

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр
на тему: «Оптимізація процесу реновації кухонного приміщення
житлового помешкання з використанням моделей і методів дослідження
операцій»

Виконав: студент 6 курсу, групи ПДМ–61
спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності/спеціалізації)

Кузнєцов Ілля Іванович

(прізвище та ініціали)

Керівник Жебка В.В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ –2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Магістр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення Негоденко О.В.

“ ___ ” _____ 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

КУЗНЄЦОВУ ІЛІ ІВАНОВИЧУ

(прізвище, ім'я, по батькові)

Тема роботи: Оптимізація процесу реновації кухонного приміщення житлового помешкання з використанням моделей і методів дослідження операцій

Керівник роботи: Жебка Вікторія Вікторівна , д.т.н., доц., завідувач кафедри ТЦР ,

Затверджені наказом вищого навчального закладу від «12» жовтня 2022 року №122.

Строк подання студентом роботи _____

Вхідні дані до роботи

Емпіричні дані та дослідження, які використовуються для розробки математичної моделі.

Оптимізаційні методи, методи дослідження операцій

Науково-технічна література з питань, пов'язаних з розробкою оптимізаційних систем.

Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Дослідження актуальності проблематики дипломної роботи

4.2 Оцінка актуальності існуючих рішень.

4.3 Опис математичних моделей та проектування системи.

4.4 Опис проектування системи.

Перелік демонстраційного матеріалу (назва основних слайдів)

- Актуальність проблеми
- Існуюче програмне забезпечення
- Побудова прогнозу на основі обраних методів
- Аналіз статистичних даних для проведення прогнозу
- Моделювання процесу розміщення зарядних станцій
- Проектування системи
- Аналіз ефективності розробленої системи

Дата видачі завдання _____ 14.10.2022 _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	14.10-25.10	Виконано
2	Вимоги до системи	28.10-05.11	Виконано
3	Оцінка якості тестування до системи	06.11-09.11	Виконано
4	Метод побудови	11.11-20.11	Виконано
5	Концепція та архітектура програмного забезпечення	21.11-30.11	Виконано
6	Вступ, висновки, реферат	30.11-05.11	Виконано
7	Розробка презентації	06.11-11.12	Виконано

Студент _____

Керівник роботи _____

РЕФЕРАТ

Текстова частина магістерської роботи 70 с., 46 рис., 24 джерела.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, АРХІТЕКТУРА, ДИЗАЙН ІНТЕР'ЄРУ,
МЕТОДИ ДОСЛІДЖЕННЯ ОПЕРАЦІЙ, НЕЛІНІЙНЕ ПРОГРАМУВАННЯ,
МЕТОД МНОЖНИКІВ ЛАГРАНЖА, МЕТОД ГРАДІЄНТНОГО СПУСКУ

Мета: оптимізація процесу реновації кухонного приміщення житлового помешкання з використанням моделей і методів дослідження операцій.

Об'єкт: процес реновації кухонного приміщення.

Предмет: методи та засоби дослідження операцій.

Методи дослідження: емпіричні дослідження, методи оптимізації.

Оптимізація планування кухні є важливим аспектом ремонту кухні, оскільки вона може мати значний вплив як на ергономіку, так і на безпечність харчових продуктів.

Для проведення прогнозування завантаження ринку та класифікації даних в магістерській роботі було розроблено математичну модель, представлену у вигляді оптимізаційної задачі та розглянуто можливі методи її розв'язання. На основі використання запропонованих методів було проведено обчислення з урахуванням різних критеріїв. Все це дозволило визначити оптимальне розташування кухонного приладдя у приміщенні. Підводячи підсумок, можна сказати, що оптимізація планування кухні має вирішальне значення для створення безпечної та функціональної кухні, на якій легко та комфортно працювати.

В якості перспективних напрямків подальшого дослідження є альтернативні підходи до оптимізації планування кухні та інтегрування рішення у програми візуалізації готових дизайнів.

Галузь використання – архітектурна галузь.

ЗМІСТ

ВСТУП.....	8
1. РОЗДІЛ 1. ОСОБЛИВОСТІ РЕНОВАЦІЇ ЖИТЛОВОГО ТА КУХОННОГО ПРИМІЩЕННЯ.....	10
1.1. Особливості реконструкції житлових помешкань у пост-воєнний період.....	10
1.2. Пріоритетність реновації кухонного приміщення.....	12
1.3. Особливості реновації житлового приміщення.....	13
1.4. Розташування кухні в квартирі, необхідні комунікації.....	14
1.5. Роль планування розташування кухні у процесі реновації кухонного приміщення.....	15
1.6. Роль розташування кухні у харчовій безпеці.....	16
2. РОЗДІЛ 2. ЕРГОНОМІЧНІ КОНЦЕПЦІЇ У ПРОЦЕСІ РЕНОВАЦІЇ КУХОННОГО ПРИМІЩЕННЯ.....	19
2.1. Роль ергономіки у плануванні розташування кухні.....	19
2.2. Концепція золотого трикутника кухні у процесі планування кухні.....	20
2.3. Порівняльна характеристика існуючих рішень для планування розташування кухонного приміщення у процесі реновації.....	21
2.4. Побудова математичної моделі.....	27
2.5. Обґрунтування методу Лагранжа у розв'язанні створеної оптимізаційної задачі.....	29
2.6. Обґрунтування обраної мови програмування для реалізації математичної моделі.....	35
2.6.1 Загальна характеристика бібліотеки SymPy.....	37
2.6.2 Загальна характеристика бібліотеки NumPy.....	40
2.6.3 Загальна характеристика бібліотеки SciPy.....	43
2.6.4 Загальна характеристика бібліотеки matplotlib.....	47
2.7. Програмна реалізація математичної моделі і застосування методу Лагранжа.....	47

3. РОЗДІЛ 3. РЕЗУЛЬТАТИ РОЗРОБКИ ТА АНАЛІЗ ОТРИМАНОГО РІШЕННЯ.....	51
3.1. Демонстрація роботи з розробленим рішенням.....	51
3.2. Огляд розробленого рішення і обґрунтування його доцільності.....	55
3.3. Можливі майбутні вдосконалення.....	56
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:.....	59
ДОДАТКИ.....	62

ВСТУП

З точки зору ергономіки, добре продумане планування кухні може зменшити фізичне навантаження на користувачів, зробивши роботу на кухні комфортнішою та ефективнішою. Це може включати такі фактори, як висота прилавків, розміщення приладів і легкість доступу до сховища. Враховуючи ергономіку під час ремонту кухні, це може допомогти запобігти травмам і зробити кухню більш доступною для людей з обмеженими можливостями пересування.

З точки зору безпеки харчових продуктів, добре продумане планування кухні може зменшити ризик перехресного зараження між сирими та вареними продуктами, що може призвести до захворювань харчового походження. Правильне розміщення раковини та зони підготовки, а також використання гігієнічних матеріалів, таких як нержавіюча сталь або стільниця з твердою поверхнею, можуть допомогти зменшити ймовірність забруднення. Крім того, використання енергозберігаючих приладів може полегшити прибирання кухні та зменшити ймовірність накопичення бактерій.

Загалом, оптимізація планування кухні з урахуванням ергономіки та безпеки харчових продуктів може значно покращити функціональність, безпеку та комфорт кухонного простору. Важливо проконсультуватися з професійним дизайнером або дизайнером кухні, який знайомий з місцевими будівельними нормами та правилами, щоб переконатися, що ремонт вашої кухні відповідає всім необхідним стандартам безпеки та вимогам користувачів.

Мета: оптимізація процесу зарядки електромобілів за рахунок прогнозування навантаження та моделювання розташування зарядних станцій з використанням методів дослідження операцій.

Об'єкт: процес зарядки електромобілів.

Предмет: методи та засоби оптимізації та моделювання.

Завдання дослідження:

- Аналіз ергономічності існуючих концепцій;
- Перевірка доцільності рішень на основі емпіричних досліджень;
- Вибір концепції для розробки математичної моделі;
- Розробка математичної моделі;
- Оптимізація математичної моделі з допомогою існуючих методів;
- Аналіз ефективності запропонованих методів.

Наукова новизна магістерської роботи: побудова інформаційної технології визначення оптимального розподілу зарядних станцій на основі прогнозованих даних завантаження мережі електромобілями та з використанням методів дослідження операцій.

Методи дослідження: оптимізаційні методи, методи нелінійного програмування, методи математичного аналізу.

Практичне значення: побудовану інформаційну технологію можна використовувати для оптимального розміщення ключових точок кухонного приміщення.

Запропоновані результати дослідження були апробовані на XV Науково-технічній конференції «Сучасні інфокомунікаційні технології» в Державному університеті телекомунікацій.

За результатами отриманих досліджень були опубліковані наступні матеріали:

1. Кузнецов І.І. КОНЦЕПЦІЯ ЗОЛОТОГО ТРИКУТНИКА ДЛЯ ЗАБЕЗПЕЧЕННЯ ОПТИМАЛЬНОСТІ ПЛАНУ ДИЗАЙНУ КУХНІ З ТОЧКИ ЗОРУ ФУНКЦІОНАЛЬНОСТІ // XV Науково-технічна конференція «Сучасні інфокомунікаційні технології» . – Київ: ДУТ, 2022. – с. 91-92
2. Кузнецов І.І. Оптимізація процесу реновації кухонного приміщення житлового помешкання з використанням моделей і методів дослідження операцій // ТІТ. №3, 2022, с. 31-35

1. РОЗДІЛ 1. ОСОБЛИВОСТІ РЕНОВАЦІЇ ЖИТЛОВОГО ТА КУХОННОГО ПРИМІЩЕННЯ

1.1. Особливості реконструкції житлових помешкань у пост-воєнний період

Реконструкція відноситься до процесу відновлення або ремонту будівель та інфраструктури, які були пошкоджені або зруйновані, часто після війни або стихійного лиха. У післявоєнний період зусилля з реконструкції часто зосереджувались на відбудові будинків, шкіл, лікарень та інших важливих споруд з метою відновлення громад і повернення життя до нормального, наскільки це можливо. У повоєнний період було кілька основних особливостей відбудовних робіт:

- Пріоритет надавався відбудові основної інфраструктури, такої як дороги, мости та комунікації, щоб сприяти відбудові будинків та інших споруд і відновити основні послуги.
- Тимчасове житло, таке як намети або збірні конструкції, часто надавалося людям, які втратили свої домівки, щоб забезпечити притулок на час будівництва більш постійного житла.
- Реконструкція часто передбачала використання нових будівельних технологій і матеріалів, таких як збірні панелі та модульні конструкції, щоб прискорити процес реконструкції.
- У деяких випадках зусилля з реконструкції супроводжувалися ініціативами земельної реформи, яка мала на меті перерозподіл землі між тими, хто втратив її під час війни, або тими, хто раніше був безземельним.
- Реконструкція часто фінансувалася міжнародними організаціями з надання допомоги, урядами та приватними донорами, і часто

виконувалася спільно місцевими працівниками та міжнародними працівниками.

Проте слід зазначити, що відбудова не завжди передбачає відновлення повністю зруйнованого житла. Частіше зустрічатимуться випадки, коли житлове приміщення було пошкоджене лише частково і незначуще, тому відповідальність за фінансування реконструкції буде лежати саме на власнику квартири чи будинка. Особливо це стосується процесу реновації житлового помешкання, коли план, фінансовий бюджет та ресурси не покриваються державними ініціативами.



Рисунок 1.1 – Інфографіка руйнувань приватних будинків станом на 05.07.2022

1.2. Особливості реновації житлового помешкання

Ремонт будинку – це процес покращення конструкції, як правило, будинку, шляхом ремонту, реконструкції або оновлення різних аспектів власності. Це може включати в себе встановлення нової підлоги, фарбування стін, оновлення сантехніки та електричних систем, а також ремонт кухні чи ванної кімнати.

Існує багато різних типів проектів ремонту будинку, які можуть бути здійснені в залежності від потреб власника будинку. Деякі поширені види ремонту включають:

- Ремонт кухні: це може включати оновлення шаф, стільниць і приладів, а також переробку підлоги та освітлення.
- Ремонт ванної кімнати: це може включати встановлення нових приладів, таких як новий туалет або душ, а також оновлення плитки, підлоги та освітлення.
- Оздоблення підвалу: це може включати додавання житлової площі дому шляхом обробки підвалу, який можна використовувати як сімейну кімнату, домашній офіс або додаткову спальню.
- Доповнення: цей тип ремонту передбачає додавання додаткового простору до будинку, наприклад будівництво другого поверху або солярію.

