цього модуля, можемо створити інструмент, який буде використовуватись працівниками під час процесу виготовлення товару.

За створення ресурсів, які використовують ці інструменти, відповідає модуль "Resources". Використовуючи методи цього модуля, можемо створити необхідні ресурси, які використовуватимуться разом з інструментами під час виробництва товару.

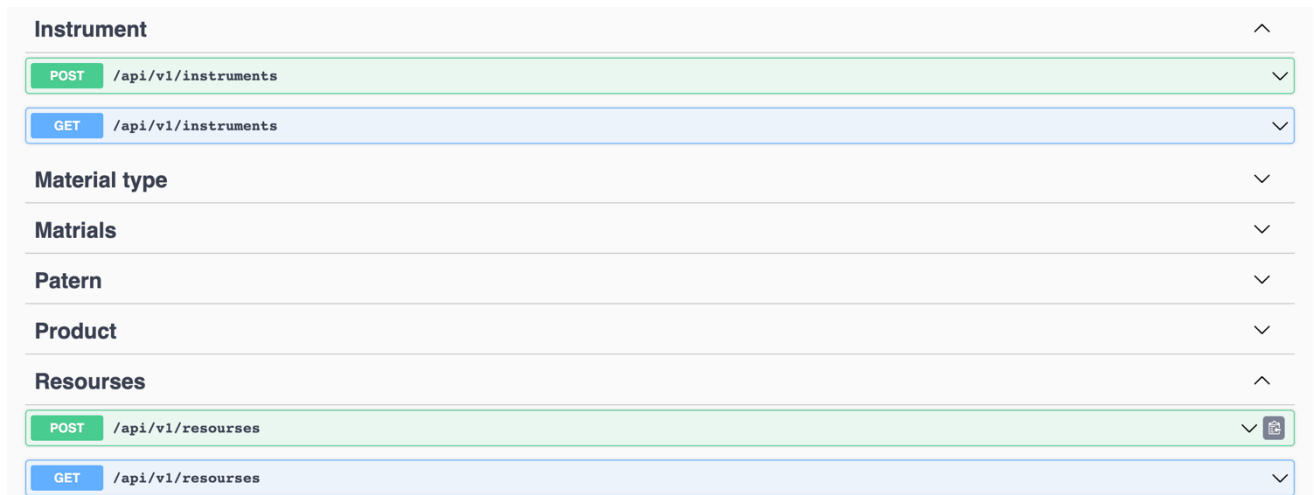


Рисунок 3.6 – Модулі Instrument та Resources

Таким чином, за допомогою цих методів і модулів буде створено інструмент, який може використовуватися працівниками при виготовленні товару, а також будуть створені необхідні ресурси для використання разом з цими інструментами.

Наступним кроком буде створення патерну, який буде використаний у мурашиному алгоритмі для знаходження максимального прибутку, який може бути врахований на основі процесів, які базуються на патернах.

Після успішного створення патерну отримується інформація про сам патерн, який може бути використаний як ланцюжок процесів, що ґрунтуються на патернах. Цей ланцюжок використовує ресурси і робочу силу для досягнення бажаного результату продукту. При використанні мурашиного алгоритму отримується максимальна продуктивність, оскільки він оптимізує розподіл ресурсів та робочої сили під час обробки. Таким чином, поєднання патернів, ланцюжка процесів, ресурсів і мурашиного алгоритму допомагає досягти оптимальних результатів і максимальної продуктивності у роботі над проектами.

Для запуску обрахування із використанням мурашиного алгоритму, необхідно вказати деякі параметри працювання алгоритму. По–перше, потрібно вказати продукцію, яку бажаємо обраховувати за допомогою цього алгоритму.

По–друге, потрібно вказати кількість мурах, які будуть проходити алгоритм в одному циклі. Це кількість мурах, які відправляються у пошуковий процес для знаходження рішення. Зазвичай вибір цього параметра залежить від розміру задачі і часу, який можна витратити на обчислення.

По–третє, слід вказати кількість циклів, які потрібно пропрацювати для знаходження оптимального варіанту. Це кількість ітерацій алгоритму, за яку він спробує знайти найкраще рішення. Вибір цього параметра також залежить від характеристик задачі і обмежень часу.

Загалом, для успішного запуску мурашиного алгоритму необхідно вказати продукцію для обрахунку, кількість мурах у циклі та кількість циклів для пошуку оптимального варіанту. Після налаштування цих параметрів алгоритм буде готовий розпочати обчислення і знайти оптимальне рішення для максимального прибутку продукту.

3.4 Вимоги до розгортання системи

Розгортання середовища для NestJS проекту включає налаштування серверного середовища, встановлення необхідного програмного забезпечення та налагодження параметрів конфігурації. Основні кроки розгортання середовища можуть бути наступними:

- Вибір серверного середовища: Спочатку вам потрібно вирішити, на якому серверному середовищі ви плануєте розгорнути NestJS проект. Це може бути локальний комп'ютер, фізичний сервер або хмарна платформа, така як Amazon Web Services (AWS), Google Cloud Platform (GCP) або Microsoft Azure.

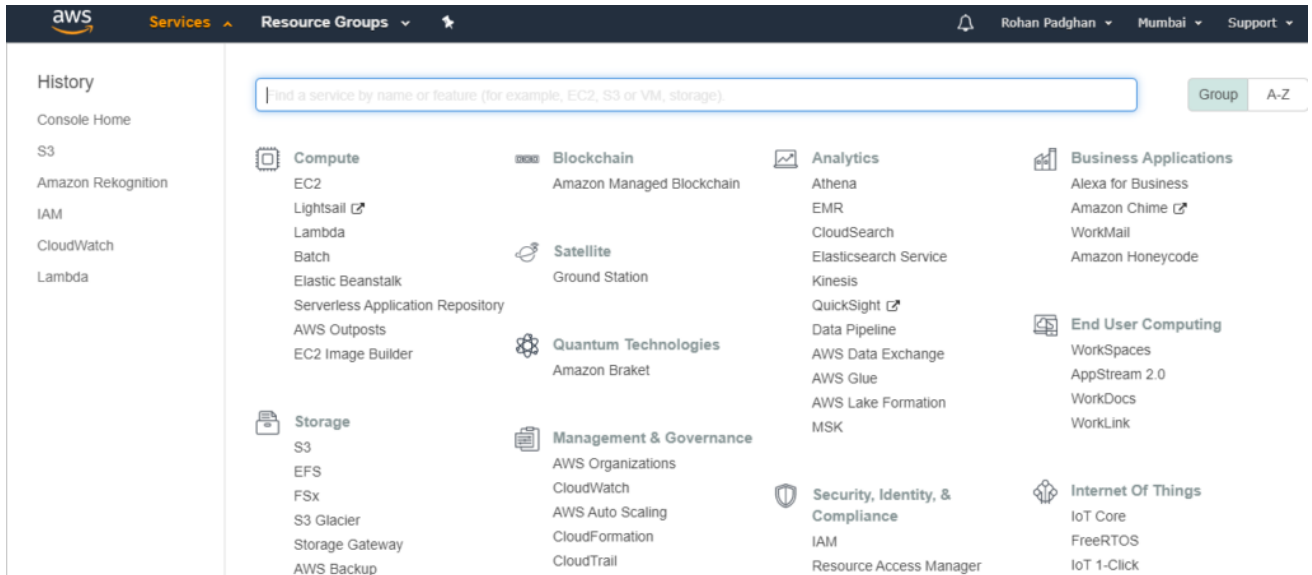


Рисунок 3.7 – Головний сайт платформи aws

– Встановлення залежностей проекту: Забезпечте, щоб всі необхідні залежності NestJS-проекту були встановлені в середовищі розгортання. Можна використовувати пакетний менеджер `npm` або `yarn` для встановлення залежностей, виконавши команду `npm install` або `yarn install`.

