

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Тема: Метод автоматизованого підбору тимчасового житла для
категорій споживачів за генетичним алгоритмом

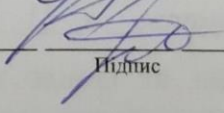
Галузь знань: 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність: 122 – Комп'ютерні науки
Шифр і назва спеціальності

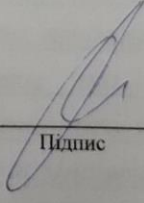
Освітня програма: Комп'ютерні науки
Назва освітньої програми

Виконала: студентка 2 курсу, група КНм-22-1  Ю.В. Собкова
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: к.ф.-м.н., доцент кафедри КН  В.І. 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
Науковий ступінь, посада Підпис Ініціали, прізвище

12 грудня 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: Інформаційних технологій

Кафедра: Комп'ютерних наук

Освітній ступінь: Магістр

Галузь знань: 12 – Інформаційні технології

Спеціальність: 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


Підпис

д.т.н., професор Бармак О. В.

« 01 » вересня 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом».
2. Завдання видано студентці: Собковій Юлії Василівні
Прізвище, ім'я, по батькові
3. Керівник роботи: к.ф.-м.н., доцент Міхалевський Віталій Цезарійович
Посада, Прізвище, ім'я, по батькові
4. Затверджено наказом університету від « 15 » серпня 2023 р. № 30
5. Зміст пояснювальної записки (перелік задач) та вихідні дані: Мета кваліфікаційної роботи магістра – розробка методу підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Метод має забезпечити знаходження послідовності множин варіантів вибору житла з найбільшою відповідністю зазначеним вимогам та враховувати зміни параметрів. Потрібно вирішити проблеми ефективного підбору тимчасового житла для категорій споживачів, використовуючи множину варіантів житла. Передбачаються вхідні дані про вимоги до категорій та характеру житла, його властивості та характеристики, вагові коефіцієнти пріоритетності вибору, відслідковування стану роботи. Також необхідно порівняти ефективність роботи розробленого методу з існуючими рішеннями завдань підбору тимчасового житла для різних категорій споживачів.

Реферат

Кваліфікаційна робота магістра вирішує науково-технічну задачу автоматизованого підбору тимчасового житла для категорій споживачів. Результатом роботи є розроблений метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом, що дозволяє обирати відповідним чином житло користувачу.

Актуальність теми. В останні роки цифровізація змінила пошук житла. Сьогодні онлайн-платформи полегшують обмін інформацією про ринок житла та розширюють читабельність ринку житла для продавців, покупців, орендодавців та орендарів. Такі платформи можуть демократизувати доступ до інформації та урізноманітнити інформаційне забезпечення шукачів житла. Це, у свою чергу, може розширити параметри вибору, збільшити радіус підбору, зменшити витрати на пошук і відійти від традиційних підходів, щоб допомогти шукачам житла реалізувати ефективніший пошук з кращим результатом. У цій кваліфікаційній роботі досліджується, як саме певна онлайн-платформа формує пошук житла, впливаючи на постачання інформації, представлення та споживання. Протиріччя виникають, коли старі гейткіпери розробляють нові стратегії збереження влади в цифровій сфері, а нові гейткіпери з'являються, щоб отримати вигоду від цифрових тенденцій. Політики можуть зіграти важливу роль у збереженні та розвитку суспільних переваг цифровізації ринку житла, одночасно пом'якшуючи її шкоду.

Процес підбору тимчасового житла завжди залишається важливою задачею для багатьох людей, оскільки вони можуть шукати нове місце проживання з різних причин, таких як зміна роботи, створення сім'ї, покращення житлових умов, і багато інших. Люди постійно змінюють свої потреби щодо житла. Завдяки зміні житлових обставин, таких як одруження, народження дітей та інше, вони можуть шукати житло, яке відповідає їхнім новим потребам. Люди постійно мігрують і переїжджають в нові регіони або міста з різних причин. Це може створювати попит на житло в нових місцях. Багато людей вважають нерухомість важливим видом інвестицій. Особи, які шукають можливості для інвестування грошей, можуть шукати нерухомість для покупки або оренди. З ростом населення зростає і потреба в житлі. Особливо це стосується мегаполісів та місць з інтенсивною інфраструктурною забудовою. Стан ринку нерухомості

також може впливати на актуальність підбору тимчасового житла. Зміни в цінах на житло, доступність кредитів та інші ринкові чинники можуть впливати на рішення щодо придбання чи оренди житла.

В умовах війни актуальність підбору тимчасового житла суттєво зростає. Особливо важливо знайти тимчасове житло. У цьому контексті виникає актуальне завдання, яке полягає в розробці та впровадженні методу автоматизованого підбору тимчасового житла для різних категорій споживачів.

Мета і задачі роботи. Мета кваліфікаційної роботи магістра полягає у розробці методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз предметної області та відомих підходів до організації підбору тимчасового житла;
- вдосконалити інформаційну модель підбору тимчасового житла для категорій споживачів;
- розробити метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом;
- підготувати набір даних для застосування у генетичному алгоритмі;
- використати генетичний алгоритм для автоматизованого підбору тимчасового житла для категорій споживачів;
- провести функціональне та прикладне дослідження ефективності розробленого методу, в тому числі для тестового аналізу автоматизованого підбору тимчасового житла для категорій споживачів.

Об’єкт дослідження – процес підбору тимчасового житла для категорій споживачів.

Предмет дослідження – моделі, методи та засоби автоматизації процесу підбору тимчасового житла для категорій споживачів.

Методи дослідження, застосовані для вирішення поставлених завдань: для організації підбору тимчасового житла використовуються основні положення методів аналізу даних і теорії множин, а для реалізації інформаційної системи – методології проектування інформаційних систем і об’єктно-орієнтований підхід.

Наукова новизна одержаних результатів. В результаті роботи були отримані наступні результати:

– вдосконалено інформаційну модель підбору тимчасового житла для категорій споживачів, яка відрізняється тим, що містить формальне подання всіх необхідних сутностей для операцій підбору тимчасового житла для категорій споживачів;

– розроблено метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом, що дозволяє за множиною наявних параметрів житла, початкових та проміжних вимог до житла, апаратно-технічних засобів автоматично визначати множину можливих варіантів тимчасового житла та визначити оптимальний, що збільшує ефективність підбору тимчасового житла і забезпечує рівень задоволення від отриманого результату в середньому на 12,5%.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні наукові й практичні результати кваліфікаційної роботи магістра доповідались на XV Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2023» (17-18 листопада 2023 року) у доповіді на тему «Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом»; за темою роботи автором виконано наукову публікацію: Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К. Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». - Хмельницький, 2023. - С. 286-288.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновку, переліку посилань із 45 найменувань та 5 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 117 сторінки, з них 97 сторінок основного тексту та 20 сторінок додатків. У роботі наведено 45 рисунків та 14 таблиць.

Ключові слова: тимчасове житло, категорії споживачів, генетичний алгоритм, Python, аналіз даних, програмне забезпечення, модель методу, інформаційна технологія.