1.3. Пріоритетність реновації кухонного приміщення

У контексті зусиль з відновлення після війни чи стихійного лиха пріоритетність ремонту кухні може відрізнитися залежно від конкретних обставин і потреб постраждалої громади. Загалом, пріоритетність ремонту кухні, швидше за все, буде нижчою, ніж відновлення основної інфраструктури, як-от доріг, мостів та комунікацій, а також будівництво тимчасового чи постійного житла для людей, які втратили дім.

Однак ремонт кухні може бути важливою частиною процесу реконструкції, особливо якщо він потрібен для відновлення основних послуг і функціональності будинків і громад. Наприклад, якщо водопостачання громади було пошкоджено або забруднено, може знадобитися ремонт кухні, щоб встановити нову сантехніку та відновити доступ до чистої води. Подібним чином, якщо кухня була значно пошкоджена внаслідок стихійного лиха, може знадобитися ремонт, щоб зробити простір безпечним і функціональним для приготування їжі та приготування їжі.

Pu

Зрештою, пріоритетність ремонту кухні в контексті реконструкції залежатиме від конкретних потреб і ресурсів постраждалої громади та може мати різний пріоритет залежно від ситуації.

У контексті ремонту будинку ремонт кухні – це процес оновлення або покращення функціональності, зовнішнього вигляду та/або енергоефективності кухонного простору. Це може включати низку заходів, таких як заміна старих приладів на нові, встановлення нових стільниць або шаф, оновлення планування кухні або додавання нових елементів, таких як острів чи комора.

Ремонт кухні може бути серйозним завданням і може вимагати досвіду підрядників, електриків, сантехніків та інших професіоналів. Вартість і масштаби ремонту кухні залежатимуть від низки факторів, включаючи розмір кухні, обсяг ремонту, що проводиться, матеріали та обладнання, що використовується, а також витрати на оплату праці.

Ремонт кухні може підвищити цінність дому та покращити якість життя власників житла, оскільки функціональна та добре спроектована кухня може зробити приготування та приготування їжі ефективнішим та приємнішим. Однак ремонт кухні також може бути дорогим і трудомістким, тому власникам будинків важливо ретельно зважити свій бюджет і потреби, перш ніж приступати до ремонту.

1.4. Розташування кухні в квартирі, необхідні комунікації

Ремонт кухні є популярним проектом покращення будинку, оскільки кухню часто вважають серцем дому. Існує багато різних аспектів ремонту кухні, які можуть бути розглянуті залежно від конкретних потреб і цілей власника будинку. Деякі загальні елементи ремонту кухні включають:

- **Шафи:** заміна або переробка шаф може надати кухні нового вигляду та забезпечити додатковий простір для зберігання.
- **Стільниці:** Оновлення стільниць також може суттєво змінити зовнішній вигляд кухні, а також підвищити довговічність і функціональність. Доступні різні варіанти, такі як граніт, мармур, бетон, кварц і ламінат тощо.
- **Підлога:** заміна підлоги також може значно змінити зовнішній вигляд кухні. Плитка, листяна деревина та вінілова підлога є популярними варіантами.
- **Побутова техніка:** заміна на нову, енергоефективну техніку може покращити функціональність кухні та знизити витрати на комунальні послуги.
- **Освітлення:** нові освітлювальні прилади та покращене природне освітлення також можуть значно покращити атмосферу кухні.

Коли справа доходить до ремонту кухні, важливо враховувати як естетику та функціональність приміщення, так і те, як воно буде використовуватися. Багато домовласників вирішують співпрацювати з дизайнером або майстром з ремонту кухні, щоб допомогти спланувати ремонт і переконатися, що він відповідає їхнім потребам і вподобанням. Також важливо врахувати вартість і бюджет на ремонт, а також визначити пріоритетність найважливіших для вас елементів.

Завжди доцільно отримати кілька пропозицій від різних підрядників і перевірити їхні рекомендації, перш ніж прийняти остаточне рішення, а також проконсультуватися з місцевими будівельними нормами для конкретної області.

1.5. Роль планування розташування кухні у процесі реновації кухонного приміщення

Планування планування кухні є важливим етапом у процесі ремонту кухні, оскільки воно визначає хід роботи та організацію інструментів та обладнання на кухні. Добре продумане планування кухні може допомогти зробити кухню більш функціональною, ефективною та приємною для роботи, тоді як погано сплановане планування може призвести до таких проблем, як затори, недостатнє зберігання та труднощі з підтриманням чистоти.

При плануванні планування кухні необхідно враховувати кілька факторів, зокрема:

- Розмір і форма кухні: планування кухні має бути адаптоване до розміру та форми кухонного простору з урахуванням розташування вікон, дверей та інших фіксованих елементів.
- Робочий процес на кухні: планування кухні має бути розроблено таким чином, щоб оптимізувати робочий процес користувача, щоб предмети та прилади, якими найчастіше користуються, були легко доступні.
- Зберігання та організація: План кухні повинен передбачати адекватне зберігання та організацію для інструментів та обладнання, а також для їжі та інших запасів.
- Комфорт і ергономіка: планування кухні має враховувати комфорт і ергономіку користувача, з оптимізованими зонами досяжності та висотою стільниці для зручності використання.

Враховуючи ці та інші фактори під час планування кухні, можна створити кухонний простір, який буде функціональним, ефективним і приємним для роботи.

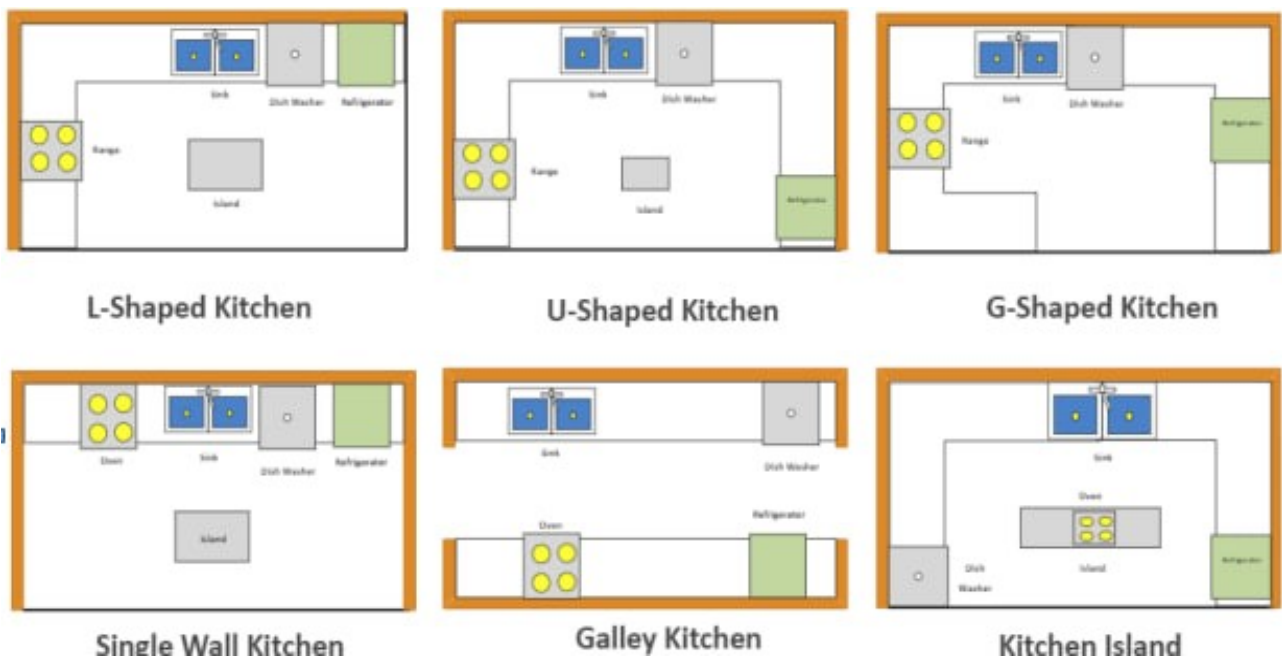


Рис. 1.3 – можливі варіанти планування розташування кухні

1.6. Роль розташування кухні у харчовій безпеці

Планування кухні може відігравати значну роль у безпечності харчових продуктів, оскільки воно визначає хід роботи та організацію інструментів та обладнання. Погано спроектоване планування кухні може призвести до таких проблем, як перехресне забруднення між сирими та вареними продуктами, неправильне зберігання їжі та труднощі з підтриманням чистоти.

Є кілька факторів, які слід враховувати при проектуванні планування кухні з урахуванням безпеки харчових продуктів:

- Розподіл завдань: планування кухні має розділяти такі завдання, як приготування їжі, приготування їжі та прибирання, щоб мінімізувати ризик перехресного зараження. Це може передбачати проектування

окремих зон для цих завдань або використання інструментів і обладнання, призначених для конкретних завдань.

- Доступ до засобів для миття рук: Миття рук є важливою частиною безпеки харчових продуктів, і планування кухні має забезпечувати легкий доступ до засобів для миття рук з усіх куточків кухні.
- Відповідний простір для зберігання продуктів: План кухні повинен передбачати достатній простір для зберігання продуктів, наприклад, холодильників, комор і сухих складських приміщень, щоб гарантувати, що продукти зберігаються при належній температурі та вологості.
- Легкий доступ до інструментів і обладнання: планування кухні має забезпечувати легкий доступ до інструментів і обладнання, оскільки це може допомогти зменшити ризик нещасних випадків і зробити приготування їжі більш ефективним.

Враховуючи ці та інші фактори безпеки харчових продуктів під час проектування планування кухні, можна створити простір, який буде одночасно функціональним і безпечним для приготування та зберігання їжі.

2.1. Роль ергономіки у плануванні розташування кухні

Ергономіка — це наука про те, як люди взаємодіють із навколишнім середовищем, і займається проектуванням і організацією об'єктів і просторів у спосіб, який є ефективним і зручним для користувача. У контексті дизайну кухні ергономіка відіграє важливу роль у плануванні планування кухні, щоб зробити її максимально функціональною та ефективною для користувача.

Є кілька ключових факторів, які слід враховувати при впровадженні ергономіки в дизайн планування кухні:

- Робочий процес: планування кухні має бути розроблено таким чином, щоб оптимізувати робочий процес користувача, щоб предмети та прилади, якими найчастіше користуються, були легко доступні.
- Зони досяжності: Предмети та прилади слід розташовувати на кухні таким чином, щоб користувач міг легко дістатися до них, не розтягуючись або надмірно нахиляючись.
- Висота стільниці: стільниці мають бути на зручній висоті для користувача, зазвичай близько 36 дюймів для дорослих користувачів.
- Освітлення: відповідне освітлення є важливим для безпеки та ефективності на кухні, і його слід розміщувати таким чином, щоб мінімізувати тіні та відблиски.

Враховуючи ці та інші ергономічні фактори під час планування кухні, можна створити комфортний, ефективний простір, у якому приємно працювати.

2.2. Концепція золотого трикутника кухні у процесі планування кухні

Правило золотого трикутника - це принцип дизайну кухні, який передбачає, що предмети, які найчастіше використовуються на кухні, повинні бути розташовані у формі трикутника, де три точки трикутника - мийка, холодильник і плита. Це засновано на ідеї, що ці три предмети найчастіше використовуються на кухні, і що їх розміщення у формі трикутника забезпечує найбільш ефективний робочий процес.

Щоб оптимізувати планування кухні за правилом золотого трикутника, можна виконати наступні дії:

1. Виміряйте відстань між раковиною, холодильником і плитою: це допоможе визначити оптимальний розмір трикутника та відстань між трьома точками.
2. Визначте основного користувача: Розмір і форма трикутника можуть змінюватися залежно від основного користувача кухні, а також від його висоти та довжини. Враховуйте планування кімнати: форму та розмір кухні слід брати до уваги при визначенні розміщення трикутника, а також будь-яких інших фіксованих елементів, таких як вікна чи двері.
3. Розташуйте раковину, холодильник і плиту в формі трикутника: точки трикутника повинні бути на однаковій відстані одна від одної, причому раковина зазвичай розташована у верхній частині трикутника.
4. Розташуйте інші часто використовувані предмети в трикутнику: коли трикутник буде встановлено, інші часто використовувані предмети, такі як мікрохвильова піч, тостер і посудомийна машина, слід розмістити поблизу трикутника, щоб оптимізувати робочий процес на кухні.