– Налаштування параметрів конфігурації: Переконайтеся, що всі необхідні параметри конфігурації, такі як порт сервера, параметри підключення до бази даних, ключі API тощо, налаштовані правильно для середовища розгортання. Зазвичай ці параметри знаходяться в конфігураційних файлах або змінних середовища.

– Переміщення коду проекту: Завантажте код NestJS-проекту на сервер або вибрану хмарну платформу. Це може здійснюватися шляхом клонування репозиторію Git, завантаження файлів через FTP або інші протоколи.

– Встановлення та конфігурація веб-сервера: Встановіть веб-сервер, такий як Nginx або Apache, якщо плануєте використовувати його для проксування запитів до NestJS-сервера. Налаштуйте правила маршрутизації, щоб перенаправляти запити на правильний порт, на якому працює NestJS-сервер.

– Гарантування безпеки: Зверніть увагу на гарантування безпеки середовища розгортання. Переконайтеся, що фаєрвол встановлений та

налаштований для захисту сервера, інфраструктури та даних. Рекомендується також встановити SSL–сертифікат для шифрування комунікації з сервером.

При розгортанні системи буде доступне графічне подання системи у вигляді списку запитів, які викликають методи взаємодії користувача із системою, що стало доступним за допомогою бібліотеки `swagger`.

Швидкодія NestJS-застосунку залежить від кількох факторів, включаючи обладнання, на якому він працює. Ось деякі вимоги та рекомендації для досягнення оптимальної швидкодії:

- Процесор: NestJS використовує JavaScript і запускається на платформі Node.js. Тому кращі результати можна очікувати на системах з швидким процесором. Більша кількість ядер процесора дозволяє розподілити завдання на більшу кількість потоків, що може покращити продуктивність застосунку.

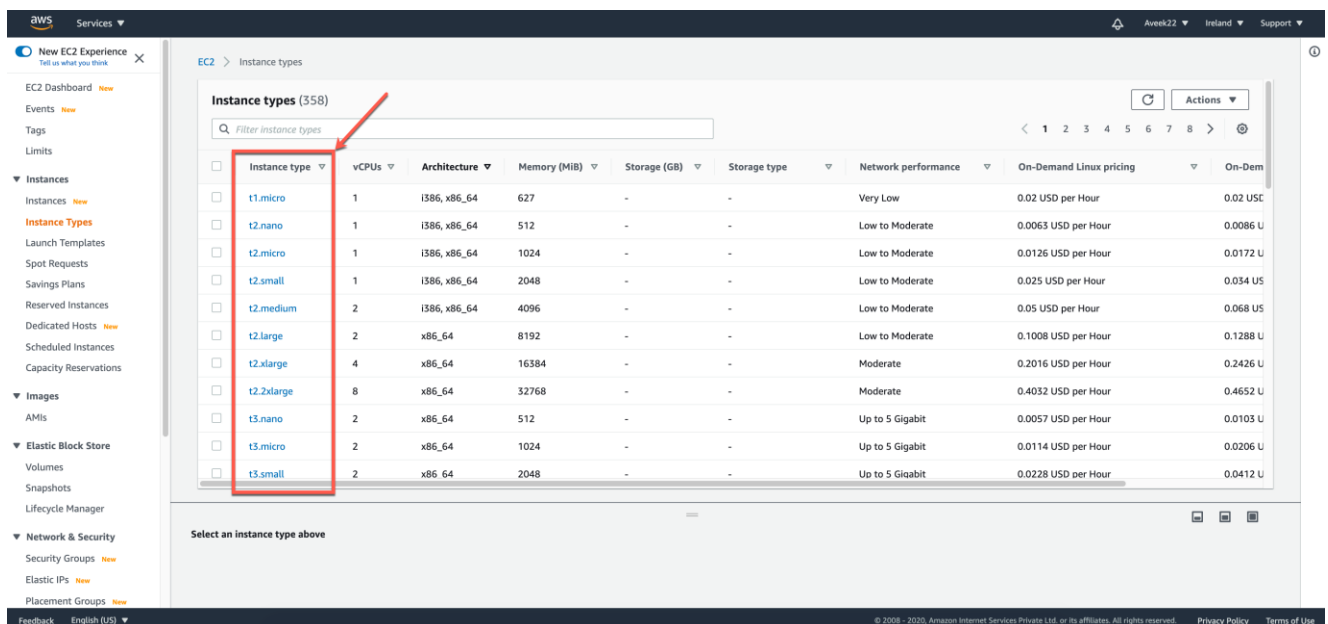


Рисунок 3.8 – Типи aws instance

- Оперативна пам'ять: Нест–застосунки використовують пам'ять для завантаження та виконання коду, а також для зберігання даних. Більша кількість оперативної пам'яті дозволяє обробляти більше запитів одночасно та зменшує імовірність виникнення проблем із пам'яттю. Рекомендується мати щонайменше

1–2 ГБ оперативної пам'яті для запуску простих застосунків, але залежно від розміру проекту та навантаження може знадобитися більше.

– База даних: Швидкодія застосунку також може залежати від швидкодії бази даних, з якою він взаємодіє. Виберіть швидку та ефективну базу даних, яка підходить для використання. Оптимізуйте запити до бази даних, використовуйте індекси та кешування, якщо необхідно, для поліпшення продуктивності.

Amazon Web Services (AWS) Instance – це віртуальний сервер, який надається в рамках хмарної інфраструктури Amazon Web Services. Він дозволяє запускати застосунки та виконувати обчислення у безпечному та масштабованому середовищі, без необхідності власного фізичного сервера.

AWS пропонує широкий вибір інстансів, кожен з яких має свої характеристики і призначений для різних видів завдань.

Основні характеристики включають:

- Тип інстансу: Різні типи для різних завдань.
- Розмір інстансу: Обсяг обчислювальних ресурсів (пам'ять, процесор, сховище).
- Операційна система: Вибір операційної системи, такої як Linux або Windows.
- Сховище: Варіанти зберігання даних, такі як EBS, S3 та EFS.
- Мережеві характеристики: Пропускна здатність, доступ до Інтернету, комунікація між інстансами.
- Безпека: Управління доступом, інтеграція з IAM, файрволи, шифрування даних.
- Масштабованість: Зміна розміру та розподілу ресурсів залежно від потреб.

AWS Instance надає гнучкість та продуктивність для виконання обчислювальних завдань у хмарному середовищі.

3.5 Проведення експериментів та дослідження роботи системи

Задля перевірки функціональності системи проведемо експеримент, в якому будемо відображати, як впливає різна кількість циклів мурашиного алгоритму на отримання результату. На вхідні параметри будуть подані середні значення малого підприємства – кількості працівників, інструментів, ресурсів, матеріалів, товарів.

Експеримент 1

Процесом буде емуляція виготовлення верхнього одягу. Обмеження будуть встановлені для ресурсів, матеріалів і робітників. Прибуток рахується у декілька етапів, кожного разу буде збільшена кількість циклів мурашиного алгоритму.

Запишемо результат роботи у таблицю і побудуємо діаграму за результатами мурашиного алгоритму.