ЗМІСТ

Перелік скорочень	3
Вступ.....	4
Розділ 1. Аналіз предметної області підбору тимчасового житла та постановка задачі.....	7
1.1 Аналіз предметної області.....	7
1.2 Аналіз існуючих наукових публікацій про підбір житла на основі генетичного алгоритму	12
1.3 Аналіз моделей підбору тимчасового житла та їх рішень	18
1.4 Постановка задачі	24
Висновки для Розділу 1	25
Розділ 2. Розробка методу автоматизованого пошуку житла для категорій споживачів за генетичним алгоритмом.....	26
2.1 Модель оптимізації пошуку житла	26
2.2 Метод генетичного алгоритму для пошуку житла	33
2.3 Інформаційна модель методу пошуку житла	42
Висновки до Розділу 2	45
Розділ 3. Програмна реалізація методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.....	46
3.1 Вимоги до розробки програмного забезпечення.....	46
3.2 Програмна реалізація моделі методу автоматизованого підбору житла	57
3.3 Обґрунтування вибору середовища розробки	70
Висновки до Розділу 3	75
Розділ 4. Дослідження методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом	76
4.1 Підготовка даних для програмних компонентів методу та прикладне тестування методу	76
4.2 Функціональне дослідження та визначення ефективності методу підбору тимчасового житла для категорій споживачів	82
Висновки до розділу 4	89
Загальні висновки.....	90
Перелік посилань.....	92
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
ГА	Генетичний алгоритм
ПЗ	Програмне забезпечення
SQL	Structured Query Language
API	Application programming interface
БД	База даних
ООП	Об'єктно-орієнтоване програмування
ООМІС	Об'єктно-орієнтоване програмування моделей інформаційної системи
DML	Data Manipulation Language
DDL	Data Definition Language
UI	User Interface
UML	Unified Modeling Language
PyPI	Python Package Index
ОС	Операційна система

Вступ

Актуальність теми. В останні роки цифровізація змінила пошук житла. Сьогодні онлайн-платформи полегшують обмін інформацією про ринок житла та розширюють читабельність ринку житла для продавців, покупців, орендодавців та орендарів. Такі платформи можуть демократизувати доступ до інформації та урізноманітнити інформаційне забезпечення шукачів житла. Це, у свою чергу, може розширити параметри вибору, збільшити радіус підбору, зменшити витрати на пошук і відійти від традиційних підходів, щоб допомогти шукачам житла реалізувати ефективніший пошук з кращим результатом. У цій кваліфікаційній роботі досліджується, як саме певна онлайн-платформа формує пошук житла, впливаючи на постачання інформації, представлення та споживання. Протиріччя виникають, коли старі гейткіпери розробляють нові стратегії збереження влади в цифровій сфері, а нові гейткіпери з'являються, щоб отримати вигоду від цифрових тенденцій. Політики можуть зіграти важливу роль у збереженні та розвитку суспільних переваг цифровізації ринку житла, одночасно пом'якшуючи її шкоду.

Процес підбору тимчасового житла завжди залишається важливою задачею для багатьох людей, оскільки вони можуть шукати нове місце проживання з різних причин, таких як зміна роботи, створення сім'ї, покращення житлових умов, і багато інших. Люди постійно змінюють свої потреби щодо житла. Завдяки зміні житлових обставин, таких як одруження, народження дітей та інше, вони можуть шукати житло, яке відповідає їхнім новим потребам. Люди постійно мігрують і переїжджають в нові регіони або міста з різних причин. Це може створювати попит на житло в нових місцях. Багато людей вважають нерухомість важливим видом інвестицій. Особи, які шукають можливості для інвестування грошей, можуть шукати нерухомість для покупки або оренди. З ростом населення зростає і потреба в житлі. Особливо це стосується мегаполісів та місць з інтенсивною інфраструктурною забудовою. Стан ринку нерухомості також може впливати на актуальність підбору тимчасового житла. Зміни в цінах

на житло, доступність кредитів та інші ринкові чинники можуть впливати на рішення щодо придбання чи оренди житла.

В умовах війни актуальність підбору тимчасового житла суттєво зростає. Особливо важливо знайти тимчасове житло. У цьому контексті виникає актуальне завдання, яке полягає в розробці та впровадженні методу автоматизованого підбору тимчасового житла для різних категорій споживачів.

Мета і задачі роботи. Мета кваліфікаційної роботи магістра полягає у розробці методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз предметної області та відомих підходів до організації підбору тимчасового житла;
- вдосконалити інформаційну модель підбору тимчасового житла для категорій споживачів;
- розробити метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом;
- підготувати набір даних для застосування у генетичному алгоритмі;
- використати генетичний алгоритм для автоматизованого підбору тимчасового житла для категорій споживачів;
- провести функціональне та прикладне дослідження ефективності розробленого методу, в тому числі для тестового аналізу автоматизованого підбору тимчасового житла для категорій споживачів.

Об’єкт дослідження – процес підбору тимчасового житла для категорій споживачів.

Предмет дослідження – моделі, методи та засоби автоматизації процесу підбору тимчасового житла для категорій споживачів.

Методи дослідження, застосовані для вирішення поставлених завдань: для організації підбору тимчасового житла використовуються основні положення методів аналізу даних і теорії множин, а для реалізації інформаційної системи – методології проектування інформаційних систем і об’єктно-орієнтований підхід.

Наукова новизна одержаних результатів. В результаті роботи були отримані наступні результати:

- вдосконалено інформаційну модель підбору тимчасового житла для категорій споживачів, яка відрізняється тим, що містить формальне подання всіх необхідних сутностей для операцій підбору тимчасового житла для категорій споживачів;
- розроблено метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом, що дозволяє за множиною наявних параметрів житла, початкових та проміжних вимог до житла, апаратно-технічних засобів автоматично визначати множину можливих варіантів тимчасового житла та визначити оптимальний, що збільшує ефективність підбору тимчасового житла і забезпечує рівень задоволення від отриманого результату в середньому на 12,5%.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні наукові й практичні результати кваліфікаційної роботи магістра доповідались на XV Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2023» (17-18 листопада 2023 року) у доповіді на тему: «Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом»; за темою роботи автором виконано наукову публікацію: Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К. Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». - Хмельницький, 2023. - С. 286-288.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновку, переліку посилань із 45 найменувань та 5 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 117 сторінки, з них 97 сторінок основного тексту та 20 сторінок додатків. У роботі наведено 45 рисунків та 14 таблиць.

Розділ 1. Аналіз предметної області підбору тимчасового житла та постановка задачі

1.1 Аналіз предметної області

Аналіз предметної області для подальшої розробки методу підбору тимчасового житла для категорій споживачів – це завдання створення єдиного уявлення про інформацію, що обробляється в предметній області, з урахуванням не тільки її структури, але і правил зберігання і обробки, відображених у функціях і поставлених завданнях.

Для розробки методу автоматичного підбору тимчасового житла, а також його програмної реалізації, необхідно розглянути різні аспекти і функціональні вимоги, щоб зрозуміти, як оптимально розробити і вдосконалити такий метод та відповідне йому ПЗ. Ось кілька ключових аспектів аналізу:

1. Користувачі:
 - визначити цільову аудиторію вашого сайту для підбору тимчасового житла – це можуть бути орендарі, покупці, власники, агенти по нерухомості;
 - розглянути їх потреби і очікування щодо підбору тимчасового житла.
2. Функціональність:
 - визначити, які основні функції повинен надавати сайт: пошук, фільтрація, деталізація об'єктів, контакт із продавцем або агентом, перегляд фотографій тощо;
 - розглянути можливість інтеграції з картами, відеооглядами тощо.
3. База даних:
 - розробити базу даних для зберігання інформації про житло, включаючи характеристики об'єктів, ціни, розташування, статус (на продаж, в оренду) тощо;
 - забезпечити систему ведення профілів користувачів та зберігання їх історії підбору.
4. Дизайн і користувацький інтерфейс:

- створити зручний і інтуїтивний інтерфейс для користувачів, щоб вони легко могли знаходити та переглядати об'єкти нерухомості;

- розглянути адаптивний дизайн для забезпечення зручного відображення на різних пристроях.

5. Безпека:

- забезпечити захист від шахраїв та зловмисників, особливо коли мова йде про обмін особистою інформацією та фінансовими операціями.

6. Правові аспекти:

- розглянути закони та регуляції, які регулюють ринок нерухомості та інтернет-посередників;

- переконатися, що сайт відповідає всім правовим вимогам і нормам.

7. Маркетинг:

- розробити стратегію просування і привертання користувачів на сайт;
- розглянути можливість співпраці з агентами по нерухомості та рекламодавцями.

8. Зворотний зв'язок:

- забезпечити можливість користувачам залишати відгуки та оцінки, щоб інші користувачі могли отримати інформацію про досвід інших.

Аналіз предметної області – це лише початок роботи над сайтом для підбору тимчасового житла. Цей процес допоможе вам зрозуміти потреби користувачів та створити ефективний та зручний інструмент для підбору та реклами нерухомості. Однією з перших платформ для підбору тимчасового житла був веб-сайт Craigslist, який був запущений у 1995 році. Craigslist дозволяв користувачам розміщувати оголошення про продаж та оренду нерухомості.