Дотримуючись цих кроків, можна створити функціональне та ефективне планування кухні, використовуючи принципи правила золотого трикутника.

TRIANGLES IN KITCHEN

ERGONOMY TO FOOD SAFETY

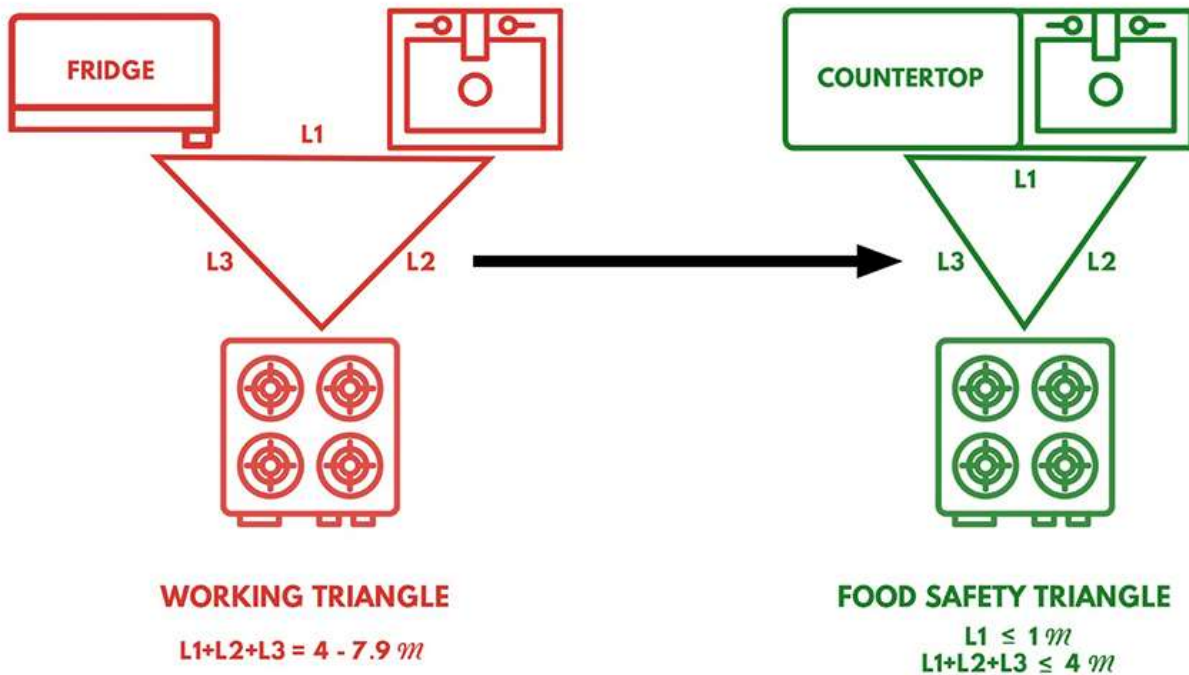


Рис. 2.1 – Концепція та правила золотого трикутника з точки зору ергономіки та харчової безпеки

2.3. Порівняльна характеристика існуючих рішень для планування розташування кухонного приміщення у процесі реновації

Існує багато різних програм автоматизованого проектування (CAD), які можна використовувати для ремонту кухні, кожна з яких має свої унікальні функції та можливості. Ось декілька програм САПР, які зазвичай використовуються в контексті ремонту кухні, і порівняння їхніх основних функцій:

- AutoCAD: AutoCAD — це програма САПР професійного рівня, яка широко використовується в різних галузях, зокрема в архітектурі та

дизайні інтер'єрів. Він має широкий набір інструментів і функцій для створення точних 2D і 3D дизайнів і підходить для складних ремонтів і планування кухонь на замовлення.

- SketchUp: SketchUp — це зручна програма САПР, яка користується популярністю серед дизайнерів і ентузіастів DIY. Має простий інтерфейс і велику бібліотеку готових об'єктів, що дозволяє легко створювати 3D-моделі ремонту кухні.
- Sweet Home 3D: Sweet Home 3D — це безкоштовна програма САПР із відкритим вихідним кодом, яка спеціально розроблена для проектування та ремонту будинків. Він має ряд інструментів для створення 2D і 3D планів поверхів і дозволяє користувачам візуалізувати свої проекти в 3D з реалістичним освітленням і матеріалами.
- Homestyler: Homestyler — це безкоштовна веб-програма САПР, яка має низку функцій для створення 2D та 3D дизайнів. Він має велику бібліотеку попередньо розроблених об'єктів і дозволяє користувачам візуалізувати свої проекти в 3D з реалістичними матеріалами та освітленням.

Ось порівняння основних функцій і переваг програм САПР, згаданих раніше:

1. AutoCAD:

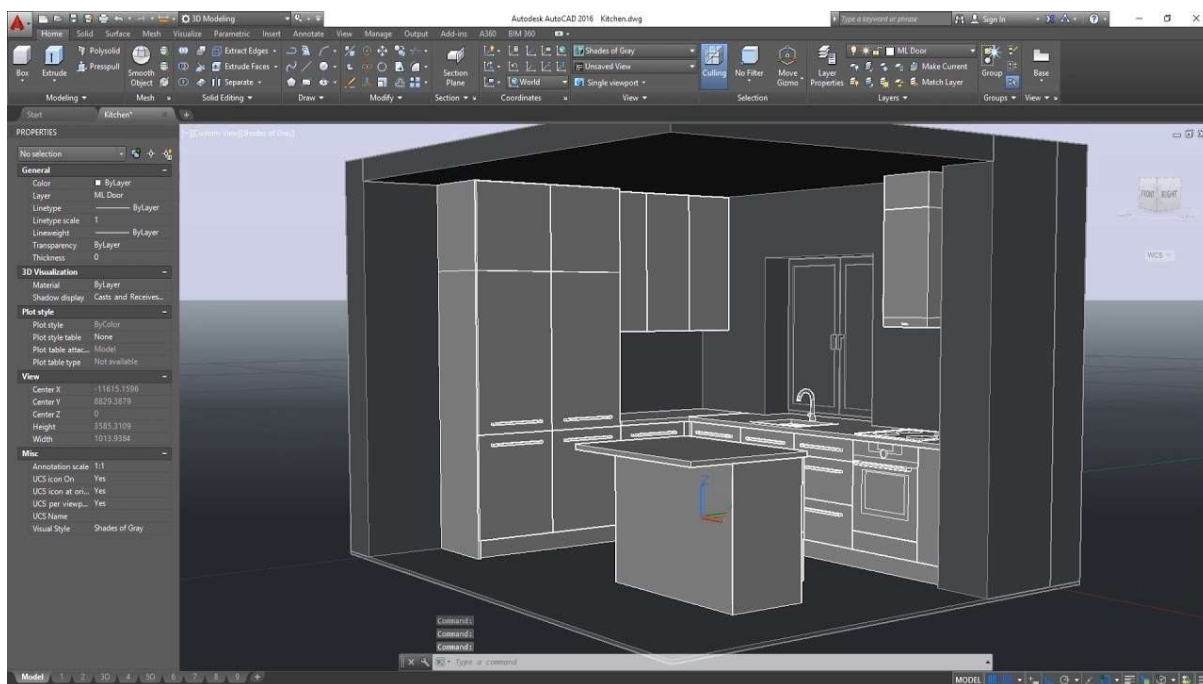


Рис. 2.2 – Приклад використання програми AutoCAD

- Програма CAD професійного рівня з широким набором інструментів і функцій для створення точних 2D і 3D проєктів;
- Підходить для комплексного ремонту та нестандартного планування кухні;
- Широко використовується в різних галузях, включаючи архітектуру та дизайн інтер'єру.

2. SketchUp:



Рис. 2.3 – Приклад роботи у програмі SketchUp

- Зручна програма САПР із простим інтерфейсом і великою бібліотекою попередньо розроблених об'єктів;
- Легко створювати 3D моделі ремонту кухні;
- Популярний серед дизайнерів і любителів саморобки.

3. Sweet Home 3D:



Рис. 2.4 – Приклад роботи у Sweet Home 3D

- Безкоштовна програма CAD з відкритим вихідним кодом, спеціально розроблена для проектування та ремонту будинків;
- Набір інструментів для створення 2D і 3D планів поверхів;
- Дозволяє користувачам візуалізувати свої проекти в 3D з реалістичним освітленням і матеріалами.

4. Homestyler:



Рис. 2.5 – Приклад роботи у Homestyler

- Безкоштовна веб-програма САПР;
- Набір функцій для створення 2D і 3D дизайнів;
- Велика бібліотека попередньо спроектованих об'єктів і дозволяє користувачам візуалізувати свої проекти в 3D з реалістичними матеріалами та освітленням.

Зрештою, найкраща програма САД для ремонту кухні залежатиме від конкретних потреб і вподобань користувача, а також від складності проекту ремонту. Деякі користувачі можуть віддати перевагу програмі більш професійного рівня з широким набором інструментів і функцій, тоді як інші можуть віддати перевагу більш зручній програмі з простішим інтерфейсом.

Однак, слід зазначити, що процес планування в даних програмах не є автоматизованим, оскільки основний пріоритет в них полягає у візуалізації вже готового плану чи шаблонів. Користувач вручну підбирає, що з його естетичної точки зору йому підходить, що не завжди відповідає ергономічним стандартам у

його індивідуальній ситуації. Тому при плануванні кухні відповідальний дизайнер звіряється з належними метриками вручну, часто вдаючись до вимірювань відстаней і складних математичних обчислень.

2.4. Побудова математичної моделі

На основі концепції правила золотого трикутника можливо побудувати математичну модель у вигляді оптимізаційної задачі.

Нехай відповідні вершини фігури представлені у вигляді набору координат x та y . Враховуючи це, площею утвореної з цих вершин фігури ми отримуємо з формули:

$$S = \left| \frac{x_1y_2 - x_2y_1 + x_2y_3 - x_3y_2 + x_3y_4 - x_4y_3 + x_4y_1 - x_1y_4}{2} \right|$$

Рис. 2.6 – Формула фігури, утвореної з чотирьох точок

Далі, описуємо математично довжини сторін цієї фігури:

$$a_{\text{рак, пл}} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$b_{\text{рак, хол}} = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}$$

$$c_{\text{рак, ст}} = \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2}$$

$$d_{\text{хол, пл}} = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$$

Рис. 2.7 – Формули довжини відстані між двома точками

Маючи необхідні змінні, записуємо їх в оптимізаційну задачу, враховавши правило золотого трикутника та трикутника харчової безпеки:

$$f = S \rightarrow \max$$

$$\left\{ \begin{array}{l} 4 \leq a + b + d \leq 8 \\ a + b + c \leq 4 \\ 1.2 \leq a, b, d \leq 2.7 \\ c \leq 1 \end{array} \right.$$

Рис. 2.8 – Цільова функція та система обмежень

Задача нелінійного програмування (NLP) – це задача знаходження мінімуму або максимуму функції, яка задана нелінійним виразом. Нелінійність означає, що функція містить нелінійні похідні чи певні нелінійні обмеження.

Тепер опишемо усі мовні позначення і що вони означають.

S – площа деякої фігури, утвореної з точок, представлених у вигляді набору координат x та y . У даному випадку S виступає у ролі цільової функції. Доцільно вважати, що ми хочемо максимізувати площу фігури, оскільки так в нас буде більше вільного простору.

a, b, c, d – сторони вищезазначеної фігури. Позначаються як відстані між ключовими точками золотого трикутника, тобто холодильником, раковиною і духовою шафою, плюс додаткова точка, яка відповідає за критерій харчової безпеки – робочий стіл. Формули площі та довжин маємо з лінійної алгебри.

Систему обмежень ми отримали з основних правил золотого трикутника, а саме обмеження на периметр трикутника і на довжини сторін. Додаткові обмеження отримані з емпіричних досліджень, направлених на поліпшення

харчової безпеки у кухонному середовищі. Безпосередньо це стосується відстані від раковини до робочого столу, задля уникнення крос-контамінаційних подій, а також отриманого тим же емпіричним досвідом трикутника, якого назвали трикутником харчової безпеки.

Внаслідок ми отримали задачу нелінійного програмування, оскільки ми маємо нелінійну цільову функцію.