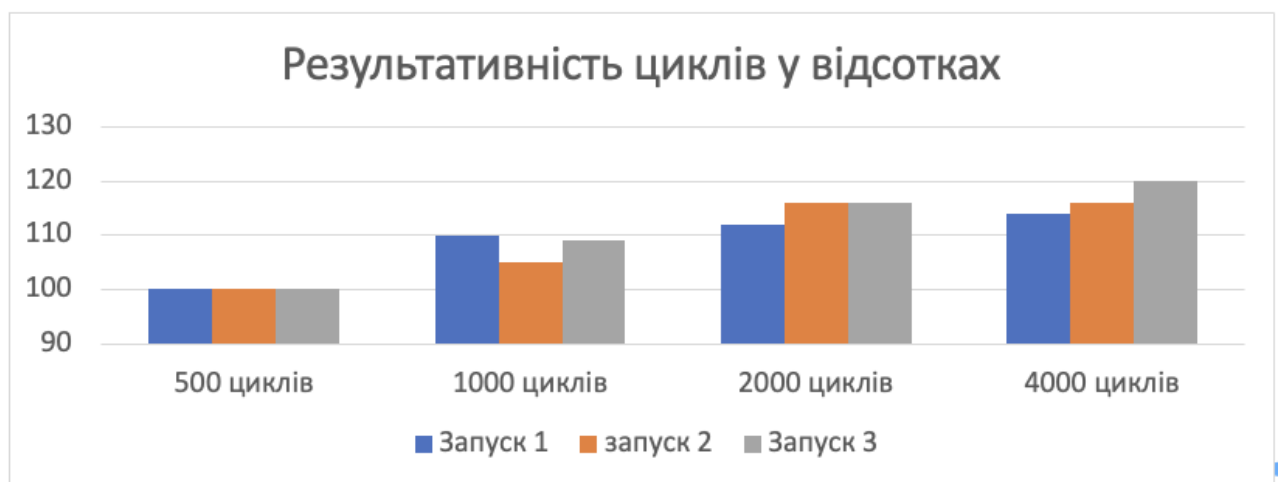


Рисунок 3.9 — Діаграма результативності мурашиного алгоритму

Таблиця 3.1 — Результат роботи програми у відсотках

Кількість циклів	Запуск 1	Запуск 2	Запуск 3

500	100	100	100
1000	110	105	109
2000	112	116	116
4000	114	116	120

За допомогою діаграми можемо зрозуміти, що результативність програми, яка використовує мурашиний алгоритм для отримання рішення, залежить від кількості циклів, які мурахи проходять, і пропорційна до отриманих результатів максимального прибутку. Значення, отримані результати після 1000 циклів, демонструють значні відмінності. Це пояснюється тим, що мурахи проходять процеси рандомно, тому лише при більшій кількості циклів вони знаходять кращі варіанти отримання максимального прибутку. При збільшенні кількості циклів максимальний прибуток буде збільшуватись. Відпрацювання кількості циклів прямо впливає на кількість часу відпрацювання алгоритму. Для забезпечення максимальної продуктивності програми за невелику частину часу потрібно використовувати потужні процесори та швидкодійну пам'ять. Такі ресурси надаються сервісом AWS. Середня кількість циклів відпрацювань, які потрібні для отримання кращого результату, ніж у інших програм, приблизно 12 хвилин. Для зменшення цього часового проміжку рекомендується використовувати краще обладнання для опрацювання.

Експеримент 2

Проведемо експеримент порівняння результативності програми. Опишемо мале швейне виробництво і прорахуємо максимальний прибуток в умовах без обмежень і в умовах обмежень. Перше обмеження буде при нестачі ресурсів, друге обмеження буде при нестачі робочої сили, а також при нестачі матеріалів. Перше значення подане без обмежень, наступні значення подані з параметром обмежень половини матеріалів, обмежень половини робочої сили, половини нестачі ресурсів від початкового результату без обмежень.

Результат запишемо у таблицю і побудуємо діаграму:

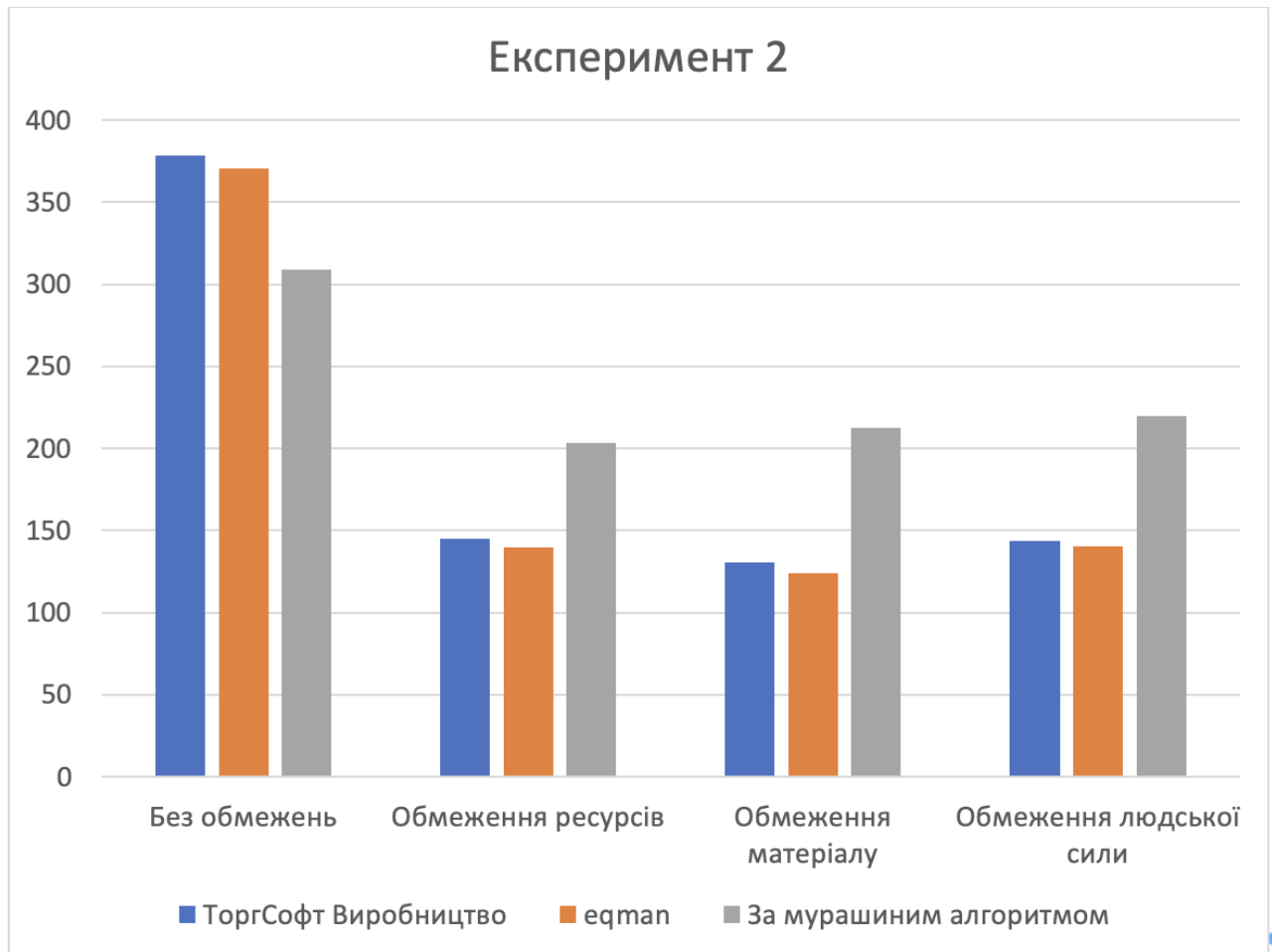


Рисунок 3.10 - Діаграма прибутку за різними програмами при обмеженнях у 50%

Таблиця 3.2 — Прибуток (тис. грн.), отриманий без умов обмежень та при наявності обмежень у 50%

Умови експерименту	ТоргСофт Виробництво	Eqman	За мурашиним алгоритмом
Без обмежень	378,7	370,4	308,8
Обмеження ресурсів	144,9	139,8	203,3
Обмеження матеріалу	130,7	124,2	212,7
Обмеження людської сили	143,8	140,7	219,8

За результатами експерименту 2 можемо зробити висновки про результати, які отримали. При використанні мурашиного алгоритму при умовах без обмежень отримуємо менший прибуток, ніж за програми, які пристосовані до розрахунку в умовах без обмежень, але при використанні мурашиного алгоритму в умовах з обмеженнями у 50%, отримуємо більший прибуток на 52-63%, ніж при використанні інших програм.

Експеримент 3

Проведемо експеримент, при якому на вхідні параметри будуть обмеження, які дають 75% ресурсів та матеріалів від необхідного. За результатом мурашиного алгоритму можемо зробити висновок, що мурашиний алгоритм ефективніший при виконання роботи з такими обмеженнями (- 25%).