Craigslist — це приватна американська компанія, яка керує веб-сайтом із закритими оголошеннями з розділами, присвяченими вакансіям, житлу, продажу, необхідним речам, послугам, громадським роботам, концертам, резюме та дискусійним форумам.

Започаткував компанію Крейг Ньюмарк в 1995 році як список розсилки електронною поштою друзям із місцевими подіями в районі затоки Сан-

Франциско. У 1996 році він став веб-сервісом і розширився до інших класифікованих категорій. Він почав поширюватися на інші міста США та Канади в 2000 році і зараз охоплює 70 країн.

Сайт обслуговує понад 20 мільярдів переглядів сторінок на місяць, що ставить його на 72 місце серед веб-сайтів у всьому світі та на 11 місце серед веб-сайтів у США, понад 49,4 мільйонів унікальних відвідувачів щомісяця тільки в Сполучених Штатах. Щомісяця з'являється понад 80 мільйонів нових рекламних оголошень, Craigslist є провідною службою оголошень у будь-якому середовищі.

Ще в 2009 році сайт отримував понад 2 мільйони нових списків вакансій щомісяця, що зробило його одним із найкращих дошок вакансій у світі. Станом на жовтень 2011 року 23 найбільші міста США, перелічені на домашній сторінці Craigslist, отримують понад 300 000 повідомлень лише в розділах «для продажу» та «житла». У 2009 році в Craigslist працювало 28 осіб.

Згодом, багато інших онлайн-сервісів та платформ виникли для полегшення підбору тимчасового житла. Список деяких з них:

1. Realtor.com [1]: Realtor.com (рисунок 1.1) був створений у 1995 році та надає інформацію про доступну нерухомість у США. Він допомагає покупцям та орендарам знаходити варіанти житла на основі їх потреб та бюджету.

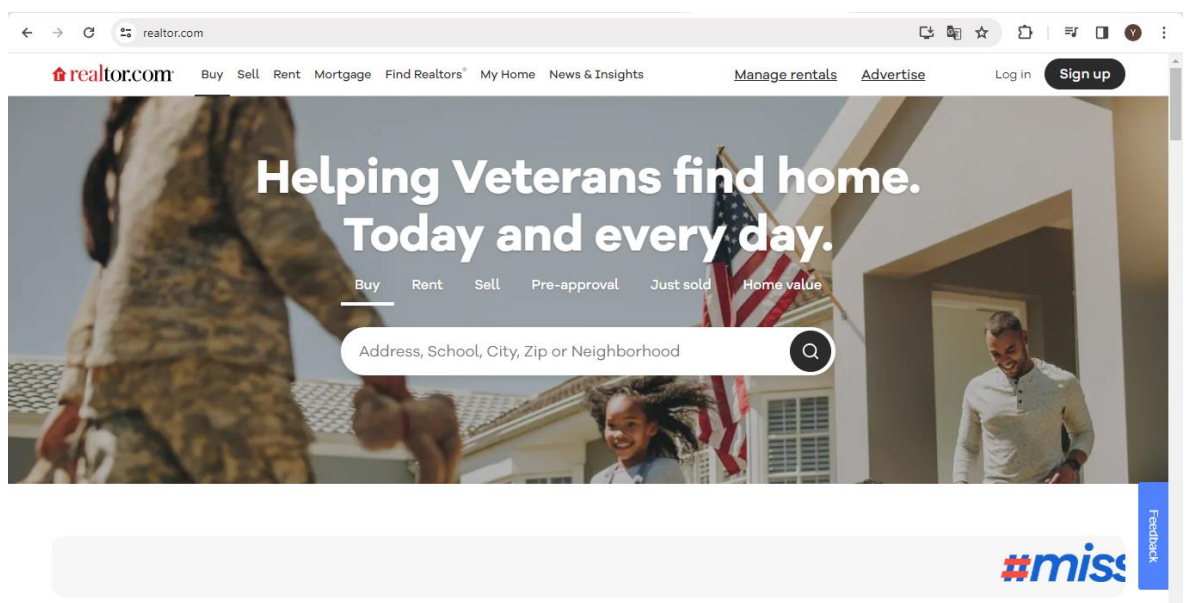


Рисунок 1.1 – Дизайн титульної сторінки сайту Realtor.com [1]

2. Zillow.com [2]: Zillow (рисунок 1.2) створений в 2006 році і є однією з найпопулярніших платформ для підбору тимчасового житла в США. Він надає інформацію про продаж, оренду, та оцінку вартості нерухомості.

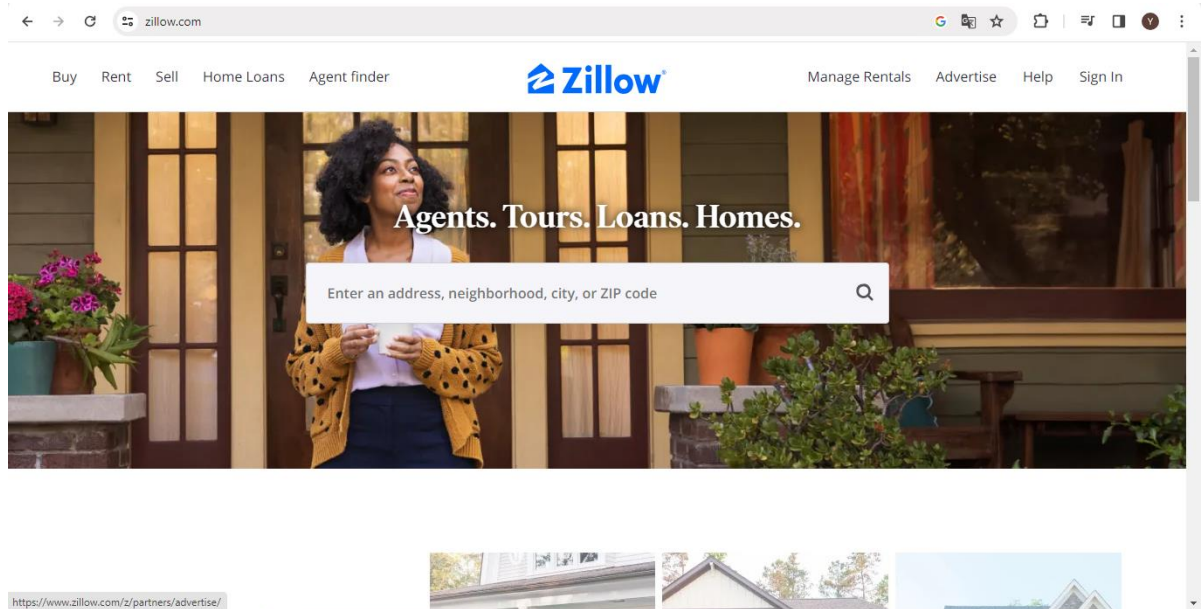


Рисунок 1.2 – Дизайн титульної сторінки сайту Zillow.com [2]

3. Airbnb.com [3]: Airbnb заснований в 2008 році і спеціалізується на короткостроковій оренді житла. Ця платформа дозволяє власникам нерухомості розміщувати своє житло для оренди туристам та мандрівникам.