2.5. Обґрунтування методу Лагранжа у розв'язанні створеної оптимізаційної задачі

Метод Лагранжа — це техніка математичної оптимізації, яка використовується для знаходження мінімального або максимального значення функції, що підпадає під набір обмежень. Це особливо корисно для оптимізації нелінійних функцій, які не мають лінійного зв'язку між входами та виходами.

Метод Лагранжа працює, виражаючи функцію, яку потрібно оптимізувати (звану цільовою функцією), як лінійну комбінацію набору змінних (званих множниками Лагранжа) і обмежень проблеми. Далі цей метод передбачає розв'язування набору одночасних рівнянь для знаходження значень множників Лагранжа, які призводять до оптимального значення цільової функції.

Є кілька причин, чому метод Лагранжа особливо корисний для оптимізації нелінійних функцій:

- Він дозволяє включати обмеження: метод Лагранжа може обробляти обмеження на змінні проблеми, що неможливо з деякими іншими методами оптимізації. Це робить його особливо корисним для оптимізації нелінійних функцій, які мають складні обмеження.
- Він широко застосовний: метод Лагранжа можна використовувати для оптимізації широкого діапазону нелінійних функцій, включаючи функції з декількома змінними та функції, які не є безперервно диференційованими.

- Його відносно просто реалізувати: метод Лагранжа відносно простий у реалізації та не потребує використання складних математичних методів, що робить його доступним методом оптимізації для багатьох користувачів.

Загалом, метод Лагранжа є потужним і широко застосовним методом оптимізації, який особливо добре підходить для оптимізації нелінійних функцій.

Метод Лагранжа працює шляхом побудови полінома, який проходить через набір точок, відомих як точки інтерполяції, які вибираються як репрезентативні для поведінки функції. Ось кроки для використання методу Лагранжа для пошуку наближеного розв'язку нелінійної функції:

- 1 Виберіть набір із $n+1$ точок інтерполяції, де n — ступінь полінома, який ви бажаєте побудувати. Точки інтерполяції вибрано так, щоб відображати поведінку функції.
- 2 Для кожної точки інтерполяції обчисліть значення функції в цій точці.
- 3 Для кожної точки i обчисліть поліном Лагранжа $L(x) = \sum L_i(x)f(x_i)$, де $L_i(x) = \prod (x-x_j)/(x_i-x_j)$ для $j \neq i$. Розв'язком функції є значення $L(x)$ у точці x .
- 4 Повторіть кроки 3 і 4 для кожного значення x в області визначення функції.

Загальний вигляд задачі, яка вирішується методом Лагранжа, представлений так:

$$Z = f(x_1, x_2, \dots, x_n) \rightarrow \max(\min)$$

Рис. 2.9 – Загальна формула цільової функції задачі нелінійного програмування

з такими умовами:

$$q_i(x_1, x_2, \dots, x_n) = b_i$$

$$(i = \overline{1, m})$$

Рис. 2.10 – Загальний вигляд критерію у системі обмежень

Ідея методу множників Лагранжа полягає в тому, щоб замінити цю задачу однією: знайти екстремум більш складної функції, але без обмежень. Ця функція називається функцією Лагранжа і має вигляд:

$$L(x_1, x_2, \dots, x_n; \lambda_1, \lambda_2, \dots, \lambda_m) =$$

$$= f(x_1, x_2, \dots, x_n) + \sum_{i=1}^m \lambda_i (b_i - q_i(x_1, x_2, \dots, x_n)),$$

Рис. 2.11 – Загальна формула функції Лагранжа

Де лямбда величини зветься множниками Лагранжа.

Після цього ми знаходимо частинні похідні за усіма змінними. В результаті ми отримуємо рівняння, які ми прирівнюємо до нуля. Так ми отримуємо систему:

$$\frac{\partial L}{\partial x_j} = 0 \quad (j = \overline{1, n}),$$

$$\frac{\partial L}{\partial \lambda_i} = 0 \quad (i = \overline{1, m}),$$

Рис. 2.12 – Частинні похідні функції лагранжа

$$\begin{cases} \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial q_i(x_1, x_2, \dots, x_n)}{\partial x_j} = 0 & (j = \overline{1, n}), \\ b_i - q_i(x_1, x_2, \dots, x_n) = 0 & (i = \overline{1, m}), \end{cases}$$

Рис. 2.13 – Розв'язання оптимізаційної задачі нелінійного програмування

Розв'язавши її ми отримуємо множину розв'язків.

$$X^* = (x_1, x_2, \dots, x_n)$$

$$\lambda^* = (\lambda_1, \lambda_2, \dots, \lambda_m)$$

Рис. 2.14 – Загальний вигляд розв'язку

Вони можливо відповідають умовам екстремуму.

Розглянемо приклад розв'язання задачі нелінійного програмування. Товариство з обмеженою відповідальністю відвело 1200 га сільськогосподарських угідь під основні культури – озиму пшеницю та цукровий буряк. Техніко-економічні показники вирощування цих культур представлені в таблиці:

Показник	Площа, га, відведена	
	під озиму пшеницю, x_1	під цукрової буряк, x_2
Урожайність, т/га	4	35
Ціна, грн./т	800	300
Собівартість, грн./т	$y_1 = 12,5x_1^2 - 200x_1 + 1200$	$y_2 = 12,5x_2^2 - 150x_2 + 650$

Нехай x_1 — площа ріллі, відведена під сотні гектарів озимої пшениці; x_2 – площа ріллі, відведена під цукрові буряки, сотень га. Зазначимо, що вартість тонни пшениці та цукрових буряків залежить від відповідної посівної площі. Запишемо економіко-математичну модель. За критерій оптимальності беремо максимізацію валового прибутку:

$$f = 4(800 - 12,5x_1^2 + 200x_1 - 1200)x_1 + 35(300 - 12,5x_2^2 + 150x_2 - 650)x_2 =$$

$$= 4(-12,5x_1^3 + 200x_1^2 - 400x_1) + 35(-12,5x_2^3 + 150x_2^2 - 350x_2)$$

з критерієм:

$$x_1 + x_2 = 12$$

Функція Лагранжа виглядатиме так:

$$L(x_1, x_2, \lambda_1) = 4(-12,5x_1^3 + 200x_1^2 - 400x_1) +$$

$$+ 35(-12,5x_2^3 + 150x_2^2 - 350x_2) + \lambda_1(12 - x_1 - x_2) = 0.$$

Частинні похідні виглядатимуть так:

$$\begin{aligned}\frac{\partial L}{\partial x_1} &= 4(-37,5x_1^2 + 400x_1 - 400) - \lambda_1 = 0 \\ \frac{\partial L}{\partial x_2} &= 35(-37,5x_2 + 300x_2 - 350) - \lambda_1 = 0 \\ \frac{\partial L}{\partial \lambda} &= 12 - x_1 - x_2 = 0.\end{aligned}$$

Далі визначимо сідлову точку. Прирівнюємо перший і другий вирази,

$$-150x_1^2 + 1600x_1 - 1600 = -1312,5x_2^2 + 10500x_2 - 12250.$$

Використавши метод підстановки, отримуємо:

$$\begin{aligned}-150(12 - x_2)^2 + 1600(12 - x_2)^2 - 1600 &= -1312,5x_2^2 + 10500x_2 - 12250, \\ \text{або } 1162x_2^2 - 8500x_2 + 11450 &= 0.\end{aligned}$$

Що є квадратним рівнянням. Яке ми можемо розв'язати і отримати:

$$x_2^{(1)} \approx 1,78 \text{ (178 га); } x_2 \approx 5,53 \text{ (553 га)}.$$

$$x_1^{(1)} \approx 10,22 \text{ (1022 га); } x_1^2 \approx 6,47 \text{ (647 га)}.$$

Що є сідловими точками. Підставивши їх у цільову функцію, знаходимо, що друга пара точок є максимумом.

$$f(x_1 = 6,47; x_2 = 5,53) = 4(800 - 523,26 + 1294 - 1200)647 + 35(300 - 382,26 + 829,5 - 650)553 = 4\ 625\ 863.$$

Відповідно функція набуває максимального значення за умови, що озима пшениця вирощується на площі 647 га, а цукровий буряк — на площі 553 га.

2.6. Обґрунтування обраної мови програмування для реалізації математичної моделі

Python — потужна та універсальна мова програмування, яка широко використовується для різноманітних завдань. Однією з її головних переваг є легкий для читання та запису синтаксис, що робить її чудовою мовою як для початківців, так і для досвідчених програмістів. Він має велику та активну спільноту, яка надає велику кількість ресурсів, документації та бібліотек майже для будь-яких завдань.

Інтерпретований характер Python дозволяє швидко створювати прототипи та розробляти. Він підтримує кілька парадигм програмування, включаючи об'єктно-орієнтоване, функціональне та процедурне, що робить його придатним для широкого кола завдань. Python також підтримує кілька операційних систем і платформ, що робить його кросплатформною мовою.

Стандартна бібліотека Python є великою та всебічною, вона має вбудовану підтримку для широкого кола завдань, таких як файловий ввід/вивід, мережевий зв'язок, стиснення даних тощо. Крім того, Python має дуже велику екосистему сторонніх бібліотек, таких як NumPy, SciPy, SymPy, pandas, matplotlib, TensorFlow та багато інших, які можна використовувати для розширеного аналізу даних, машинного навчання, штучного інтелекту та наукових обчислень. Ці бібліотеки полегшують виконання складних завдань і дозволяють

розробникам зосередитися на проблемі, що розглядається, а не на основних деталях реалізації.

Python використовується в багатьох різних галузях, таких як веб-розробка, наукові обчислення, аналіз даних, штучний інтелект, освіта тощо. Він широко використовується в наукових та інженерних проектах, аналізі та візуалізації даних, веб-розробці та для створення сценаріїв у різних сферах. Python також використовується в індустрії програмного забезпечення для розробки настільних і веб-додатків, аналізу даних і штучного інтелекту.

Підсумовуючи, Python — це потужна та універсальна мова програмування, яку легко вивчати та використовувати, вона має велику та активну спільноту, а також має широкий спектр бібліотек і фреймворків для підтримки різних сфер, це чудовий інструмент для початківців і досвідчених. Як для програмістів, Python є чудовим вибором для широкого кола завдань, від простого створення сценаріїв до розробки складного програмного забезпечення.

Є кілька причин, чому Python може бути хорошим вибором для реалізації математичних функцій і методів оптимізації:

- Python — це мова програмування загального призначення, яка широко використовується в наукових обчисленнях і аналізі даних, завдяки чому вона добре підходить для реалізації математичних функцій і алгоритмів.
- Python має велику стандартну бібліотеку та велику кількість бібліотек сторонніх розробників, які забезпечують підтримку математичних і наукових обчислень, включаючи бібліотеки для лінійної алгебри, оптимізації та чисельних обчислень.
- Python має простий, зрозумілий синтаксис і велику спільноту користувачів, що полегшує вивчення та використання, а також надає велику кількість онлайн-ресурсів і підтримки для тих, хто новачок у програмуванні.
- Python має хорошу продуктивність і здатний виконувати завдання, що містять інтенсивні обчислення, що робить його придатним для реалізації

методів оптимізації та інших математичних алгоритмів, які можуть потребувати значних обчислювальних ресурсів.

Загалом Python — це універсальна та потужна мова програмування, яка добре підходить для реалізації математичних функцій і методів оптимізації.

2.6.1 Загальна характеристика бібліотеки SymPy

SymPy — це бібліотека Python для символьної математики, яка забезпечує підтримку широкого спектру математичних операцій і функцій. Він призначений для використання в наукових і інженерних програмах і широко використовується в наукових колах і промисловості для таких завдань, як алгебраїчні маніпуляції, обчислення та розв'язування систем рівнянь.

Деякі з основних функцій SymPy включають:

- Символьна математика: SymPy дозволяє користувачам виконувати символьну математику, яка передбачає маніпулювання математичними виразами символічним, а не числовим способом. Це дає змогу точно представити математичні концепції, а не покладатися на наближення.
- Алгебраїчні маніпуляції: SymPy має набір інструментів для виконання алгебраїчних маніпуляцій, таких як розширення та спрощення виразів, розкладання поліномів на множники та розв'язування рівнянь.
- Обчислення: SymPy має ряд функцій для виконання операцій обчислення, таких як диференціювання та інтегрування функцій, знаходження меж і розв'язування диференціальних рівнянь. Розв'язування систем рівнянь: SymPy можна використовувати для розв'язування систем рівнянь як символічно, так і чисельно, використовуючи такі методи, як видалення, підстановка та матричні операції.