Запишемо результат роботи у таблицю і побудуємо діаграму за результатами роботи програм.

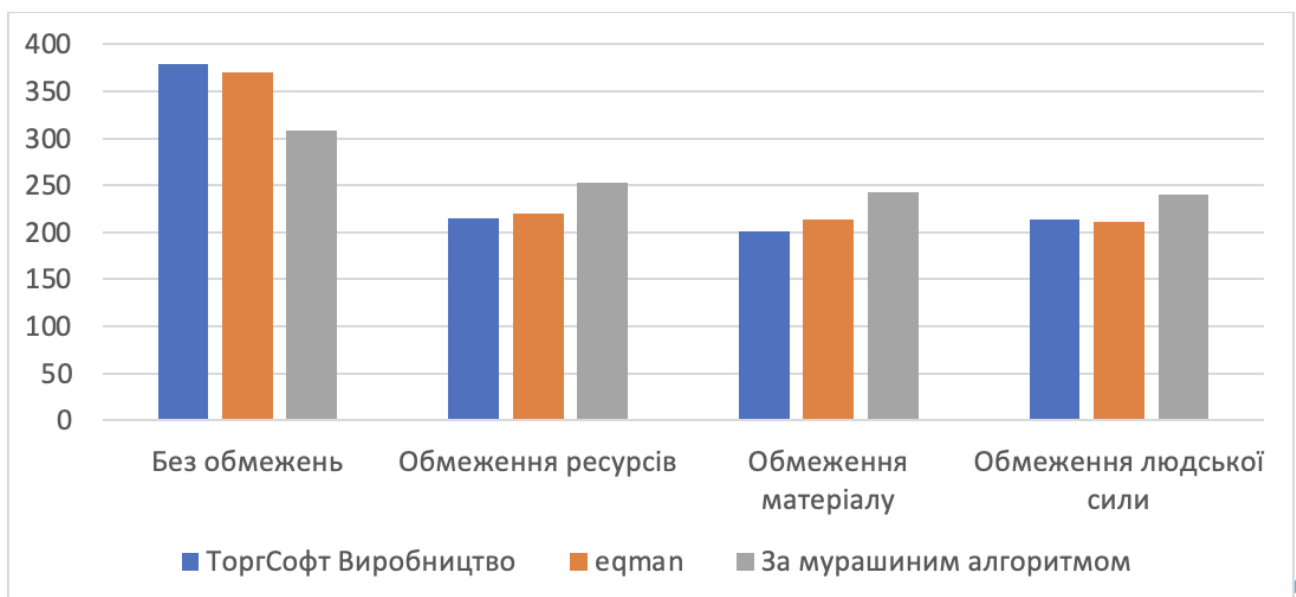


Рисунок 3.11 — Діаграма прибутку при умовах обмежень у 25%

Таблиця 3.3 — Прибуток (у тис. грн.), отриманий без умов обмежень та при наявності обмежень у 25%

Умови експерименту	ТоргСофт Виробництво	Eqman	За мурашиним алгоритмом
Без обмежень	378,7	370,4	308,8
Обмеження ресурсів	214,9	219,8	253,3
Обмеження матеріалу	200,7	214,2	242,7
Обмеження людської сили	213,8	210,7	239,8

За результатами експерименту 3 можемо зробити висновки про результати, які отримали. При використанні мурашиного алгоритму при умовах без обмежень отримуємо менший прибуток, ніж за програми, які пристосовані до розрахунку в умовах без обмежень, але при використанні мурашиного алгоритму в умовах з обмеженнями у 25%, отримуємо більший прибуток на 14-17%, ніж при використанні інших програм.

Експеримент 4

Проведемо експеримент відповідності вибору найкращого виробничого процесу із відповідними інструментами. Наприклад, у швейному цеху є три різні швейні машинки: Siruba DL7200-ВМ1-16 (потужність 900Вт, продуктивність 30 од./год.), Jack JK-A4 (потужність 1500Вт, продуктивність 36 од./год.), Juki DDL-8700N-7WB/AK85/SC920/M92/CP180 (потужність 1800Вт, продуктивність 50 од./год.).

Визначимо, яку саме машинку потрібно вибрати для отримання максимального прибутку (мінімальних витрат). Вхідними параметри на програмні застосунки є обмеження на джерела електроживлення —бензинові генератори, які надають можливість використовувати швейні машинки на виробництві. При прогнозованій відсутності електроживлення більше 3 год і при електроживленні від різних генераторів проведемо експеримент і результати запишемо в таблицю:

Таблиця 3.4 — Прогнозована кількість продукції (шт/год) машинками при обмеженні електроживлення

Спосіб електроживлення \ Програмний застосунок	ТоргСофт Виробництво	eqman	За мурашиним алгоритмом
Генератор, 2КВт	0, 0, 0	0, 0, 0	30, 0, 0
Генератор, 10КВт	0, 0, 0	0, 0, 0	30, 36, 50

За результатами аналізу таблиці робимо висновок. Спосіб за мурашиним алгоритмом в умовах обмежень електроживлення пропонує у першому випадку запускати одну машинку, у другому випадку всі три, в той час як іншими застосунками можливість роботи не передбачено.

За результатами проведених експериментів можемо сказати, що при збільшенні відсотка обмежень на основні параметри або повної відсутності одного чи кількох параметрів збільшується ефективність застосування нашої інформаційної системи.

Отже, наш спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом виявився дієвим і показав необхідність свого застосування у сучасних умовах.

Висновки

Кваліфікаційна робота бакалавра вирішує проблему отримання рішення для швейного виробництва в умовах обмежень. Проектування та реалізація відповідної інформаційної системи має на меті забезпечити ефективне та зручне управління виробництвом, використовуючи оптимальні технологічні налаштування при наявних обмеженнях на виробництво. У процесі проектування були визначені основні модулі системи, включаючи модулі інструментів, матеріалів, продукції, робочої сили, аналізу та візуалізації. Кожен з цих модулів має свою функціональність та взаємозв'язок, що дозволяє системі працювати як єдине ціле.

За результатами виконання поставленої проблеми вирішено наступні задачі:

1. Проведено аналіз предметної області та відомих підходів до автоматизованого отримання рішення для швейного виробництва в умовах обмежень.

2. Розроблено спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

3. Розроблено інформаційну технологію автоматизованого отримання рішення в умовах обмеження.

4. Спроектовано функціональну та інформаційну структуру інформаційної системи отримання рішення.

5. Розроблено програмну реалізацію інформаційної системи отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

6. Проведено функціональне та прикладне дослідження розробленої інформаційної системи для запропонованого способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом.

Запропоновані технології інформаційної системи (JavaScript, TypeScript) разом з NestJs API з бібліотекою валідації дають можливість розробити

потужний та функціональний API-застосунок для отримання рішень для швейного виробництва в умовах обмежень. Застосунок створювався на принципах REST-API, а також враховані особливості інформаційної системи швейного виробництва, що дозволяє отримати зручний та ефективний інструмент для виконання технологічних процесів.

За результатами виконання API-застосунку для отримання рішень для швейного виробництва в умовах обмежень з використанням запропонованого способу за мурашиним алгоритмом було отримано позитивні результати, що підтверджують його ефективність і корисність. Порівнюючи з відомими рішеннями, такими як ТоргСофт Виробництво і Eqman, обраний API-застосунок продемонстрував переваги в плані отримання прибутку при наявності обмежень на параметри виробництва. Це свідчить про ефективність розробленого способу на основі мурашиного алгоритму. Виграш у прибутковості отримуємо завдяки особливості функціональності розробленого способу, який передбачає можливість зміни послідовності процедур на певних етапах виробництва при наявності обмежень. Це сприяє зменшенню навантаження та залежності на ділянках виробництва, які найбільше піддаються обмеженням, і, як наслідок, забезпечує збільшення прибутків.