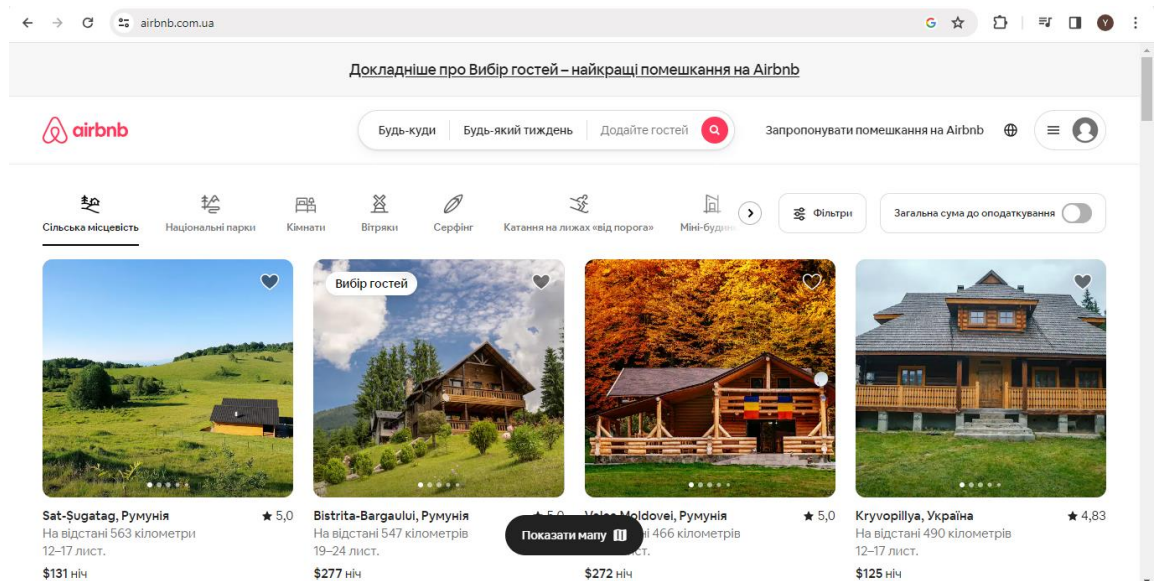


Рисунок 1.3 – Дизайн титульної сторінки сайту Airbnb.com [3]

4. Booking.com [4]: Booking.com (рисунок 1.4) заснований у 1996 році, надає можливість забронювати готелі та апартаменти для короткострокового проживання в різних країнах світу.

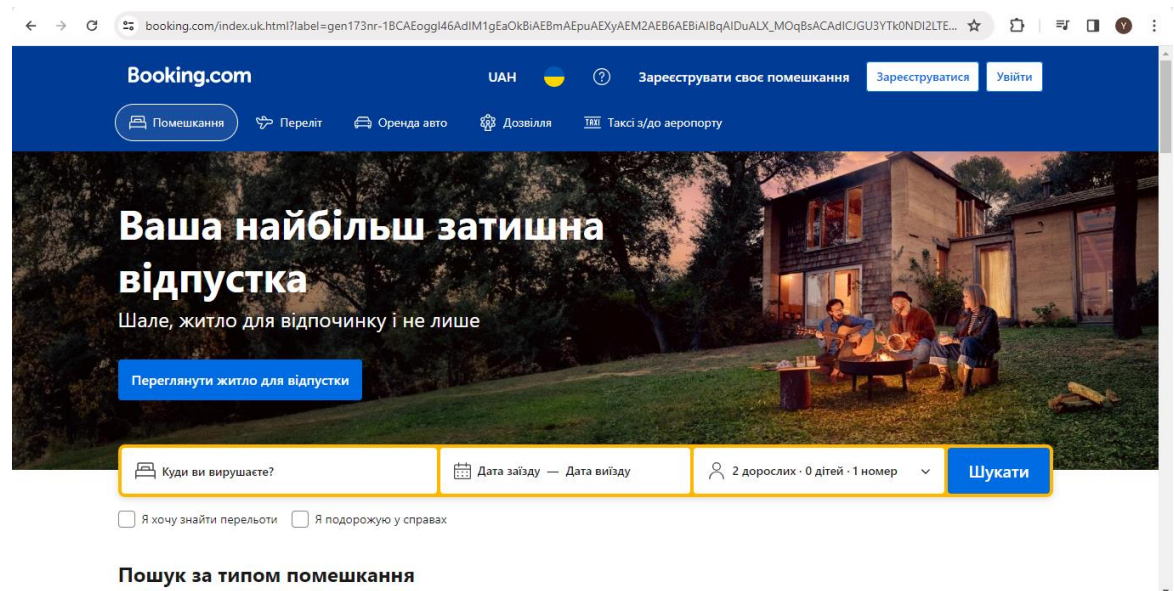


Рисунок 1.4 – Дизайн титульної сторінки сайту Booking.com [4]

Ці платформи революціонізували спосіб, яким люди шукають житло, роблять бронювання та здійснюють оплату. Вони надають більше можливостей для вибору житла та забезпечують доступ до детальної інформації про нерухомість, включаючи фотографії, відгуки користувачів і карту розташування.

Зважаючи на війну в Україні нещодавно був створений сайт для підбору тимчасового житла для біженців у Німеччині. Німеччина запустила нову житлову програму для українських біженців. Там можна дізнатися, як працює онлайн-платформа, що потрібно для реєстрації та подання заявки, а також що нового для українців. Тепер українським біженцям у Німеччині стало простіше знайти житло. Нова онлайн-платформа полегшує для всіх бажаючих вибір житла та подання заявки.

Веб-сайт проекту helfendewaende.de (рисунок 1.5) доступний українською мовою і має прості та зрозумілі функції, які дозволяють легко подати заявку та знайти житло [5].

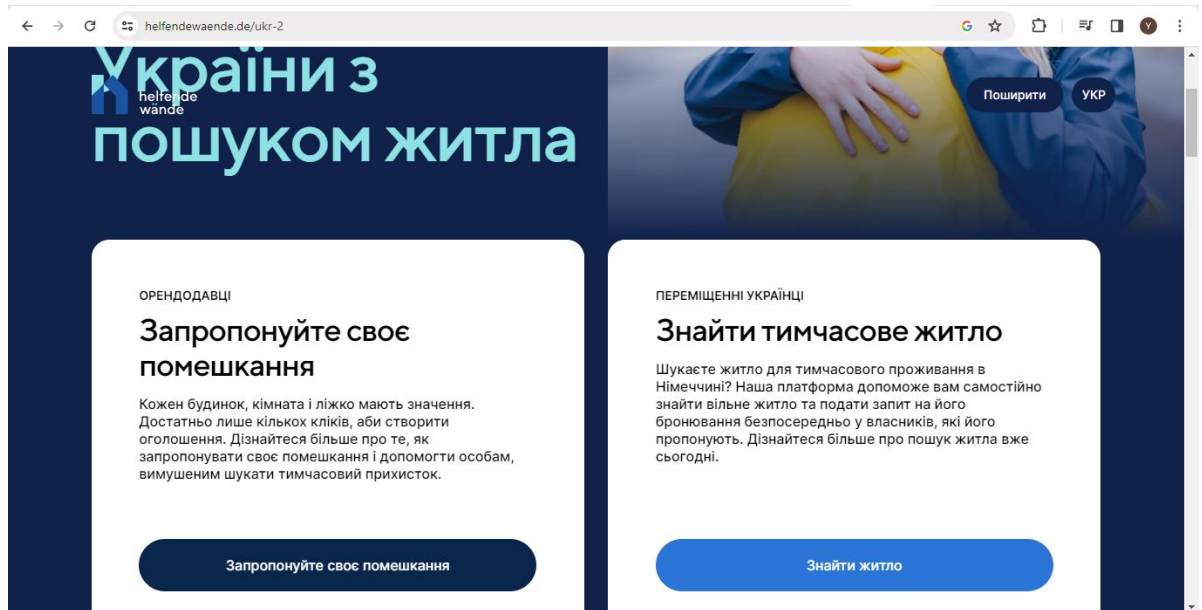


Рисунок 1.5 – Дизайн титульної сторінки сайту helfendewaende.de [5]

Крім того, сайт дозволяє користувачам:

- шукати та бронювати вподобане житло;
- відстежувати статус всіх заявок;
- ділитися своїми контактними даними з орендодавцями та узгоджувати деталі перед тим, як заселитися;

Якщо заявка прийнята, користувач повинен підтвердити заявку, щоб повідомити орендодавця про те, що він зацікавлений у житлі на даний момент. Сервіс також дозволяє шукати як безкоштовне, так і платне житло.

1.2 Аналіз існуючих наукових публікацій про підбір житла на основі генетичного алгоритму

Основним законом України закріплено фундамент щодо особливого статусу переселенців на території України, окрім того, в 2014 Верховною радою України був прийнятий Закон України «Про забезпечення прав і свобод внутрішньо переміщених осіб» відповідно до якого правовий статус переселенців набував значення в функціонування різних сфер суспільства Перш за все, закріплюється право особи на захист від примусового переміщення на

будь-якій території України, сприяння держави щодо надання особам кредитів чи іншої фінансової допомоги для забезпечення житлом [6].