- Символьне малювання: SymPy можна використовувати для створення графіків символічних виразів за допомогою різноманітних бібліотек графіків, таких як matplotlib і plotly.

Ось кілька прикладів того, що можна робити за допомогою SymPy:

1. Маніпулювання символами: за допомогою SymPy ви можете створювати символічні змінні та функції та керувати ними в математичних виразах. Наприклад, ви можете створити символ x і використовувати його у виразі, наприклад:

```
from sympy import Symbol
x = Symbol('x')
y = x**2 + 1
```

Рис. 2.15 – Приклад маніпулювання символами

2. Спрощення математичних виразів: SymPy може спрощувати математичні вирази, наприклад:

```
from sympy import simplify
expr = (x**3 + x**2 - x - 1)/(x**2 + 2*x + 1)
simplified_expr = simplify(expr)
```

Рис. 2.16 – Приклад спрощення виразів

3. Обчислення: SymPy може виконувати символічне диференціювання та інтегрування, наприклад:

```
from sympy import diff, integrate
f = x**2
diff_f = diff(f, x)
integral_f = integrate(f, x)
```

Рис. 2.17 – Приклад обчислень

4. Розв’язування рівнянь і систем рівнянь: SymPy може символічно розв’язувати рівняння та системи рівнянь, наприклад:

```
from sympy import Eq, solve
eq1 = Eq(x**2 - 4, 0)
solutions = solve(eq1)

eq2 = Eq(x + y, 1)
eq3 = Eq(x - y, 2)
solutions = solve((eq2, eq3), (x, y))
```

Рис. 2.18 – Приклад розв’язання рівняння

5. Підтримка матриць і лінійної алгебри: SymPy також підтримує операції матриць і лінійної алгебри, такі як обернення, визначник, власні значення, власні вектори тощо. Наприклад:

```
from sympy import Matrix
m = Matrix([[1, 2], [3, 4]])
m.inv()
m.det()
```

Рис. 2.19 – Приклад матриці

6. Можливості побудови: SymPy також підтримує побудову функцій і виразів, наприклад:

```
from sympy import plot
p = plot(x**2)
```

Рис. 2.20 – Приклад побудови

Це лише кілька прикладів того, що ви можете робити за допомогою SymPy. Бібліотека має набагато більше функцій і можливостей, включаючи підтримку комплексних чисел, логіку та складніші математичні операції, такі як диференціальні рівняння та спеціальні функції.

SymPy — це потужна та багатofункціональна бібліотека для символічної математики, яка широко використовується в наукових та інженерних програмах.

2.6.2 Загальна характеристика бібліотеки NumPy

NumPy — це бібліотека для мови програмування Python, яка додає підтримку великих багатовимірних масивів і матриць разом із великою колекцією математичних функцій високого рівня для роботи з цими масивами. Деякі з його основних функцій включають:

- N-вимірні масиви (ndarrays), які є основною структурою даних, що використовується в numpy;
- Математичні операції над масивами, такі як поелементне додавання, множення тощо;
- Підтримка операцій лінійної алгебри, таких як множення матриць, визначник і інверсія;
- Підтримка перетворення Фур'є та інших складних математичних операцій;
- Підтримка інтеграції з іншими бібліотеками, такими як SciPy і Matplotlib.

Ось кілька прикладів того, що ви можете робити з numpy:

1. Створення масивів: за допомогою numpy можна створювати масиви з різними типами даних, формами та розмірами, наприклад:

```
import numpy as np

# Creating a 1-D array
a = np.array([1, 2, 3])

# Creating a 2-D array
b = np.array([[1, 2], [3, 4]])

# Creating an array with a specific data type
c = np.array([1, 2, 3], dtype='float32')
```

Рис. 2.21 – Приклад масивів

2. Математичні операції. Ви можете виконувати математичні операції над масивами numpy, наприклад:

```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# Element-wise addition
c = a + b

# Element-wise multiplication
d = a * b

# Matrix multiplication
e = np.dot(a, b)
```

Рис. 2.22 – Приклад математичних операцій

3. Операції лінійної алгебри: ви можете виконувати операції лінійної алгебри над масивами numpy, наприклад:

```
a = np.array([[1, 2], [3, 4]])

# Determinant of a matrix
det = np.linalg.det(a)

# Inverse of a matrix
inv = np.linalg.inv(a)
```

Рис. 2.23 – Приклад лінійної алгебри

4. Перетворення Фур'є: Ви можете виконати перетворення Фур'є для масивів numpy, наприклад:

```
a = np.array([1, 2, 3])

# Fourier Transform
fourier = np.fft.fft(a)

# Inverse Fourier Transform
inverse_fourier = np.fft.ifft(fourier)
```

Рис. 2.24 – Приклад перетворення Фур'є

5. Інтеграція з іншими бібліотеками: ви можете використовувати масиви numpy в інших бібліотеках, таких як Matplotlib і SciPy, наприклад:

```
import matplotlib.pyplot as plt

a = np.random.rand(100)
plt.hist(a, bins=20)
plt.show()
```

Рис. 2.25 – Приклад використання інших бібліотек в контексті NumPy

Це лише кілька прикладів того, що ви можете робити з NumPy. Бібліотека має набагато більше функцій і можливостей, включаючи підтримку складніших математичних операцій, індексування та нарізки тощо. Він широко використовується в наукових обчисленнях та аналізі даних і часто використовується як основа для інших бібліотек, таких як SciPy і SymPy.

2.6.3 Загальна характеристика бібліотеки SciPy

SciPy — це бібліотека для мови програмування Python, яка побудована на основі NumPy і надає широкий спектр алгоритмів для оптимізації, обробки сигналів, лінійної алгебри, інтерполяції, інтеграції тощо. Деякі з його основних функцій включають:

- Алгоритми оптимізації, такі як мінімізація, пошук кореня та підгонка кривої;
- Інструменти обробки сигналів, такі як фільтрація, генерація сигналів і обробка зображень;
- Функції лінійної алгебри, такі як розріджені матриці, розв'язувачі розріджених лінійних рівнянь, розв'язувачі власних значень тощо;
- Функції інтерполяції, інтеграції та оптимізації;
- Структури та алгоритми просторових даних;
- Функції статистики.

Ось кілька прикладів того, що можна робити за допомогою SciPy:

1. Оптимізація: Ви можете використовувати функцію `minimize()`, щоб знайти мінімум функції, наприклад:

```
from scipy.optimize import minimize
import numpy as np

def func(x):
    return x**2 + 10*np.sin(x)

x_min = minimize(func, x0=0)
print(x_min)
```

Рис. 2.26 – Приклад оптимізації рівняння

2. Обробка сигналу: Ви можете використовувати функцію `lfilter()`, щоб застосувати фільтр до сигналу, наприклад:

```
from scipy import signal
import numpy as np

b = [1, 0, 0] # numerator
a = [1, -2, 0.5] # denominator
x = np.random.rand(1000)
y = signal.lfilter(b, a, x)
```

Рис. 2.27 – Приклад обробки сигналу

3. Лінійна алгебра: ви можете використовувати розріджений модуль для роботи з розрідженими матрицями, наприклад:

```
from scipy import sparse

A = sparse.rand(100, 100, density=0.05, format='csr')
x = np.random.rand(100)
b = A.dot(x)
```

Рис. 2.28 – Приклад лінійної алгебри

4. Інтерполяція: Ви можете використовувати функцію `interp1d()`, щоб інтерполювати функцію, наприклад:

```
from scipy.interpolate import interp1d
import numpy as np

x = np.linspace(0, 10, num=11, endpoint=True)
y = np.cos(-x**2/9.0)
f = interp1d(x, y)
f2 = interp1d(x, y, kind='cubic')
```

Рис. 2.29 – Приклад інтерполяції

5. Інтеграція: Ви можете використовувати функцію `integrate.quad()`, щоб інтегрувати функцію, наприклад:

```
from scipy import integrate
import numpy as np

def func(x):
    return x**2

result, error = integrate.quad(func, 0, 1)
```

Рис. 2.30 – Приклад інтегралу

6. Статистика: Ви можете використовувати модуль `scipy.stats` для виконання статистичного аналізу, наприклад:

```
from scipy import stats
import numpy as np

data = np.random.normal(5, 2, 100)
mean, var, std = stats.bayes_mvs(data)
```

Рис. 2.31 – Приклад статистичного аналізу

Це лише кілька прикладів того, що ви можете робити за допомогою SciPy. Бібліотека має набагато більше функцій і можливостей, включаючи підтримку складніших математичних операцій, обробки сигналів, обробки зображень тощо. Він широко використовується в наукових обчисленнях, аналізі даних та інженерних додатках.

2.6.4 Загальна характеристика бібліотеки `matplotlib`

`Matplotlib` — це бібліотека графічних зображень для Python, вона надає об'єктно-орієнтований API для вбудовування графіків у програми за допомогою наборів інструментів загального призначення GUI, таких як Tkinter, wxPython, Qt або GTK. Його можна використовувати в поєднанні з іншими бібліотеками, такими як NumPy і SciPy, щоб створити середовище, схоже на середовище MATLAB для наукових і інженерних обчислень. Деякі з його основних функцій включають:

- Підтримка різних типів графіків, таких як лінійні діаграми, точкові діаграми, стовпчасті діаграми, гістограми тощо;
- Параметри налаштування графіків, зокрема стилі ліній, кольори, маркери та легенди;
- Підтримка кількох графіків на одній фігурі та підсхем;
- Підтримка збереження графіків у різних форматах файлів, таких як PNG, PDF, SVG тощо;
- Підтримка рендерингу LaTeX тексту та міток на графіках.

2.7. Програмна реалізація математичної моделі і застосування методу Лагранжа

Програмна реалізація, за стандартом, починається з імпортування необхідних бібліотек. Також ми визначили деякі змінні, необхідні для роботи алгоритму у рамках використаних бібліотек.

```
main.py
1 # This is a sample Python script.
2 import math
3
4 # Press Shift+F10 to execute it or replace it with your code.
5 # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
6 import numpy as np
7 import sympy as sy
8
9 import matplotlib.pyplot as plt
10
11 # x1 = 0
12 # y1 = 0
13
14 # x2 = 2.7
15 # y2 = 0.
16 X_lower, X_upper = 0., 5.
17 Y_lower, Y_upper = 0., 5.
18
19 x1, x2, x3, x4, y1, y2, y3, y4 = sy.symbols('x1 x2 x3 x4 y1 y2 y3 y4', real=True)
20
```

Рис. 2.32 – Перші рядки коду програми. Ініціалізація змінних

Далі ми розпишемо вирази, які часто використовуватимуться у подальшому коді.

```
main.py
19 x1, x2, x3, x4, y1, y2, y3, y4 = sy.symbols('x1 x2 x3 x4 y1 y2 y3 y4', real=True)
20
21 X = [x1, x2, x3, x4]
22 Y = [y1, y2, y3, y4]
23
24 # Define the target function
25 f = ((x1 * y2 - x2 * y1 + x2 * y3 - x3 * y2 + x3 * y4 - x4 * y3 + x4 * y1 - x1 * y4) / 2) ** 2
26
27
28 def norm(var1, var2):
29     return sy.sqrt(var1 ** 2 + var2 ** 2)
30
31
32 def norm2(var1, var2):
33     return var1 ** 2 + var2 ** 2
34
35
36 a, b, c, d = norm((x1 - x2), (y1 - y2)), norm((x1 - x3),
37                                     (y1 - y3)), norm(
38     (x1 - x4), (y1 - y4)), norm((x3 - x2),
```

Рис. 2.33 – Ініціалізація цільової функції

На цьому етапі ми розпишемо цільову функцію та систему обмежень. Вони виглядають таким чином оскільки програмна реалізація не передбачає двосторонніх нерівностей.

```

35
36 a, b, c, d = norm((x1 - x2), (y1 - y2)), norm((x1 - x3),
37             (y1 - y3)), norm(
38             (x1 - x4), (y1 - y4)), norm((x3 - x2),
39             (y3 - y2))
40 a2, b2, c2, d2 = norm2((x1 - x2), (y1 - y2)), norm2((x1 - x3), (
41             y1 - y3)), norm2((x1 - x4), (y1 - y4)), norm2((x3 - x2),
42             (y3 - y2))
43 # Define the constraints:
44 con1 = a2 - 1.2 ** 2
45 con2 = 2.7 ** 2 - a2
46 con3 = b2 - 1.2 ** 2
47 con4 = 2.7 ** 2 - b2
48 con5 = d2 - 1.2 ** 2
49 con6 = 2.7 ** 2 - d2
50 con7 = 1.0 ** 2 - c2
51 con8 = (a + b + d) - 4
52 con9 = 8 - (a + b + d)
53 con10 = 4 - (a + b + c)
54