Отримані результати роботи підтверджують актуальність та практичну цінність розробки інформаційної системи для отримання рішень у швейному виробництві в умовах обмежень за допомогою мурашиного алгоритму. Ці результати підкреслюють переваги розробленої системи у порівнянні з відомими рішеннями та відкривають перспективи для подальшого вдосконалення та використання її у швейній промисловості.

Перелік посилань

1. Роз'яснення до Закону №2136 від 15.03.2022 року «Про організацію трудових відносин в умовах воєнного стану», з урахуванням коментарів Мінекономіки. URL:<https://appu.org.ua/main-news/rozyasnennya-do-zakonu-%E2%84%962136-vid-15-03-2022-roku-pro-organizacziyu-trudovih-vidnosin-v-umovah-vo%D1%94nnogo-stanu-z-urahuvannyam-komentariv-minekonomiki/>
2. Українське суспільство в умовах війни. 2022: Колективна монографія / С. Дембіцький, О. Злобіна, Н. Костенко та ін.; за ред. член.-кор. НАН України, д. філос. н. Є. Головахи, д. соц. н. С. Макеєва. Київ: Інститут соціології НАН України, 2022. - 410 с.
3. Проблеми та перспективи розвитку бізнесу в Україні : матеріали III Міжнародної наук.-практ. конф. молодих вчених і студентів (м. Львів, 2 березня 2023 р.) : тези доповідей. – Львів : Львівський торговельно-економічний університет, 2023. – 560 с.
4. Експрес-оцінка впливу війни на мікро-, малі та середні підприємства в Україні. Аналітичний звіт. Київ: Програма розвитку ООН в Україні, 2022, 77 с. URL: https://www.undp.org/sites/g/files/zskgke326/files/2022-10/UA_Rapid_Assessment_of_War_on_MSMEs_in_Ukraine_0.pdf
5. М. О. Куцевський, Г. С. Швець — Матеріалознавство швейного виробництва: Навч. посібник /. – К.: Видавничий дім «Кондор», 2021. – 412 с.
6. Коментар Мінекономіки до Закону України від 15 березня 2022р. «Про організацію трудових відносин в умовах воєнного стану». URL:<https://me.gov.ua/Documents/Detail?lang=uk-UA&id=7ca63edc-a033-4a4e-a29b-9157cb5a69a2&title=KomentarMinekonomikiDoZakonuUkrainiVid15-Bereznia2022-R-2136-ikhproOrganizatsiiuTrudovikhVidnosinVUmovahVonnogoStanu>
7. Беляєва Н.С. Сумісна оптимізація виробничої програми та витрат на виробництво продукції підприємствами швейної промисловості. Проблеми і перспективи розвитку підприємництва. 2014. № 2 (1). С. 56-59.

8. Довідник швейного обладнання провідних фірм: Навчальний посібник /К.І. Бондар, Т.Д. Терещенко, В.С. Дубач – Хмельницький: ТУП, 2003.– 166с.
9. Проектування гнучких технологічних процесів: лабораторний практикум з дисципліни для студентів денної та заочної форми навчання спеціальності 182 «Технології легкої промисловості» «Конструювання та технології швейних виробів» / укл. А.І.Попова. – Мукачево : МДУ, 2016. – 71 с.
10. Штука Н. Перекроїти все. Війна зруйнувала понад половину швейного бізнесу України. URL:<https://forbes.ua/inside/perekroiti-vse-viyna-zruynuvala-ponad-polovinu-shveynogo-biznesu-ukraini-yak-vin-namagaetsya-postati-z-porelu-13042022-5415>
11. Попри відключення електрики роботу не зупиняли. Куницька І. Скалярчук О. URL:<https://suspilne.media/342782-popri-vidklucenna-elektro-postacanna-robotu-ne-zupinali-direktorka-kropivnickogo-svejnogo-pidpriemstva/>
12. Виробництво легкої промисловості у довоєнний час URL:<https://ukrlegprom.org/ua/news/statystyka-vyrobnyctva-legkoji-promyslovosti-za-sichen-lyutyj-2020/>M.Dorigo, G. Di Caro, and L.M. Gambardella, “Ant algorithms for discrete optimization,” *Artificial Life*, vol. 5, no. 2, 1999.
13. Левкулич В.В. Формування витрат, пов'язаних з забезпеченням та покращенням якості продукції на швейних підприємствах. Економічні науки. Сер.: Облік і фінанси. 2012. Вип. 9 (2). С. 258-265.
14. Державна служба статистики України. URL:<http://www.ukrstat.gov.ua/>
15. Українська асоціація підприємств легкої промисловості. URL:<https://ukrlegprom.org/ua/>
16. Дорожня карта розвитку галузей виробництва одягу та взуття / Українська асоціація підприємств легкої промисловості: URL:<https://ukrlegprom.org/files/ukrlegprom-roadmap.pdf>
17. Легка промисловість України. URL:<http://zno.academia.in.ua/mod/book/view.php?id=2524&chapterid=580>

18. Обладнання швейного виробництва: Навчально-методичний посібник . Хоменко Л.М. –Умань: ВПЦ «Візаві», 2011. -132 с.
19. Технологічне обладнання швейного виробництва: Навчально – методичний посібник для самостійної роботи студентів/ уклад. Людмила Миколаївна Хоменко. – Умань: ФОП Жовтий О.О., 2016. – 134 с.
20. Технологічне обладнання галузі (швейне виробництво). Б.В. Орловський, Н. С. Абрінова. URL: <https://docplayer.net/71208893-B-v-orlovskiy-n-s-abrinova-tehnologichne-obladnannya-galuzi-shveyne-virobnictvo.html>
21. Буханцова Л., Привала В. Процеси виготовлення легкого плечового одягу. Кондор. 2016р. -308с. URL: <https://www.yakaboo.ua/ua/procesi-vigotovlennja-legkogo-plechovogo-odjagu-navchal-nij-posibnik.html>
22. Проектування технологічних процесів швейного підприємства. Валентина Горобчишина, Людмила Буханцова. Кондор. 2016р. - 276с. URL:<https://mybook.biz.ua/ua/legka-promislovist/proektuvannya-tehnologichnih-procesiv-shveyynogo-virobnictva-navchalniy-posibnik/>
23. Hitoshi K., Junichi O. Solving Time–Dependent Traveling Salesman Problems Using Ant Colony Optimization Based on Predicted Traffic // Advances in Intelligent and Soft Computing. 2012. URL:https://www.researchgate.net/publication/266010119_Solving_Time-Dependent_Traveling_Salesman_Problems_Using_Ant_Colony_Optimization_Based_on_Predicted_Traffic
24. M.Dorigo, G. Di Caro, and L.M. Gambardella, “Ant algorithms for discrete optimization,” Artificial Life, vol. 5, no. 2, 1999.
25. Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others. URL: <https://www.altexsoft.com/blog/business/comparing-database-management-systemsmysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>
26. Bhattacharya P. Roadmap–based path planning– Using theVoronoi diagram for a clearance–based shortest path / P.Bhattacharya, M. L. Gavriloa // Robotics & Automation Magazine, IEEE. 2008. №15 (2).

URL:https://www.researchgate.net/publication/3344960_Roadmap-Based_Path_Planning_-_Using_the_Voronoi_Diagram_for_a_Clearance-Based_Shortest_Path

27. Pan P. Improved Ant Colony Algorithm for Path Planning of Soccer Robot // Journal of Convergence Information Technolog. 2013.№8(7).