Через прояв дискримінації у великої кількості переселенців виникає проблема щодо оренди житла, через невпевненість орендаря щодо оплати послуг. Крім того, дискримінація впливає на особу також в сфері працевлаштування на новому місці перебування через недостатність документів для влаштування на роботу, чи за інших обставин, що не залежать від неї [7].

На сьогодні, в юридичну термінологію входить нове поняття «шелтер», що тлумачиться як притулок, тобто, місце, де тимчасово проживають особи, які постраждали від війни. В більшості випадків, наявність шелтерів - це діяльність волонтерів, які надають допомогу людям на всій території країни [7].

Повертаючись до гарантій забезпечення житлом ВПО, виникають деякі проблемні питання, чіткої організаційної системи щодо розміщення ВПО в безпечних житлових приміщеннях не було, через що, на сьогодні велика кількість переселенців перебуває в державних помешканнях, що не призначені для проживання. В перші місяці війни громадян, що потребували житла, приймали в школах, студентських гуртожитках, коледжах, університетах, через що на початку вересня ВПО стикнулися з проблемою пошуку нового житла, оскільки студенти та школярі мали навчатися офлайн у закладах освіти. Щодо цієї ситуації дала коментар Віце-прем'єр Ірина Верещук та наголосила на «неможливості примусового виселення даної категорії осіб з приміщень, якщо для них не буде надано альтернативного місця проживання» [8]. Це порушує права ВПО, відповідно до частини 8 статті 11 Закону України «Про забезпечення прав і свобод внутрішньо переміщених осіб». Місцеві державні адміністрації в межах своїх повноважень забезпечують надання у тимчасове користування внутрішньо переміщеним особам житлового приміщення або соціального житла, придатного для проживання» [6][7].

Отже, кожна особа має право на отримання тимчасового житла, хоча б до закінчення військових дій, а виселення є прямим порушенням цього права. Як альтернатива, також можливо виділення чи придбання земельних ділянок в

сільській місцевості [9; 10] для ВПО за кошти приватних спонсорів, державних чи муніципальних організацій [7].

Виникає питання розгляду нових підходів до обліку, визначення категорій придатності житла та дослідження категорій споживачів. Генетичні алгоритми мають широкий потенціал у сфері нерухомості і пропонують багато переваг, особливо з практичної точки зору. Для користувачів ринку нерухомості здатність визначати зміни орендної плати на різних ділянках, що належать до одного сегменту ринку, може бути дуже корисною навіть на невеликих локальних ринках нерухомості. Крім того, ГА дозволяють прогнозувати, вимірювати та визначати, де і як змінюється орендна плата в конкретному міському контексті, а також пов'язувати ці зміни з будь-якими явищами та економічними ефектами (моделювання змінних місцезнаходження, розрізнення районів з однорідними значеннями тощо).

У статті [11] описується система рекомендацій з купівлі житла для користувача та окреслюються проблеми для вирішення (Рисунок 1.6).

Рисунок 1.6 – Знімок екрана системи для купівля житла

Система рекомендацій щодо купівлі житла (далі СРКЖ) з використанням технології генетичного алгоритму допомагає покупцям житла знайти і купити

будинок відповідно до їхніх уподобань і бюджету. Опосередковано це дозволяє скоротити час, витрачений на перегляд будинків через веб-сайт. В даній статті можна розглянути функціональне тестування шляхом оцінки та тестування кожної функції веб-сайту, і було доведено, що СРКЖ працює успішно. Однак, система має ряд обмежень, які потребують вдосконалення. Необхідно вдосконалити такі функції, як відображення щомісячного платежу за будинок, надання можливості покупцям бачити фотографії будинку та відображення відсоткової ставки за банківським кредитом

У статті [12] запропоновано ієрархічний евристичний підхід з використанням ГА для розв'язання задачі розподілу кімнат у гуртожитку. Вплив ваг на обмеження досліджується шляхом моделювання ГА з різними наборами ваг для обмежень розподілу. У процесі моделювання отримано оптимізований розподіл ваг для обмежень на розміщення. Результати, отримані в ході експериментів, показують, що запропонований підхід придатний для розв'язання поставленої задачі. Тут описано експерименти. Початкова популяція ГА була згенерована випадковим чином. Дана публікація була спрямована на науковий експеримент [13]. Алгоритм, показаний тут, спочатку був написаний на Python Себастьяном Сімоном та іншими членами інтегрованої ІТ-команди для розподілу студентів першого курсу Політехнічної школи по студентських квартирах. За кілька тижнів назад науковці розіслали студентам анкету про їхні житлові вподобання. Ця анкета лягла в основу алгоритму. На основі відповідей вони присвоїли числове значення кожній конфігурації квартири і повторювали алгоритм, щоб максимізувати це значення. Результати експерименту були дуже багатообіцяючими. При впровадженні Python вони встановили кілька обмежень, які виключали певні параметри як "неприйнятні", такі як занадто велика різниця у віці, частота або рівень шуму, які відрізнялися більш ніж на три бали. Крім того, було встановлено суворе обмеження, наприклад, не більше половини квартир мають однакову національність і мову, щоб уникнути виключення квартир, в яких проживає лише одна людина. У цьому випадку національна різноманітність була включена як пріоритет при оцінці, щоб створити більш

різноманітні квартири і найкращу можливу композицію з точки зору початкової сумісності. Оцінки квартири показані на Рисунку 1.7 нижче.

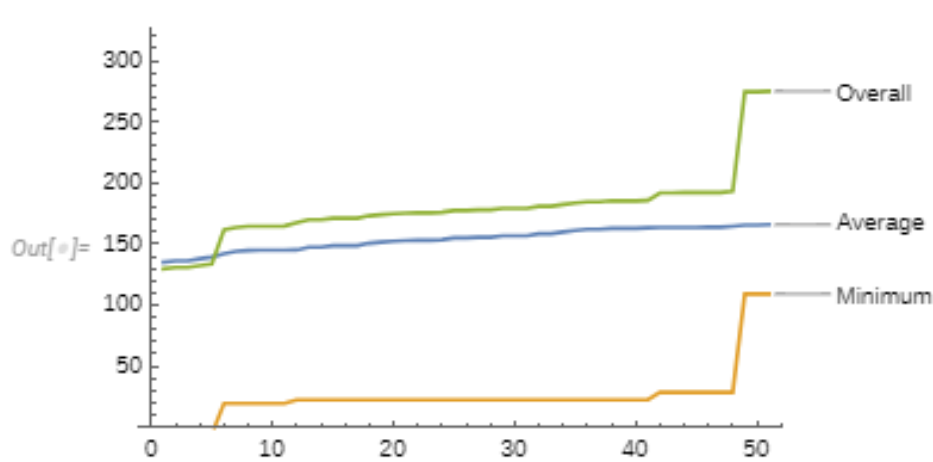


Рисунок 1.7 – Графік відстеження оцінки, мінімальної оцінки квартири та середнє значення за ітерації

Всі ці значення зростають з часом, що свідчить про те, що загальні показники покращуються. Цікавим спостереженням є те, що хоча середнє значення зростає не дуже сильно, мінімальне значення починається з дуже низького значення і підскакує приблизно до 100. Це, ймовірно, говорить нам про те, що на початку було велике стандартне відхилення і що найгірші приміщення значно покращилися протягом ітерацій. Це одна з причин, чому важливо також враховувати мінімальні значення при оцінці.

Для зменшення простору пошуку у цій статті запропоновано алгоритм на основі ГА, який знаходить напівоптимальний розв'язок за прийнятний час обчислень [14]. У цьому дослідженні розглядаються наступні п'ять життєвих звичок та вподобань у процесі розподілу кімнат у гуртожитку.

На відміну від методів кросинговеру та селекції, які зазвичай використовуються в традиційних ГА, в цьому дослідженні представлено модифікований кросинговер (тобто локальний вичерпний кросинговер) та модифіковану селекцію (тобто стратегію селекції 50-50). У методі локального вичерпного кросинговеру для двох хромосом (камер) були перебрані всі можливі комбінації генів для комбінацій генів двох камер, щоб знайти найкращий

можливий результат кросинговеру для двох камер. Таким чином, вдалося уникнути несумісності під час ітерацій ГА і отримати кращі значення пристосованості для цієї задачі розподілу гуртожитків. У стратегії половинного відбору замість того, щоб використовувати весь пул відбору, половина хромосом була відібрана з усього пулу відбору за допомогою методу колеса рулетки. Інша половина хромосом потім піддавалася локальному екстенсивному кросинговеру для отримання нащадків. Це значно скорочує час, витрачений на ітерації ГА, а продуктивність (значення пристосованості) майже така ж, як і при використанні всього пулу відбору.