```

Рис. 2.34 – Ініціалізація системи обмежень

Далі ми спростуємо ці вирази для простішої обробки. Після цього ми утворюємо функції Лагранжа і диференціюємо їх.

```

56 cons = [con1, con2, con3, con4, con5, con6, con7, con8, con9, con10] + [x**2 - X] + [y**2 - Y] + [
57
58
59     y**2 - y - y for y in Y]
60
61 params = (x1, x2, x3, x4, y1, y2, y3, y4)
62
63 partials = [sy.simplify(sy.diff(f, var)) for var in params]
64
65 from scipy.optimize import minimize
66 from sympy.utilities.lambdify import lambdify
67
68
69 def My_lambdify(p, func):
70     f_l = lambdify(params, func)
71     return (lambda x: f_l(*tuple(x)))
72
73
74 Constraints = [{'type': 'ineq', 'fun': My_lambdify(params, con)} for con in cons]
75

```

Рис. 2.35 – Утворення функції Лагранжа

Після цього ми вирішуємо їх методом градієнтного спуску. Таким чином ми знаходимо мінімум цільової функції, яка в даному випадку помножена на -1, оскільки ми в кінці повинні мати максимум.

```
main.py
76 F = My_lambdaify(params, -f) # F to minimize
77
78
79 def check_constraints(points_cord, eps=0.):
80     return all([con["fun"](points_cord) >= (-eps) for con in Constraints])
81
82
83 N_examples = 100000
84 Examples = np.random.random([N_examples, len(params)]) * np.array(
85     [[X_upper - X_lower] * len(X) + [Y_upper - Y_lower] * len(Y)] + np.array([[X_lower] * len(X) + [Y_lower] * len(Y)])
86     Constrained_examples = [ex for ex in Examples if check_constraints(ex)]
87
88     print("N_constrained_examples = ", len(Constrained_examples))
89
90     init_params = min(Constrained_examples, key=F)
91
92     print("check_constraints(init_params) = ", check_constraints(init_params))
93
94     print("init_f = ", F(init_params))
95
```

Рис. 2.36 – Розв'язок та перевірка на локальний екстремум

В результаті ми маємо набір точок, які ми ще можемо зобразити на графіку після запуску програми.

```
main.py
94 print("init_f = ", F(init_params))
95
96 results = minimize(F, init_params, constraints=Constraints)
97
98 print("results = ", results)
99 result_x = results["x"]
100 print("check_constraints(result_x) = ", check_constraints(result_x, eps=0.001))
101
102 N_points = len(result_x) // 2
103 X_res, Y_res = result_x[:N_points], result_x[N_points:]
104
105 fig, ax = plt.subplots()
106 ax.scatter(X_res, Y_res)
107
108 for i in range(N_points):
109     ax.annotate(str(i + 1), (X_res[i], Y_res[i]))
110
111 plt.show()
112
```

Рис. 2.37 – Підготовка точок до візуалізації

3. РОЗДІЛ 3. РЕЗУЛЬТАТИ РОЗРОБКИ ТА АНАЛІЗ ОТРИМАНОГО РІШЕННЯ

3.1. Демонстрація роботи з розробленим рішенням

Програмний застосунок працює доволі простим чином. Він представлений у вигляді простого Python застосунку з головним скриптом main.py. При його запуску виконуються обчислення, які потім відображають результат у вигляді набору точок.

У консолі програма видає результат обчислень:

```
C:\Users\iljuh\PycharmProjects\DiplomaProj\venv\Scripts\python.exe C:\Users\iljuh\PycharmProjects\DiplomaProj\main.py
N_constrained_examples = 145
check_constraints(init_params) = True
init_f = -3.516062636042492
results = message: Optimization terminated successfully
success: True
status: 0
fun: -4.000000070424004
x: [ 3.888e+00  2.609e+00  4.012e+00  4.605e+00  4.201e+00
     4.121e+00  2.205e+00  4.246e+00]
nit: 13
jac: [ 2.491e-01  3.992e+00 -2.491e-01 -3.992e+00 -3.992e+00
       2.491e-01  3.992e+00 -2.491e-01]
nfev: 121
njev: 13
check_constraints(result_x) = True
```

Рис. 3.1 – Приклад розв'язку у консолі

де fun – показує оптимальне рішення, при умові що відповідь надається за модулем.

x: [... , ...] – критерії, за яким оптимальне рішення набуває чинності. Представлений у вигляді масиву з пар координат. Тобто, оптимальним розв'язком буде набір точок {3.888, 2.609}, {4.012, 4.605}, {4.201, 4.121} і {2.205, 4.246}.

Графічно розв'язок представлений на намальованому графі:

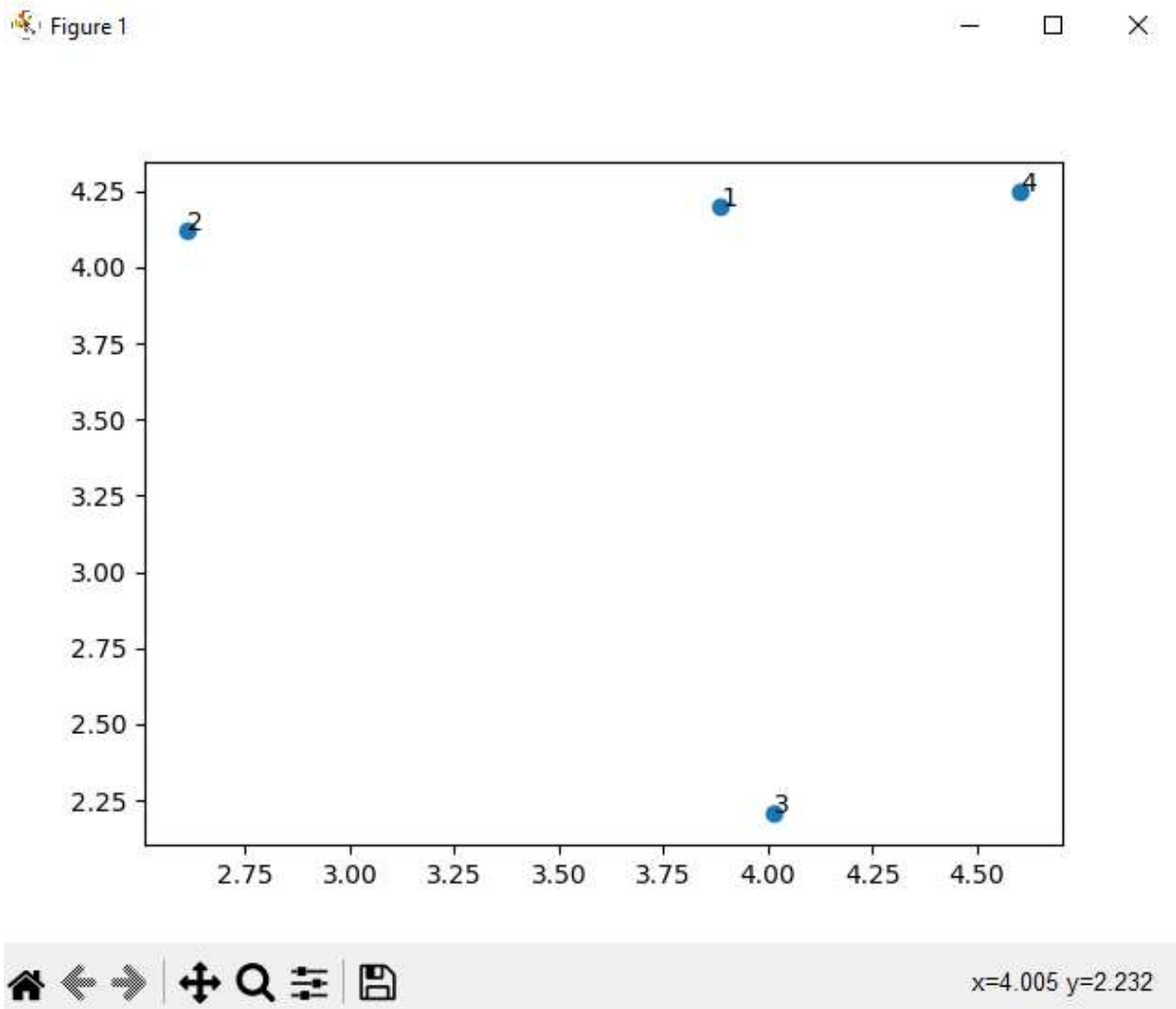


Рис. 2.37 – Візуалізація розв'язку у графічному вигляді

Слід зазначити, що програма надає рішення при сталих верхніх межах, у даному випадку верхньою межею є 5 метрів. Тобто кухонне приміщення у розмірності 5 x 5 метрів матиме приблизно таке розташування ключових робочих поверхонь. Так само можна змінювати розмірність кухні і отримувати різні варіанти вирішення планування.

Крім того, особливістю даного застосунка є можливість задання більш складних форм кухонного приміщення. Для цього треба лише у параметрах верхньої та нижньої границь прописати функції, які описують криві кімнати:

```
55
56 cons = [con1, con2, con3, con4, con5, con6, con7, con8, con9, con10] + [x - X_lower for x in X] + [x**2 + 2*x - x for x in
57 X] + [y - Y_lower for
58 y in Y] + [
59 y**2 + 2*y - y for y in Y]
60
61 rams = (x1, x2, x3, x4, y1, y2, y3, y4)
```

Рис. 3.3 – Зміна верхньої границі на квадратичну функцію для емуляції непрямокутної кімнати

Тут кімната має трохи викривленну стіну. В такому випадку рішення буде інакшим:

```
C:\Users\iljuh\PycharmProjects\DiplomaProj\venv\Scripts\python.exe C:\Users\iljuh\PycharmProjects\DiplomaProj\main.py
N_constrained_examples = 175
check_constraints(init_params) = True
init_f = -3.8519314382236005
results = message: Optimization terminated successfully
success: True
status: 0
fun: -4.000000016698152
x: [ 3.699e+00  2.803e+00  2.142e+00  4.057e+00  1.324e+00
     2.111e-01  2.579e+00  1.769e+00]
nit: 7
jac: [-3.115e+00  2.509e+00  3.115e+00 -2.509e+00  2.509e+00
       3.115e+00 -2.509e+00 -3.115e+00]
nfev: 66
njev: 7
check_constraints(result_x) = True
```

Рис. 3.4 – Розв'язок, який змінився відповідно до нових умов

Що у графічному представленні виглядає приблизно так:

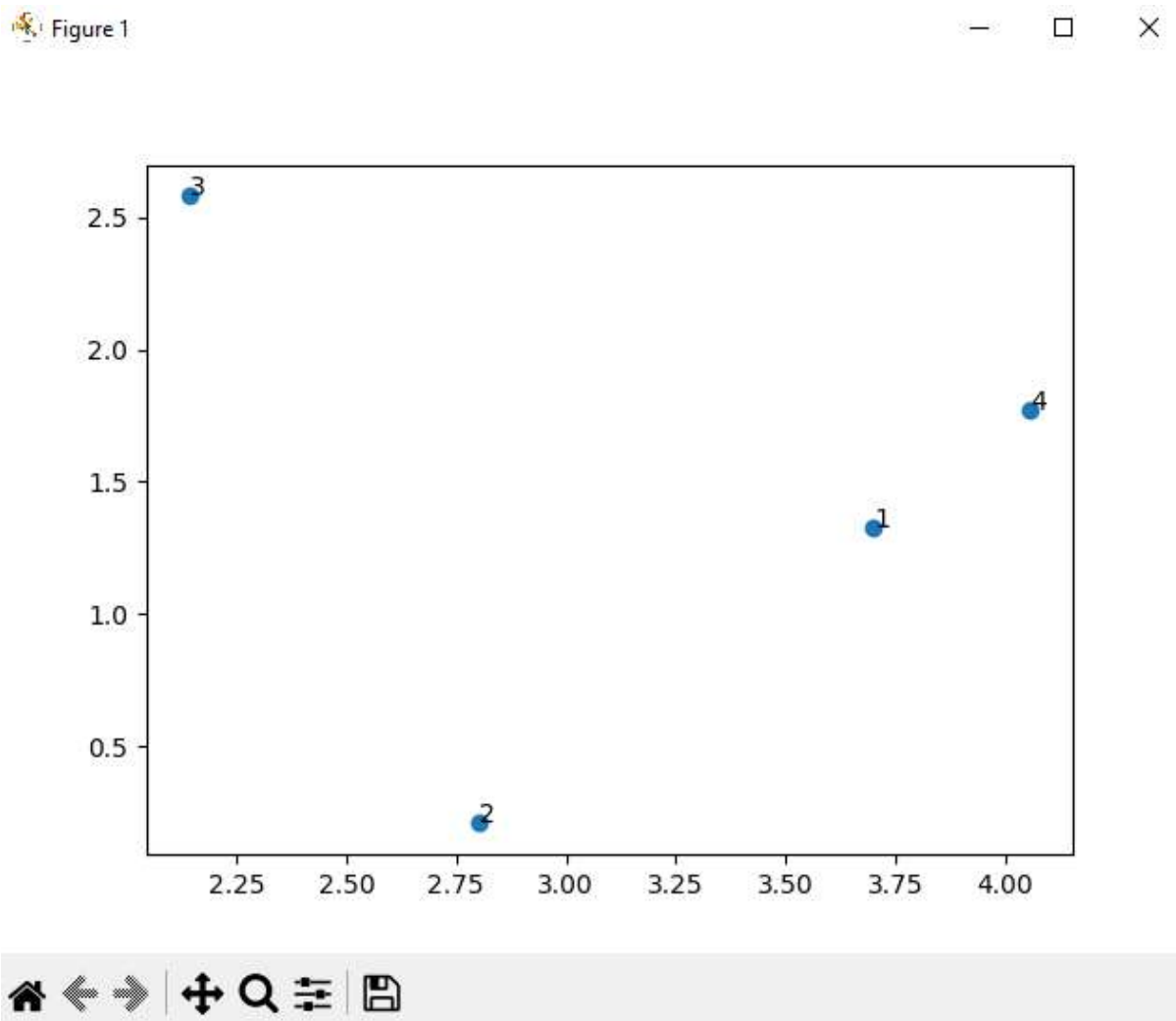


Рис. 3.5 – Графічне представлення нового розв'язку

Таким чином ми отримуємо оптимальні рішення для розташування основних утиліт, які згодом можна візуалізувати у своєму проекті AutoCAD чи інших програм з розробки дизайнів інтер'єру.

3.2. Огляд розробленого рішення і обґрунтування його доцільності

Даний застосунок, використовуючи ергономічні концепції, має на меті оптимізувати процес планування розташування кухонного приміщення шляхом надання рекомендації відповідно до плану кухні. Саме тому застосунок, замість надання альтернативи існуючим аналогам, які візуалізують існуючі плани кухонь, є більше споміжною надбудовою.

Для доведення доцільності даного рішення слід звернутися до досліджень безпосередніх концепцій, оскільки розробка є нічим іншим, як їх програмною реалізацією.

Є деякі емпіричні докази, які підтверджують ефективність правила золотого трикутника в оптимізації робочого процесу на кухні та покращенні ефективності та ергономіки планування кухні. Одне дослідження, опубліковане в *Journal of Environmental Psychology*, вивчало вплив правила золотого трикутника на ефективність і приємність планування кухні (Lin, Chen, & Chen, 2011). Дослідження показало, що учасники оцінили планування кухні з конфігурацією золотого трикутника як більш ефективне та приємне порівняно з плануванням із приладами, розташованими по прямій лінії.

Інше дослідження, опубліковане в журналі *Ergonomics*, оцінювало вплив правила золотого трикутника на ефективність і ергономічність планування кухні (Monderen, Bellemans, & De Witte, 2000). Дослідження показало, що планування кухні з конфігурацією золотого трикутника призвело до меншої кількості рухів і менше часу, витраченого на пошук інструментів і обладнання, порівняно з плануванням із приладами, розташованими по прямій лінії.

Існують також докази практичного досвіду та галузевих рекомендацій, які підтримують використання правила золотого трикутника в дизайні кухні. Наприклад, Національна асоціація кухні та ванної кімнати (NKBA), професійна асоціація дизайну кухні та ванної кімнати, рекомендує правило золотого трикутника як принцип для оптимізації робочого процесу на кухні.

Загалом емпіричні дані свідчать про те, що правило золотого трикутника може бути ефективним принципом для оптимізації робочого процесу на кухні та підвищення ефективності та ергономіки планування кухні. Однак важливо зазначити, що правило золотого трикутника слід розглядати в контексті інших факторів дизайну, таких як розмір і форма кухні, робочий процес користувача, а також зберігання та організація інструментів і обладнання в щоб створити ефективне та ергономічне планування кухні.

3.3. Можливі майбутні вдосконалення

Потенційно можна розглядати декілька шляхів для вдосконалення даного застосунку, а саме:

1. Розширення функціоналу програмної реалізації. Оскільки застосунок загалом використовується як надбудова до існуючих CAD, є доцільним розробка аналогічного програмного забезпечення, який вже інтегрує застосунок як одну із своїх функцій. Також можна реалізувати функціонал, який під рекомендації може підібрати пропозиції, які існують на ринку, з мінімізацією витрат існуючих ресурсів.
2. Альтернативні математичні моделі. Дана математична модель є оптимізаційною задачею, ціллю якої є максимізація можливого робочого простору з урахуванням ергономічних концепцій. Як варіант, можливо розробити модель, яка мінімізує кількість витрачених кроків при виконанні набору завдань. Також існуючу модель можна розширити, впровадивши додаткові критерії оптимальності, які не були враховані у побудові моделі.
3. Альтернативні оптимізаційні методи. Метод Лагранжа є доволі універсальним, що може свідчити про існування вузько направлених

методів, які можуть пришвидшити вирішення існуючої оптимізаційної задачі.

ВИСНОВКИ

Підсумовуючи, ремонт кухні – це процес оновлення та покращення дизайну, функціональності та ефективності кухні. При ремонті кухні необхідно враховувати багато факторів, включаючи планування кухні, використовувані матеріали, а також техніку та сантехніку. Планування кухні є особливо важливим аспектом ремонту, оскільки воно визначає хід роботи та організацію інструментів та обладнання на кухні.

Правило золотого трикутника — це широко використовуваний принцип у дизайні кухні, який стверджує, що мийка, холодильник і плита повинні бути розташовані у формі трикутника, з відстанню між кожною з цих точок не більше 4,5 метрів (15 футів). Ідея золотого трикутника полягає в тому, щоб оптимізувати процес роботи на кухні, розмістивши прилади та інструменти, які найчастіше використовуються, у межах легкої досяжності один від одного. Існують деякі емпіричні дані, які свідчать про те, що правило золотого трикутника може призвести до більш ефективного та ергономічного планування кухні, хоча важливо враховувати інші фактори, такі як розмір і форма кухні, робочий процес користувача, а також зберігання та зберігання. організація інструментів та обладнання.

Окрім планування кухні, інші фактори, які слід враховувати під час ремонту кухні, включають використані матеріали, такі як шафи, стільниці та підлога, а також прилади та приладдя, такі як холодильники, духовки та раковини. Ремонт кухні також може включати такі міркування, як безпека харчових продуктів, енергоефективність і впровадження нових технологій, таких як розумні прилади та системи автоматизації.

Загалом, ремонт кухні – це складний процес, який вимагає ретельного планування та врахування широкого спектру факторів, щоб створити функціональний, ефективний і приємний кухонний простір.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. I. Azevedo, H. Albano, J. Silva, P. Teixeira, Food safety in the domestic environment, *Food Control*, 37 (2014), pp. 272-276
2. J. Beamish, R.K. Parrott, J. Emmel, M.J. Peterson, *Kitchen planning: Guidelines, codes, standards*, Wiley, Hoboken, New Jersey (2013)
3. Susan Maney Lovett, "Kitchen Planning: Remodeling, Renovation, and Design"
4. Elizabeth A. Scott, "The impact of kitchen design on food safety", *Journal of Food Protection*
5. Elizabeth A. Scott, "Energy-efficient appliances and food safety", *Journal of Food Protection*
6. Методи розв'язання задач нелінійного програмування, веб-ресурс, <https://studfile.net/preview/5470183/page:21/>
7. Документація Python, веб-ресурс, <https://www.python.org/doc/>
8. Документація SymPy, веб-ресурс, <https://docs.sympy.org/latest/index.html>
9. Документація NumPy, веб-ресурс, <https://numpy.org/doc/>
10. Документація Matplotlib, веб-ресурс, <https://matplotlib.org/stable/index.html>
11. Дослідження операцій. Конспект лекцій / Уклад.: О.І. Лисенко, І.В. Алексеєва, – К: НТУУ «КПІ», 2016. – 196 с.
12. Математичні методи дослідження операцій : підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с.
13. Математичні методи дослідження операцій. Лінійне програмування. Частина 1 : навчальний посібник / А. А. Яровий, Л. М. Ваховська, Л. В. Крилик. – Вінниця : ВНТУ, 2020. – 86 с.

14. HERZOG & DE MEURON CAIXA FORUM MADRID [Електронний ресурс] – Режим доступу до ресурсу: <https://divisare.com/projects/370386-herzog-demeuron-simon-garcia-arqfoto-caixa-forum-madrid>.
15. Historia łódzkiej Manufaktury – od przetwórstwa bawełny po jedno z największych centrów handlowych w Polsce [Електронний ресурс] – Режим доступу до ресурсу: <https://magnifier.pl/manufaktura-lodz-historia/>.
16. Moritzburg Museum Extension by Nieto Sobejano Arquitectos [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dezeen.com/2011/06/17/moritzburgmuseum-extension-by-nieto-sobejanoarquitectos/>.
17. Moritzburg Museum Extension, Галле, Германия [Електронний ресурс] – Режим доступу до ресурсу: http://www.redeveloper.ru/redeveloperskieproekty/realise_actual/moritzburg-museumextension-halle-germany/.
18. Nowe Muzeum w Berlinie [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: https://wroclaw.wyborcza.pl/wroclaw/56,357_71,19480903,2011-nowe-muzeum-wberlinie,,11.html?disableRedirects=true.
19. Ustawa o rewitalizacji a ochrona zabytków – dwa światy czy efekt synergii? [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <http://www.tonz.org.pl/lodz/2282/ustawao-rewitalizacji-a-ochrona-zabytkow-dwa-swiatyczy-efekt-synergii>.
20. Бірюк С. П. Основні напрямки реконструкції міських промислових територій [Електронний ресурс] / С. П. Бірюк, А. М. Плешкановська // Містобудування та територіальне планування. – 2010. – Вип. 37. – С. 68–72. – Режим доступу : http://nbuv.gov.ua/UJRN/MTP_2010_37_14.
21. Гайко Ю. І. Особливості управління комплексною реконструкцією житлової забудови на сучасному етапі розвитку міст / Ю. І. Гайко, Е. А. Шишкін, Є. Ю. Гнатченко, В. В. Сливчук // Містобудування та

- територіальне планування : наук-техн. Збірник / Відпов. ред. М. М. Осетрін. – Київ : КНУБА, 2018. – Вип. 66. – С. 90–100.
22. Гнатченко Є. Ю. Економіко-математичне моделювання процесів проектного фінансування в девелопменті / Є. Ю. Гнатченко, Ю. І. Гайко // Науковий вісник Херсонського державного університету. Серія «Економічні науки». – Херсон, ХДУ, 2018. – Вип. 31. – С. 198–202.
23. «Про комплексну реконструкцію кварталів (мікрорайонів) застарілого житлового фонду» : Закон України № 525–V від 22.12.06. [Електронний ресурс] // Відомості Верховної Ради України (ВВР). Ст. 88 – 2007. – № 10. – Режим доступу : <https://zakon.rada.gov.ua/laws/show/525-16.40>. Зіміна О. С. Нові технології в дизайні індустриальних фасадів [Електронний ресурс]. // Архітектурний вісник КНУБА. – 2013. – Вип. 1. – С. 315–322. – Режим доступу : <https://sites.google.com/site/arhviknuba/vipuski>.
24. Проблеми та перспективи розвитку житлової забудови в умовах комплексної реконструкції міста : монографія / [Ю. І. Гайко, Т. В. Жидкова, Е. А. Шишкін та ін. ; за заг. ред. Ю. І. Гайка, Т. В. Жидкової] ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2019. – 246 с