URL:https://www.researchgate.net/publication/324948998_Mobile_robot_path_planning_using_an_improved_ant_colony_optimization

28. Stutzle Th., Hoos H. H. Max–min ant system. Future Generation Computer Systems. 2000. Vol. 16 (8). P. 889–914.

ДОДАТКИ

Додаток А

Фрагмент коду: Функція обрахування феромонів мурашиного алгоритму

```
public async calculatePaternFlow(  
  patern: Patern,  
  pheromoneArray: number[][],  
  attempt: number,  
) : Promise<number> {  
  const productCount = await this.calculatePaternProcessNode(  
    patern.paternId,  
    patern.material.productId,  
    patern.material.amount,  
  );  
  if (patern.nextPaternIds && patern.nextPaternIds.length) {  
    const length = patern.nextPaternIds.length;  
    const rand = Math.random();  
    let acc = 0;  
    let i = 0;  
    while (true) {  
      if (i > length) break;  
      if (acc > rand) break;  
      acc += pheromoneArray[attempt][i];  
      i++;  
    }  
    const nextPatern = await this.paternService.getPaternById(  
      patern.nextPaternIds[i],  
    );  
    return await this.calculatePaternFlow(  
      nextPatern,  
      pheromoneArray,  
      attempt,  
    );  
  }  
  return productCount * patern.product.cost || 1;  
}
```

Фрагмент коду: Модуль обрахування вершин по патернах мурашиного алгоритму

```
public async calculateAnt({
  ants,
  cycles,
}: StartCalculateDto): Promise<ResponseCalculateDto[]> {
  const patterns = await this.patternService.getAllPatterns();
  const pheromoneArray = this.createPheromoneArray(patterns);
  const response = await Promise.all(
    patterns.map(async (pattern) => {
      if (!pattern.material) return;
      let result = 0;
      for (let index = 0; index < ants; index++) {
        const cost = await this.calculatePatternFlow(
          pattern,
          pheromoneArray,
          0,
        );
        if (result < cost) {
          result = cost;
        }
      }
      return { result, product: pattern.product };
    }),
  );
  return response;
}
```

Фрагмент коду: Функція обрахування результату шляху мурахою по вершинах

```
public async calculatePaternProcessNode(
  paternId: string,
  materialId: string,
  materialsCount: number,
): Promise<number> {
  const patern = await this.paternService.getPaternById(paternId);
  const { instrument, material } = patern;
  const createdProductByInstrument = instrument.resoursesNeed;
  const createdProductByMaterial = patern.materialCount / material.amount;
  const maxProduct = Math.max(
    createdProductByInstrument,
    createdProductByMaterial,
  );
  return patern.productCount * maxProduct;
}
```

Фрагмент коду: модуль для відпрацювання мурашиного алгоритму

```
import { Module } from '@nestjs/common';
import { FactoryController } from './factory.controller';
import { FactoryService } from './factory.service';
import { Factory } from 'src/entities';
import { TypeOrmModule } from '@nestjs/typeorm';
import { PaternModule } from '../patern/patern.module';
import { ProductModule } from '../product/product.module';
import { MaterialsModule } from '../materials/materials.module';
import { InstrumentsModule } from '../instruments/instruments.module';
import { WorkerModule } from '../worker/worker.module';

@Module({
  imports: [
    TypeOrmModule.forFeature([Factory]),
```

```

    PaternModule,
    ProductModule,
    MaterialsModule,
    InstrumentsModule,
    WorkerModule,
  ],
  controllers: [FactoryController],
  providers: [FactoryService],
})
export class FactoryModule {}

```

Фрагмент коду: контролер який відповідає за відпрацювання запитів операцювання мурашиного алгоритму

```

import { Body, Controller, Get, Post } from '@nestjs/common';
import { FactoryService } from './factory.service';
import { Factory } from 'src/entities';
import { CreateFactoryDto } from './dto/create-factory.dto';
import { ApiTags } from '@nestjs/swagger';
import { CalculateFactoryDto } from './dto/calculate-factory.dto';

@ApiTags('Factory')
@Controller('factory')
export class FactoryController {
  constructor(private readonly factoryService: FactoryService) {}

  @Post()
  async createFactory(
    @Body() createFactoryDto: CreateFactoryDto,
  ): Promise<Factory> {
    return await this.factoryService.createFactory(createFactoryDto);
  }

  @Get()
  async getAllFactory(): Promise<Factory[]> {
    return await this.factoryService.getAllFactory();
  }

  @Post('calculate')
  async calculateAnt(@Body() startCalculateDto: CalculateFactoryDto) {
    return await this.factoryService.calculateMock(startCalculateDto);
  }
}

```

Додаток Б
Порівняльна таблиця експериментів

Назва експерименту	Обмеження	Використаний застосунок	Результат
Експеримент опрацювання швейного виробництва в умовах обмежень у 50%	Ресурсів	Мурашиний алгоритм	203,3
		eqman	139,8
		ТоргСофт виробництво	144,9
	Матеріали	Мурашиний алгоритм	212,7
		eqman	124,2
		ТоргСофт виробництво	130,7
	Робоча сила	Мурашиний алгоритм	219,8
		eqman	140,7
		ТоргСофт виробництво	143,8
Експеримент опрацювання швейного виробництва в умовах обмежень у 25%	Ресурсів	Мурашиний алгоритм	253,3
		eqman	219,8
		ТоргСофт виробництво	214,9
	Матеріали	Мурашиний алгоритм	253,3
		eqman	219,8
		ТоргСофт виробництво	214,9
	Робоча сила	Мурашиний алгоритм	239,8
		eqman	210,7
		ТоргСофт виробництво	213,8

Додаток В

Презентаційний матеріал

Кваліфікаційна робота бакалавра

СПОСІБ ОТРИМАННЯ РІШЕННЯ ДЛЯ ШВЕЙНОГО ВИРОБНИЦТВА В УМОВАХ ОБМЕЖЕНЬ ЗА МУРАШИНИМ АЛГОРИТМОМ

Виконав:
студент 3 курсу, група КНс-20-1
Пасічник Тимофій Віталійович

Керівник:
к. ф.-м.н., доцент кафедри КН
Міхалевський Віталій Цезарійович

Мета роботи

Мета кваліфікаційної роботи бакалавра полягає в розробці рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом та інформаційної системи отримання рішення, які необхідні для автоматизованого отримання рішення за відомостями множини параметрів виробництва та множини обмежень.

Для досягнення поставленої мети в розробці рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом потрібно розв'язати наступні задачі досліджень:

- Провести аналіз предметної області та відомих підходів в розробці рішень для швейного виробництва
- Вдосконалити інформаційну модель швейного виробництва в умовах обмежень
- Розробити метод отримання рішення за допомогою мурашиного алгоритму
- Розробити інформаційну технологію швейного виробництва в умовах обмежень
- Розробити інформаційну систему швейного виробництва в умовах обмежень

Також слід провести функціональне та прикладне дослідження ефективності розроблених засобів

Інформаційна модель швейного виробництва в умовах обмежень

Інформаційна модель швейного виробництва в умовах обмежень включає в себе:

- виконавці
- розклад робіт
- інструменти
- ресурси
- матеріали
- товари
- процеси