Запропонований алгоритм в цій статті [15], що використовує генетичні алгоритми, може вирішити цю проблему. Розміщення студентів у відповідних кімнатах з урахуванням їхніх уподобань щодо сусідів по кімнаті є складною проблемою. Запропонований алгоритм реалізовано за допомогою Matlab. Результати запуску алгоритму показують, що він може вирішити проблему ефективно та якісно. Таким чином, він може допомогти працівникам гуртожитку розподіляти місця швидше та якісніше. Алгоритм може бути модифікований для вирішення більш поширених ситуацій (наприклад, деякі студенти не хочуть бути поселеними разом або є два чи більше різних типів кімнат). Спеціальна версія генетичного алгоритму вперше була використана для розміщення студентів у гуртожитках відповідно до їхніх уподобань.

Проаналізувавши літературу та дослідження, можна зробити висновок, що застосування генетичного алгоритму може мати місце в різних аспектах аналізу ринку нерухомості. Генетичні алгоритми є чудовим інструментом, який використовується як компонент більш широких методів, що застосовуються в системах аналізу ринку та оцінки нерухомості. Генетичний алгоритм дозволяє мінімізувати ризик і невизначеність шляхом надання відповідних знань, які певним чином обробляються і надаються особам, що приймають рішення. Рішення приймаються шляхом аналізу інформації, доступної для даної проблеми, і вибору найкращої альтернативи для подальших дій. Надлишок релевантної інформації, тобто так званий інформаційний шум, може завадити

інтерпретації та проведенню аналізу. Традиційні аналітичні методи не відображають реальної картини ринку, тому все частіше використовуються еволюційні методи, які є одним з напрямків штучного інтелекту.

1.3 Аналіз моделей підбору тимчасового житла та їх рішень

Інформаційна модель автоматизованого підбору тимчасового житла та відповідні їм рішення зазвичай включають різні розділи та функції, що дозволяють користувачам легко знаходити житло за їхніми потребами та вимогами. Ось загальний опис:

1. Головна сторінка:
 - пошукова панель з можливістю введення параметрів підбору, таких як локація, тип житла, ціновий діапазон і т. д;
 - рекомендації та акції для заохочення користувачів пошукового сервісу;
 - посилання на найпопулярніші категорії житла або міста.
2. Результати підбору:
 - список пропозицій житла, що відповідають введеним параметрам;
 - фільтри для подальшого обмеження результатів, такі як кількість спалень, тип житла, вартість і т. д;
 - мапа з позначками на місцях розташування житла.
3. Сторінки житла:
 - детальна інформація про кожну пропозицію, включаючи фотографії, опис, ціну, адресу та контактну інформацію;
 - відгуки та рейтинги користувачів, які раніше орендували це житло;
 - посилання на поділитися цією пропозицією в соціальних мережах або надіслати її іншим користувачам.
4. Сторінка користувача:
 - профіль користувача з можливістю додавати свої пропозиції житла (для орендодавців) або зберігати вподобані пропозиції (для орендарів);

- історія пошуків та переглядів;
- опція підписатися на сповіщення про нові пропозиції, що відповідають параметрам користувача.

5. Розділ "Послуги та поради":

- статті, поради та інформація про ринок нерухомості, оренду та купівлю житла;
- можливість отримати консультації від фахівців з нерухомості.

6. Сторінка "Зв'яжіться з нами":

- контактна інформація для клієнтів, які потребують підтримки або мають запитання.

7. Розділ "Про нас":

- інформація про компанію або сервіс, який надає цей сайт.

8. Реєстрація та вхід:

- можливість створення облікового запису для користувачів;
- можливість входу за допомогою соціальних мереж чи сервісів.

Ця структура є загальною та може варіюватися в залежності від конкретних потреб і можливостей системи (сайту) для підбору тимчасового житла. Додатково важливо враховувати підтримку мобільних пристроїв та забезпечення зручного інтерфейсу для користувачів з різними пристроями.

Розглянемо роботу інформаційної системи для підбору тимчасового житла на прикладі сайту для українців «helfendewaende.de». Щоб забезпечити безпеку всіх користувачів платформи, сайт просить як тих, хто пропонує своє житло, так і тих, хто його потребує, підтвердити свою особу. Пропонується ця послуга у співпраці з безпечним партнером Veriff. Для підтвердження особи знадобиться наступне:

- дійсний документ, що посвідчує особу, виданий урядом, з вашою фотографією (вам знадобиться оригінал документа, скани не приймаються);
- пристрій з робочою камерою та підключенням до інтернету;
- хвилина на те, щоб сфотографувати себе та посвідчення особи.

На сайті Wunderflats.com (рисунок 1.8) головна сторінка попередньо запитує до якого числа ви плануєте перебувати в тимчасовому житлі. А вже потім можна налаштувати фільтри, комфортні для себе [16].

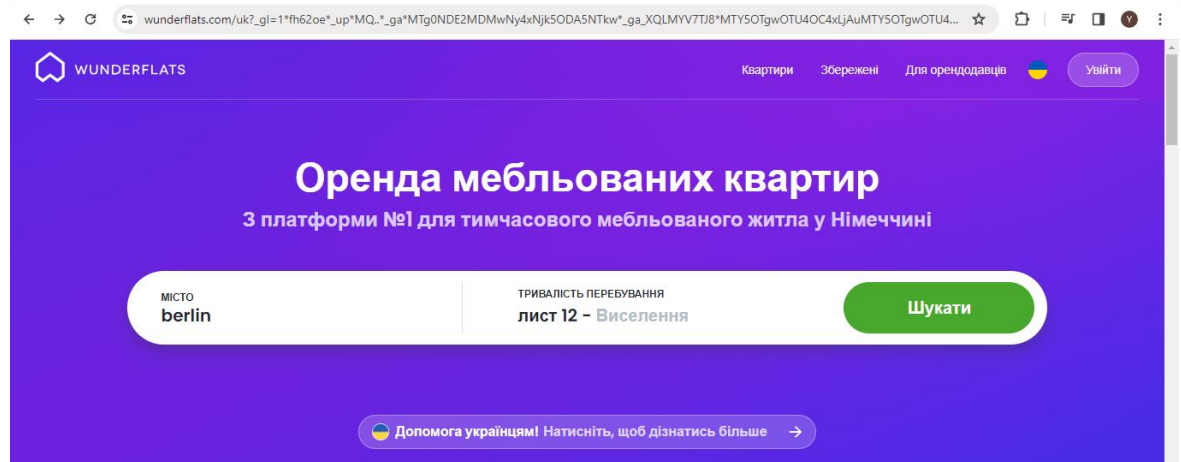


Рисунок 1.8 – Дизайн титульної сторінки сайту Wunderflats.com

Фільтри для користувачів на сайті підбору тимчасового житла допомагають користувачам знаходити нерухомість, яка відповідає їхнім потребам та вимогам. Нижче наведені типові фільтри, які можуть бути використані на сайті:

1. Розташування:
 - місто або регіон;
 - район або округ.
2. Відстань від певних об'єктів (наприклад, відстань до роботи, шкіл, магазинів).
3. Тип нерухомості:
 - квартира;
 - будинок;
 - комерційна нерухомість (офіси, магазини тощо).
4. Ціновий діапазон:
 - мінімальна та максимальна ціна нерухомості.
5. Кількість кімнат.

6. Площа:
 - мінімальна та максимальна площа нерухомості (квадратні метри).
7. Стан:
 - новобудова;
 - вторинний ринок.
8. Тип будівництва:
 - панельний будинок;
 - цегляний будинок;
 - житловий комплекс.
9. Парковка:
 - наявність парковки (гараж або місце для автомобіля).
10. Зовнішня панорама (вид з вікна):
 - вид на море/парк/гори тощо.
11. Зони комфорту:
 - басейн;
 - тераса;
 - камін.
12. Додаткові параметри:
 - можливість дозволити введення інших параметрів, що не входять до основних вимог.