ДОДАТКИ

Main.py

```
# Press Shift+F10 to execute it or replace it with your code.# Press Double Shift to search
everywhere for classes, files, tool windows, actions, and settings.import numpy as npimport
sympy as syimport matplotlib.pyplot as plt
from scipy.optimize import minimizefrom sympy.utilities.lambdify import lambdify

# x1 = 0# y1 = 0# x2 = 2.7# y2 = 0.X_lower, X_upper = 0., 2.Y_lower, Y_upper = 0., 2.x1,
x2, x3, x4, y1, y2, y3, y4 = sy.symbols('x1 x2 x3 x4 y1 y2 y3 y4', real=True)X = [x1, x2, x3,
x4]Y = [y1, y2, y3, y4]# Define the target functionf = ((x1 * y2 - x2 * y1 + x2 * y3 - x3 *
y2 + x3 * y4 - x4 * y3 + x4 * y1 - x1 * y4) / 2) ** 2

def norm(var1, var2):    return sy.sqrt(var1 ** 2 + var2 ** 2)def norm2(var1, var2):    return
var1 ** 2 + var2 ** 2

a, b, c, d = norm((x1 - x2), (y1 - y2)), norm((x1 - x3),
(y1 - y3)), norm((x1 - x4), (y1 - y4)), norm((x3 - x2), (y3 -
y2))a2, b2, c2, d2 = norm2((x1 - x2), (y1 - y2)), norm2((x1 - x3), (
y1 - y3)), norm2((x1 - x4), (y1 - y4)), norm2((x3 - x2), (y3
- y2))

# Define the constraints:con1 = a2 - 1.2 ** 2con2 = 2.7 ** 2 - a2con3 = b2 - 1.2 ** 2con4
= 2.7 ** 2 - b2con5 = d2 - 1.2 ** 2con6 = 2.7 ** 2 - d2con7 = 1.0 ** 2 - c2con8 = (a + b +
d) - 4con9 = 8 - (a + b + d)con10 = 4 - (a + b + c)
```

```

cons = [con1, con2, con3, con4, con5, con6, con7, con8, con9, con10] + [x - X_lower for
x in X] + [X_upper - x for x in X] + [y - Y_lower for y in Y] + [Y_upper - y for y in
Y]
params = (x1, x2, x3, x4, y1, y2, y3, y4)
partials = [sy.simplify(sy.diff(f, var)) for var in
params]

```

```

def My_lambdaify(p, func):    f_l = lambdaify(params, func)    return (lambda x:
f_l(*tuple(x)))

```

```

Constraints = [{'type': 'ineq', 'fun': My_lambdaify(params, con)} for con in cons]
F = My_lambdaify(params, -f) # F to minimize

```

```

def check_constraints(points_cord, eps=0.):    return all([con["fun"](points_cord) >= (-eps)
for con in Constraints])

```

```

N_examples = 100000
Examples = np.random.random([N_examples, len(params)]) *
np.array( [[X_upper - X_lower] * len(X) + [Y_upper - Y_lower] * len(Y)]) +
np.array([[X_lower] * len(X) + [Y_lower] * len(Y)])
Constrained_examples = [ex for ex in
Examples if check_constraints(ex)]
print("N_constrained_examples = ",
len(Constrained_examples))
init_params = min(Constrained_examples,
key=F)
print("check_constraints(init_params) = ", check_constraints(init_params))
print("init_f = ", F(init_params))
results = minimize(F, init_params, constraints=Constraints)
print("results = ", results)
result_x = results["x"]
print("check_constraints(result_x) = ",
check_constraints(result_x, eps=0.001))
N_points = len(result_x) // 2
X_res, Y_res = result_x[:N_points], result_x[N_points:]
fig, ax = plt.subplots()
ax.scatter(X_res, Y_res)
for i in range(N_points):
    ax.annotate(str(i + 1), (X_res[i], Y_res[i]))
plt.show()

```



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення



МАГІСТЕРСЬКА РОБОТА
«Оптимізація процесу реновації
кухонного приміщення житлового помешкання з
використанням моделей і методів дослідження операцій»

Виконав: студент групи ПДМ – 61 Кузнецов Ілля Іванович

Керівник: Жебка В.В., д.т.н., доцент, завідувач кафедри Технологій цифрового розвитку.

Київ - 2022

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

2

Мета роботи: Оптимізація процесу реновації кухонного приміщення житлового помешкання з використанням моделей і методів дослідження операцій

Об'єкт дослідження: процес реновації кухонного приміщення

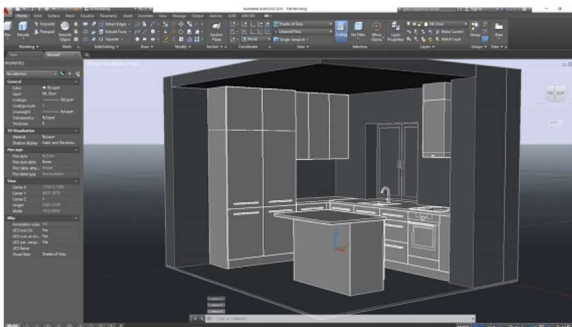
Предмет дослідження: методи та засоби дослідження операцій

ІСНУЮЧІ РІШЕННЯ ПЛАНУВАННЯ КУХНІ

3

AutoCAD

- Програма CAD професійного рівня з широким набором інструментів і функцій для створення точних 2D і 3D проектів
- Підходить для комплексного ремонту та нестандартного планування кухні
- Широко використовується в різних галузях, включаючи архітектуру та дизайн інтер'єру



ІСНУЮЧІ РІШЕННЯ ПЛАНУВАННЯ КУХНІ

4

- Безкоштовна програма CAD з відкритим вихідним кодом, спеціально розроблена для проектування та ремонту будинків
- Набір інструментів для створення 2D і 3D планів поверхів
- Дозволяє користувачам візуалізувати свої проекти в 3D з реалістичним освітленням і матеріалами



ПРОБЛЕМА ІСНУЮЧИХ РІШЕНЬ

5

- Відсутність будь-якої перевірки дизайну на предмет ергономічності
- Відсутність рекомендацій розташування утиліт коли не знаєш, з чого почати
- Відсутність перевірки на дотримання санітарних норм

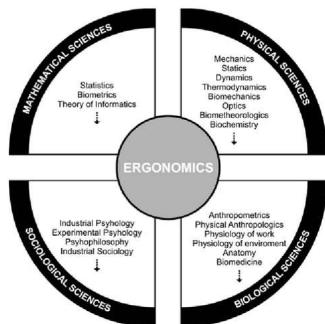


Figure 2. Interdisciplinarity of ergonomics.

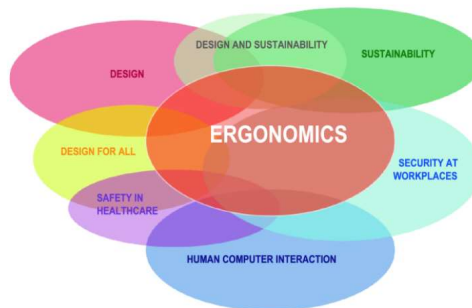
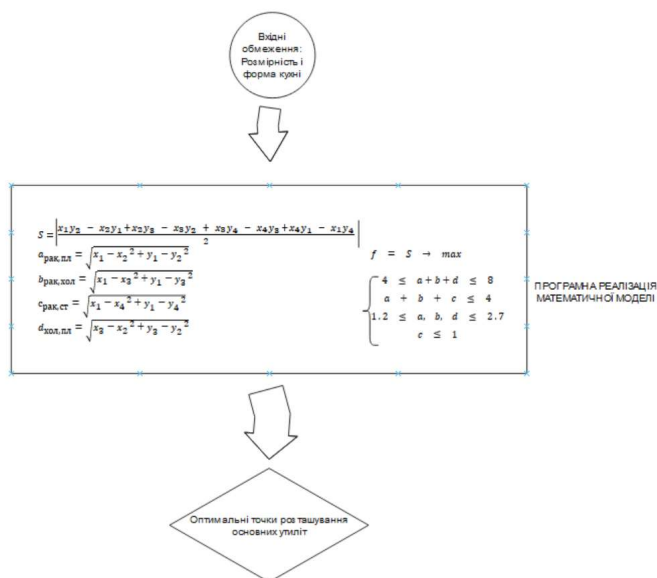


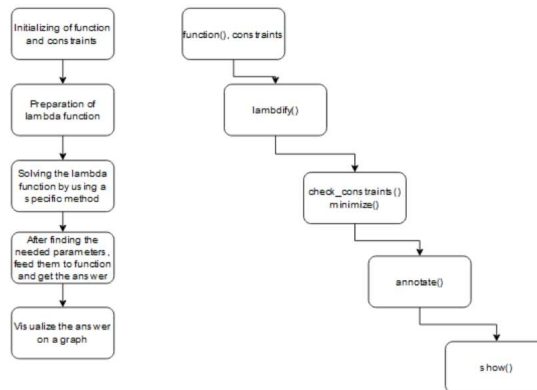
СХЕМА РОБОТИ ПРОГРАМНОГО ЗАСТОСУНКУ

6



ДІАГРАМА ДІЯЛЬНОСТІ ПРОЦЕСУ ОПТИМІЗАЦІЇ

7



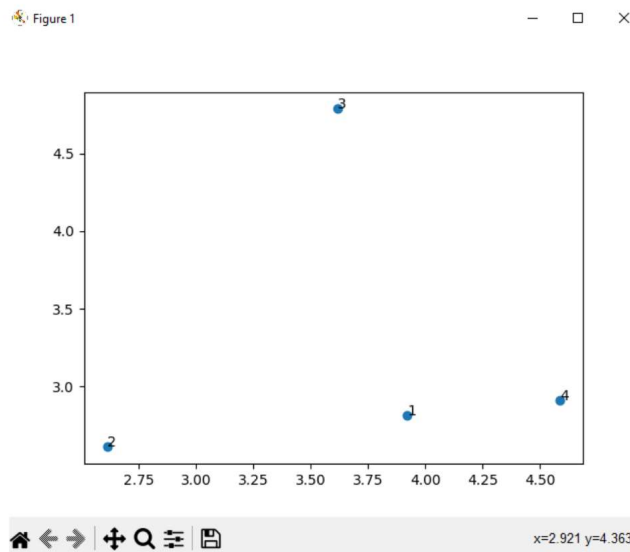
ПРИКЛАД РЕЗУЛЬТАТІВ ОПТИМІЗАЦІЇ МАТЕМАТИЧНОЇ МОДЕЛІ

8

```
N_constrained_examples = 175
check_constraints(init_params) = True
init_f = -3.8519314382236005
results = message: Optimization terminated successfully
success: True
status: 0
  fun: -4.000000016698152
    x: [ 3.699e+00  2.803e+00  2.142e+00  4.057e+00  1.324e+00
        2.111e-01  2.579e+00  1.769e+00]
  nit: 7
  jac: [-3.115e+00  2.509e+00  3.115e+00 -2.509e+00  2.509e+00
        3.115e+00 -2.509e+00 -3.115e+00]
  nfev: 66
  njev: 7
check_constraints(result_x) = True
```

ПРИКЛАД ВІЗУАЛІЗАЦІЇ РІШЕННЯ

9



ВИСНОВКИ

10

1. Проведено аналіз існуючих рішень процесу реновації кухонних приміщень, складені вимоги до додатку, що розробляється.
2. Досліджено емпіричні дані, створені на їх основі ергономічні концепції. Перевірено ці концепції на актуальність та доцільність оптимізації.
3. Створено математичну модель для оптимізації процесу реновації кухонного приміщення.
4. Розглянуто основні методи дослідження операцій. Вивчено мову Python та бібліотеки NumPy, SymPy, SciPy, Matplotlib для реалізації математичної моделі та методів оптимізації.
5. Розроблено застосунок для знаходження та візуалізації рішення задачі оптимізації процесу реновації кухонного приміщення.

Статті:

1. Оптимізація процесу реновації кухонного приміщення житлового помешкання з використанням моделей і методів дослідження операцій// ТІТ. №3, 2022, с. 31-35

Тези доповідей на конференціях:

1. Кузнецов І.І. Концепція золотого трикутника для забезпечення оптимальності плану дизайну кухні з точки зору функціональності // XV Науково-технічна конференція «Сучасні інфокомунікаційні технології» . – Київ: ДУТ, 2022. – с. 91-92

ДЯКУЮ ЗА УВАГУ!