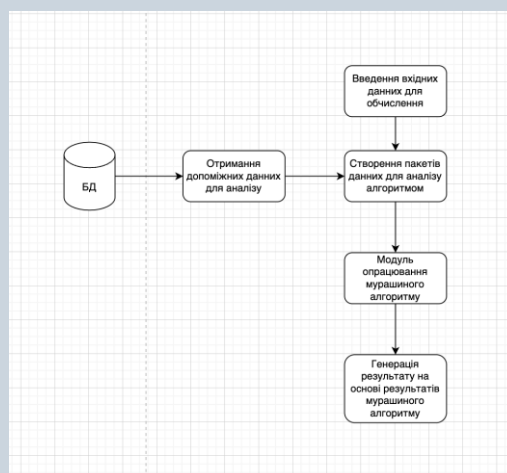


Схема методу рішення для швейного виробництва в умовах обмежень

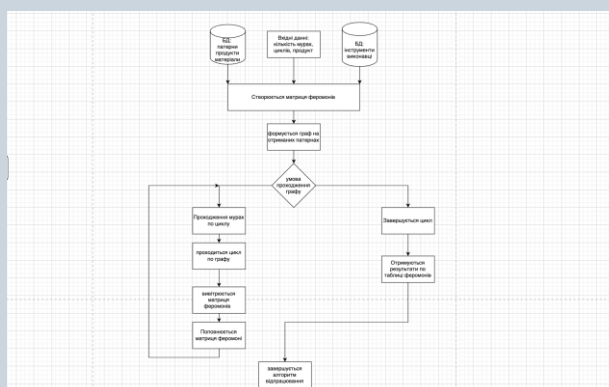
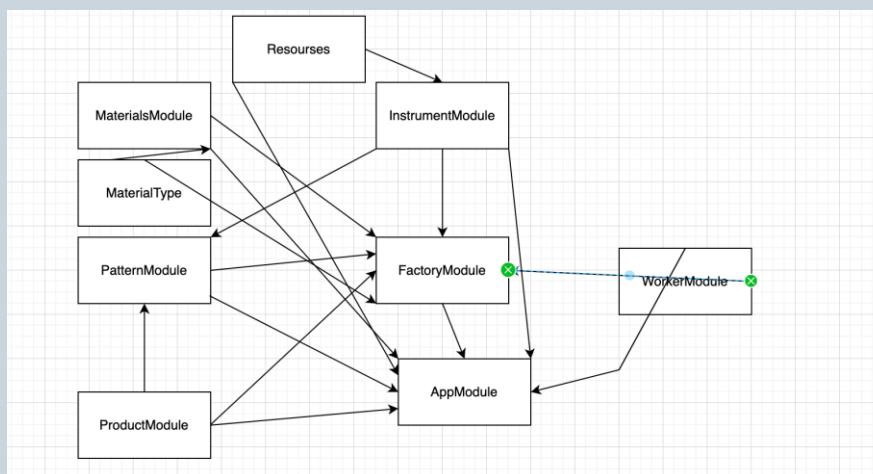


Схема взаємодії функціональних модулів інформаційної системи швейного виробництва

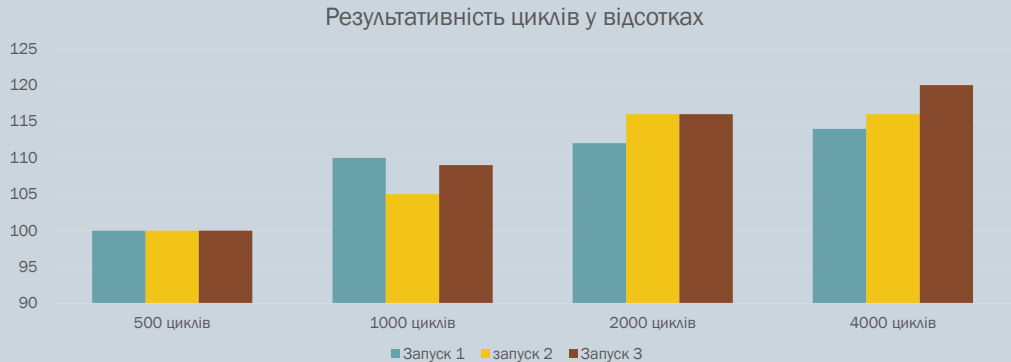


Експеримент 1.

На вхідні параметри будуть подані середні значення малого підприємства кількості працівників, інструментів, ресурсів, матеріалів, товарів. Процесом буде емуляція виготовлення верхнього одягу. Обмеження будуть встановлені для ресурсів, матеріалів і робітників. Прибуток рахується у декілька етапів, кожного етапу буде збільшена кількість циклів мурашиного алгоритму

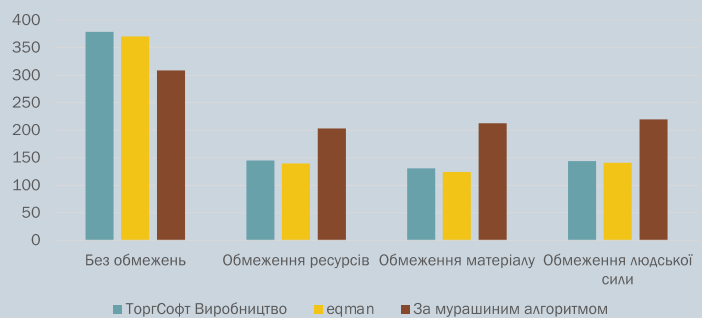
Кількість циклів	Запуск 1	Запуск 2	Запуск 3
500	100	100	100
1000	110	105	109
2000	112	116	116
4000	114	116	120

Відсотки результативності програми по кількості циклів мурашиного алгоритму



Експеримент 2

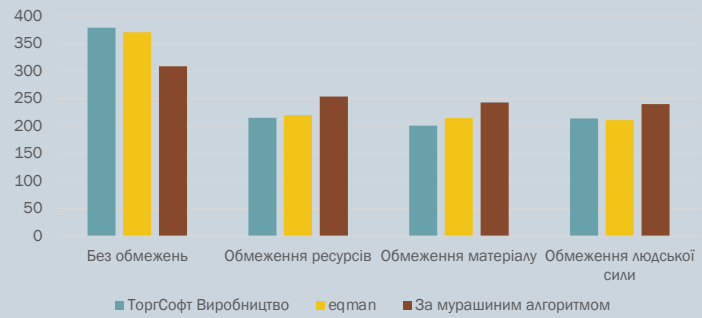
На графіку відображається результат програми за мурашиним алгоритмом, програми eqman, а також програми Виробництво сервісу Торгсофт. На вхідні параметри були подані обмеження ресурсів, матеріалу, людської сили, в половину менше без обмежень. По результатам можна зробити висновок що мурашиний алгоритм ефективніший при використанні з обмеженнями ніж використанні його без обмежень. Результати подані у тис. грн



	ТоргСофт Виробництво	eqman	Мурашиний алгоритм
Без обмежень	378,7	370,4	308,8
Обмеження ресурсів	144,9	139,8	203,3
Обмеження матеріалу	130,7	124,2	212,7
Обмеження людської сили	143,8	140,7	219,8

Експеримент 3

Проведемо експеримент при якому на вхідні параметри будуть обмеження які тільки на 75% мають ресурси та матеріали в порівнянні без обмежень. По результатам мурашиного алгоритму можемо зробити висновок що мурашиний алгоритм ефективніший при використанні навіть із незначними обмеженнями в 25%.



	ТоргСофт Виробництво	eqman	За мурашиним алгоритмом
Без обмежень	378,7	370,4	308,8
Обмеження ресурсів	214,9	219,8	253,3
Обмеження матеріалу	200,7	214,2	242,7
Обмеження людської сили	213,8	210,7	239,8

Експеримент 4

Проведемо експеримент відповідності вибору найкращого виробничого процесу із відповідними інструментами. Наприклад, у швейному цеху є три різні швейні машинки: Siruba DL7200-BM1-16 (потужність 900Вт, продуктивність 30 од./год.), Jack JK-A4 (потужність 1500Вт, продуктивність 36 од./год.), Juki DDL-8700N-7WB/AK85/SC920/M92/CP180 (потужність 1800Вт, продуктивність 2,4КВт 50од./год.).

Визначимо, яку саме машинку потрібно вибрати для отримання максимального прибутку (мінімальних витрат). На вхідні параметри на програмні застосунки подамо обмеження на джерела електроживлення —бензинові генератори, які надають можливість використовувати швейні машинки на виробництві. При прогнозованій відсутності електроживлення більше 3 год і при електроживленні від різних генераторів проведемо експеримент і результати запишемо в таблицю:

Програмний застосунок	ТоргСофт Виробництво	eqman	За мурашиним алгоритмом
Спосіб електроживлення			
Генератор, 2КВт	0, 0, 0	0, 0, 0	30, 0, 0
Генератор, 10КВт	0, 0, 0	0, 0, 0	30, 36, 50

Загальні висновки

Мурашиний алгоритм виявився ефективним інструментом для вирішення проблеми планування та оптимізації в швейному виробництві. Він дозволяє знайти оптимальний набір рішень, враховуючи обмеження, що існують у виробництві, такі як розмір партії, час виконання, розташування обладнання та інші.