Даний сайт має такі фільтри [16] (див. Рис. 1.9 – Рис. 1.13):

- дата;
- ціна;
- черга (кількість осіб, претендентів на одне житло);
- зручності та додаткові фільтри.

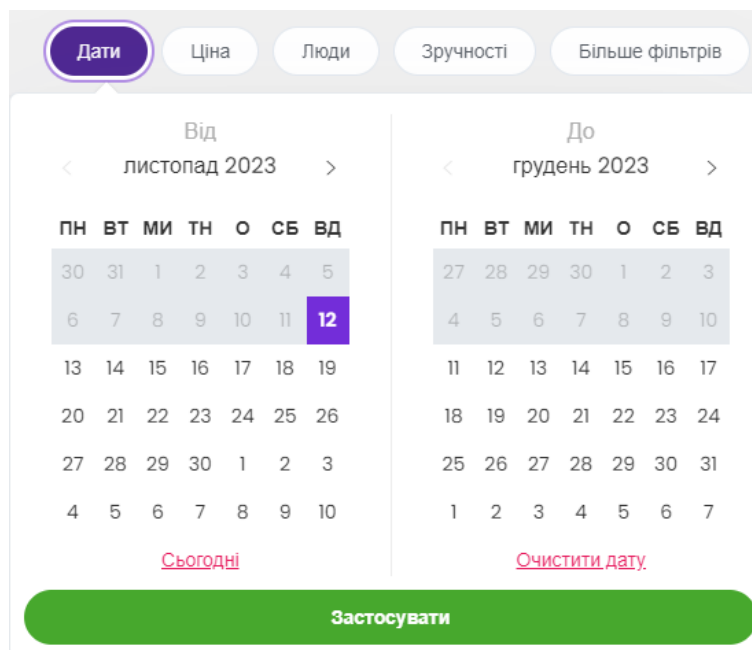


Рисунок 1.9 – Фільтри підбору «за датою» тимчасового житла на сайті для біженців в Німеччині

Користувачі можуть вибрати фільтри за своїми перевагами та відзначити ті параметри, які їм важливі. Це дозволяє отримати більш точні результати підбору тимчасового житла і спрощує процес вибору.

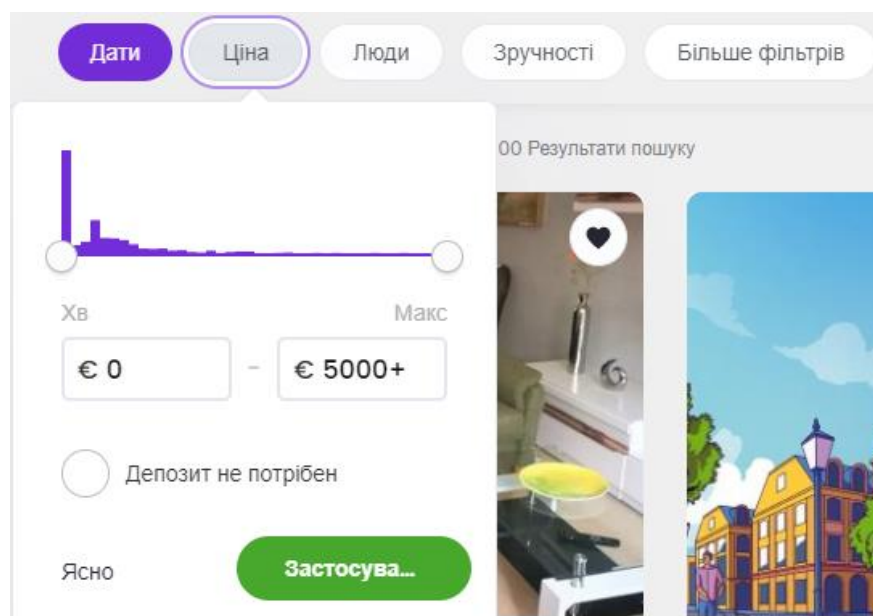


Рисунок 1.10 – Фільтри підбору «за ціною» тимчасового житла на сайті для біженців в Німеччині

Інтерфейс для налаштування фільтрів можна побачити на прикладі сайту Wunderflats.com, де користувачі можуть зручно вибирати параметри та встановлювати їхні значення.

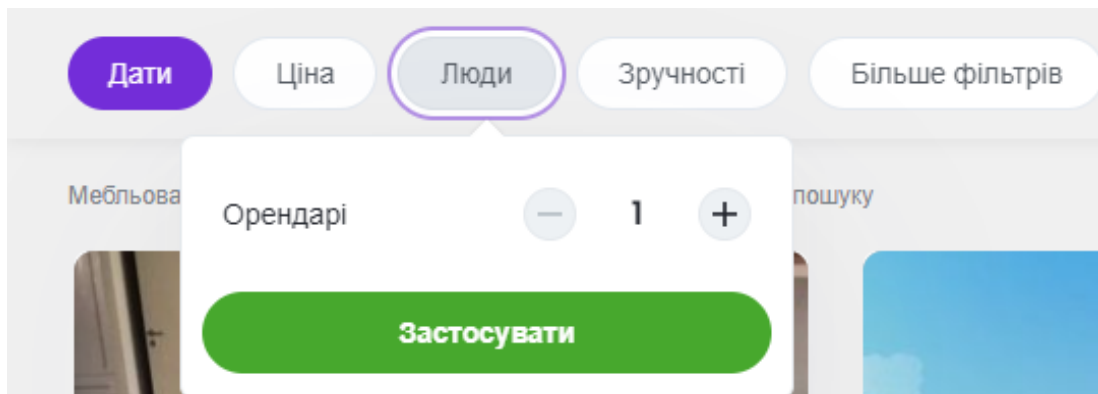


Рисунок 1.11 – Фільтри підбору «за кількістю людей на житло» тимчасового житла на сайті для біженців в Німеччині

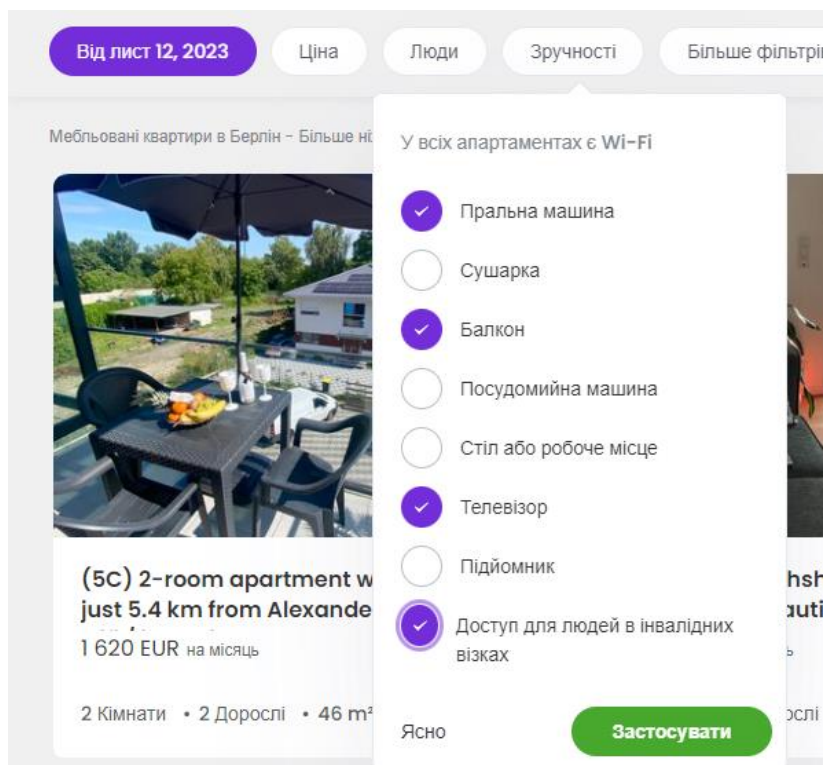


Рисунок 1.12 – Фільтри підбору «за додаткових зручностей» тимчасового житла на сайті для біженців в Німеччині