Розроблене рішення на основі мурашиного алгоритму дозволяє знизити час виробництва та витрати на операції швейного процесу. Воно забезпечує оптимальний розподіл ресурсів, таких як робоча сила, обладнання та матеріали, що сприяє підвищенню продуктивності та ефективності виробництва.

Дякую за увагу

02.06.23, 18:09

result_1324902843497498635.html

Fri Jun 02 17:15:22 BST 2023, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 25.0%

Словники перевірки: en_US, pl_PL, ua_UA. Помилки в документах: 9%

ID: 114617			
Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА			
Додано в БД: 2023-06-02			
Автор: Т.В. Пасічник			
Керівники: В.П. Міхалевський			
Консультанти:			
Опоненти:			
Документ		Сумарний збіг по Базі Даних	
Символи	Лексеми	Символи	Лексеми
72497	1131	19885 (27%)	298 (26%)

Джерело платігу

ID	Опис	Наявність платігу в документі	
		Символи	Лексеми
114079	Назва: ЗВІТ з професійної практики Додано в БД: 2023-05-26 Автор: Пасічник Т.В. Керівники: Скрипник Т.К. Консультанти: Опоненти:	18102 (25.0%)	285 (25.0%)



Ім'я користувача:
Кафедра КН

Дата перевірки:
02.06.2023 18:08:10 EEST

Дата звіту:
02.06.2023 18:14:01 EEST

ID перевірки:
1015400050

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: КНс-20-1 Пасічник

Кількість сторінок: 65 Кількість слів: 11195 Кількість символів: 88023 Розмір файлу: 2.85 MB ID файлу: 10150639

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.17%

Схожість

Найбільша схожість: 2.26% з джерелом з Бібліотеки (ID файлу: 1015013145)

7.56% Джерела з Інтернету 397 Сторінка 67

5.32% Джерела з Бібліотеки 114 Сторінка 69

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 5

Підозріле форматування 10 сторінок

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом

Автор: студент групи КНс-20-1 Пасічник Тимофій Віталійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.фіз.-мат.н., доц. Михалевський В.Ц.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Пасічника Т.В., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять поширені конструкції, загальновідомі терміни та скорочення. Серед запозичень є матеріали звіту з професійної практики бакалавра, автором яких є Пасічник Т.В.

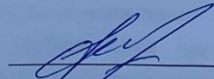
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- За системою Anti-Plagiarism: 25%, з яких 25% є співпадінням із власним звітом з професійної практики, яка забезпечує формування розділів кваліфікаційної роботи бакалавра і занесена в базу матеріалів для перевірки на запозичення. Отже, це запозичення є допустимим і не є плагіатом.

- За системою Unichesk: 9,17%.

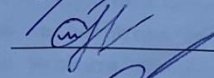
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 25% і 9,17% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



Віталій МИХАЛЕВСЬКИЙ

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента *гр. КНС-20-1 Пасічник Тимофій Віталійович*

за темою Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом

1. Актуальність теми

Актуальним завданням, яке потребує аналізу і досліджується у даній роботі, є отримання рішення для швейного виробництва, які необхідно приймати в умовах обмежень. Процеси отримання рішення завжди були актуальними, а в сьогоденні умовах війни додаються додаткові фактичні та прогнозовані обмеження (пошкодження інфраструктури, відсутність (недостатність) ресурсів, зміна логістичних характеристик тощо) для виробництва. Для ефективного використання автоматизованих систем необхідно передбачити застосування програмного модуля, який би дозволяв формувати рекомендації з отримання рішення при наявності обмежень. Розробка такого способу є актуальною задачею комп'ютерних наук.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є розробка способу отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом та розробка інформаційної системи реалізації вказаного способу. При вирішенні поставленої задачі використано методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Пасічник Тимофій Віталійович проявив себе кваліфікованим фахівцем та дисциплінованим студентом, вчасно виконуючи поставлені етапи дослідження. Як в процесі написання пояснювальної записки, так і при розробці прикладного програмного забезпечення проявив достатні для одержання успішного результату компетентності та результати навчання. Опанував професійні скіли за напрямком «Комп'ютерні науки» та достатньо значний софт скіл. Також серед особистісних якостей студента слід виділити відповідальність, здатність навчатися, нестандартність мислення.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи в достатній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, розроблено програмне забезпечення для перевірки функціональності розробленої системи та проведення експериментів.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі спосіб та його програмна реалізація можуть бути використані працівниками швейних виробництв для прийняття рішень в умовах різного виду обмежень. Ефективність застосування розробленого способу та інформаційної системи за результатами проведених експериментів, наприклад, при обмеженнях у 25% відсотків на основні ресурси, складає 14-17%.

В роботі описано процедуру розгортання та впровадження системи в життя.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи достатній рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту.

Рекомендована оцінка «задовільно».

Керівник _____ к.фіз.-мат.н., доц. Віталій МІХАЛЕВСЬКИЙ



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КНс-20-1 Пасічника Тимофія Віталійовича*

за темою: *Спосіб отримання рішення для швейного виробництва в умовах обмежень за мурашиним алгоритмом*

1. Актуальність обраної теми

В кваліфікаційній роботі бакалавра був розроблений спосіб отримання рішення для швейного виробництва в умовах обмежень. Тема роботи є актуальною на даний час, актуальність обґрунтована дослідженнями умов обмежень, особливостей роботи підприємства та наявних програмно-технічних засобів для супроводу. Актуальність теми роботи швейного виробництва при обмеженнях потрібно розглядати в умовах війни, коли завдається шкода критичній та логістичній інфраструктурі. Тому розробка рішень для оптимізації умов обмежень є надзвичайно важливою для забезпечення продуктивної та оптимізованої роботи швейного виробництва в сучасних умовах.

2. Повнота розкриття мети та завдань роботи

Завдання дослідження повністю розкривають мету роботи. Розроблено новий спосіб отримання рішення для швейного виробництва в умовах обмежень електроживлення, матеріалів, робітників тощо за мурашиним алгоритмом.

Спроектовано функціональну структуру інформаційної системи швейного виробництва в умовах обмежень, розроблено програмну реалізацію мурашиного алгоритму для інформаційної системи швейного виробництва з використанням способу оптимізації процесів швейного виробництва та проведено функціональне і прикладне дослідження розробленої інформаційної системи.

3. Зміст кожного розділу роботи

В першому розділі проаналізовано предметну область та сформульовано постановку задачі на розробку рішення для швейного виробництва в умовах обмежень за допомогою мурашиного алгоритму.

В другому розділі запропоновано і розроблено рішення оптимізації для швейного виробництва в умовах обмежень за допомогою мурашиного алгоритму.

В третьому розділі розглянуто особливості реалізації інформаційної системи та визначено засоби для її розробки. Описано функціональні можливості інформаційної системи та проведено експерименти для перевірки ефективності її роботи.

4. Оцінка розробленої інформаційної системи, її практична цінність

Інформаційна система повністю розроблена відповідно до визначених завдань. Запропонованої інформаційної моделі, схеми способу побудови оптимізації процесів швейного виробництва та функціональної структури на основі мурашиного алгоритму.

Практична цінність розробленої інформаційної системи полягає у можливостях реального її застосування для вирішення завдань. За результатами проведених експериментів оцінено ефективність роботи системи при різних обмеженнях на ресурси. Зокрема, при обмеженнях у 25% відсотків на основні ресурси ефективність роботи системи порівняно з іншими відомими застосунками складає 14-17%.

5. Якість оформлення кваліфікаційної роботи бакалавра

Пояснювальна записка кваліфікаційної роботи бакалавра оформлена відповідно до норм. Мовних, граматичних, синтаксичних помилок не виявлено.

6. Недоліки кваліфікаційної роботи бакалавра

Явних недоліків в роботі не виявлено. Можна було б провести узагальнення роботи системи шляхом розширення діапазону апаратно-технічних засобів, урізноманітнення типів обмежень та удосконалення інтерфейсу користувача.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту.

Рекомендована оцінка «задовільно».

Рецензент

Бегунець І.П.