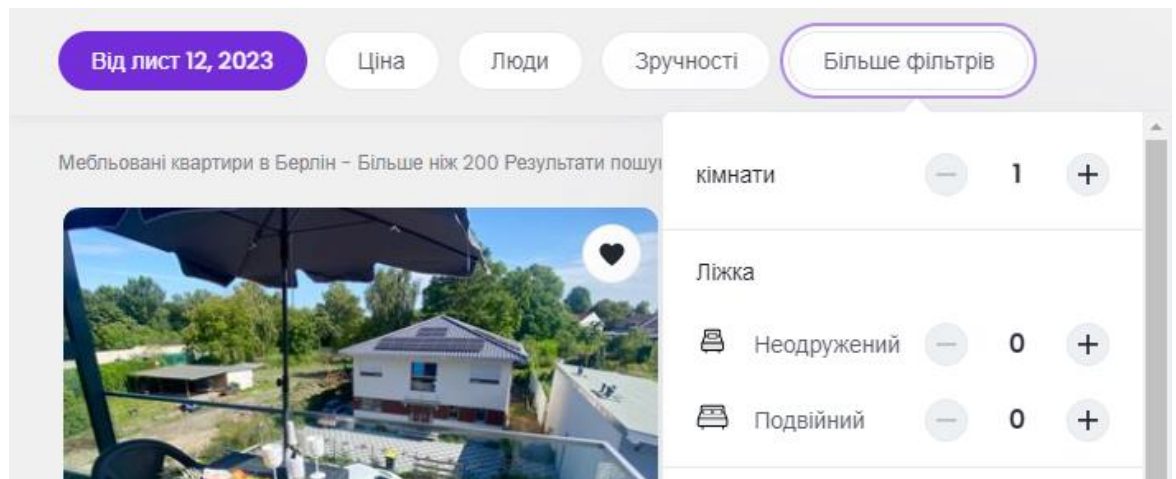


Рисунок 1.13 – Фільтри підбору «за додаткових умов» тимчасового житла на сайті для біженців в Німеччині

Особливістю цього сайту є Календар, який показує дні, коли житло вільне, і дні, коли житло ще зайняте та «нейтральний» день, коли особи можуть виселятися і заселятися.

1.4 Постановка задачі

Метою кваліфікаційної роботи магістра є розробка методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Розрахунок та логіка генетичного алгоритму повинна знаходити оптимальне житло для категорії споживачі з урахуванням їх вимог до житла, рівнів пріоритетності; відповідати очікуванням користувачам з високим рівнем задоволеності їх бажань. ПЗ має бути універсальним або легко налаштовуватись в залежності від способів організації БД. Основне, що ПЗ повинно будуватися на основі використання генетичного алгоритму.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз предметної області та відомих підходів до підбору тимчасового житла для категорій споживачів;
- вдосконалити інформаційну модель підбору тимчасового житла для категорій споживачів;

- розробити метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом;
- підготувати набір даних для застосування у генетичному алгоритмі;
- використати генетичний алгоритм для оптимізованого підбору тимчасового житла для категорій споживачів;
- провести функціональне та прикладне дослідження ефективності розробленого методу та визначити його загальну точність.

Успішне виконання завдання передбачає розробку методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом, досліджуючи житло на ступінь задоволення потреб шукачів та рівню очікування.

Висновки для Розділу 1

У даному розділі проведено аналіз проблеми проектування та знаходження оптимального тимчасового житла для категорій споживачів, а саме: розглянуто основні аспекти, проблеми, фактори при розробці даного методу.

Вдосконалення методів та технологій пошуку тимчасового житла дозволить якісно забезпечувати різні категорії споживачів необхідним помешканням. Описано актуальність завдання по розробці методу автоматизованого підбору тимчасового житла в залежності від затребуваних вимог та особливостей, факторів соціального забезпечення та послуг. Проведено порівняння різних рішень з підбору житла. Окремо розглянули систему для підбору тимчасового житла. Хоча системи підбору житла функціонують вже давно, але від 2022 року у зв'язку з великою міграцією через напад росії на Україну зросла актуальність систем для підбору саме тимчасового житла для різних категорій жителів.

Сформульовано постановку завдання, деталізовано задачі та акцентовано увагу на розробку методу автоматизованого підбору тимчасового житла для категорій споживачів на основі генетичного алгоритму.

Розділ 2. Розробка методу автоматизованого пошуку житла для категорій споживачів за генетичним алгоритмом

2.1 Модель оптимізації пошуку житла

Добре відомо, що еволюційні метафори можуть бути використані для розв'язання задач оптимізації та пошуку, якщо можливі розв'язки задачі певним чином закодовані у вигляді рядків, коректно визначені оператори відбору, мутації та кросинговеру для цих рядків, а також визначені функції, які оцінюють реалістичність цих розв'язків. Оператор кросинговеру допускає значні зміни в закодованому рядку і перебирає множину всіх можливих розв'язків, а оператор мутації поступово "підлаштовує" оптимальний розв'язок.

Коли буде застосовуватися модель оптимізації та пошуку житла (див рис. 2.1), всі розрахунки доводиться переробляти, якщо параметри середовища змінюються навіть незначно. В еволюційних підходах вже сформована популяція може бути проаналізована і адаптована до нових умов середовища, що дозволяє скоротити час виконання алгоритму і застосувати принципи машинної адаптації та навчання. Тобто, кожного разу, коли обрається житло через зміну умов, треба обов'язково змінити фільтр сортування. Кожного разу, коли беруться різні параметри для пошуку, результат буде відмінним від попереднього.

Генетичний алгоритм є стохастичним глобальним алгоритмом оптимізації. У інформатиці та дослідженні операцій генетичний алгоритм (ГА) – це метаевристика, натхненна процесом природного відбору, що належить до більшого класу еволюційних алгоритмів (ЕА). Алгоритм є типом еволюційного алгоритму та виконує процедуру оптимізації, натхненну біологічною теорією еволюції, за допомогою природного відбору з двійковим представленням і простими операторами на основі генетичної рекомбінації та генетичних мутацій.

Алгоритм вигідний тим, що спочатку створюється популяція випадкових адрес можливого проживання фіксованого розміру, які є в базі даних. Основний цикл алгоритму повторюється протягом фіксованої кількості ітерацій або доти,

доки не буде видно подальшого покращення рішення за задану кількість ітерацій.



Рисунок 2.1 – Модель оптимізації пошуку житла

Генетичні алгоритми зазвичай використовуються для створення високоякісних рішень проблем оптимізації та пошуку, покладаючись на такі біологічні оператори, як кросовер і відбір. Деякі приклади додатків ГА включають оптимізацію дерев рішень для кращої продуктивності розв’язування задач оптимізації гіперпараметрів тощо.

Для оптимізації методу автоматизованого підбору житла з урахуванням критеріїв можна побудувати таку формулу:

$$R = w_1 \cdot P + w_2 \cdot L - w_3 \cdot A - w_4 \cdot C \quad (2.1)$$

де:

R – оцінка варіанту житла;

P – показник ціни:

- розраховується як відношення бюджету користувача до ціни оренди варіанту житла;

- чим ближче до 1, тим краще (ціна повністю відповідає бюджету);

- можна використовувати логарифмічну шкалу для більш плавної градації.

L - оцінка локації (чим краще розташування, тим вища оцінка)

- визначається на основі бажаного району, близькості транспорту/об'єктів інфраструктури, екологічності району;

- рекомендую використовувати шкалу від 0 до 1;

- чим ближче до 1, тим краще локація.

A - вік або зношеність житла (чим старіше, тим гірший бал):

- оцінює якість ремонту, зношеність будівлі/квартири;

- також шкала від 0 до 1;

- чим менше значення, тим краще.

C - термін та складність оформлення договору (чим складніше, тим менший бал):

- враховує терміни та умови контракту;

- від 0 до 1, де 1 – мінімальна складність.

$w_1...w_4$ – вагові коефіцієнти для кожного фактору

За цією формулою для кожного варіанту житла розраховується загальний рейтинг R на основі окремих показників та їх ваги.

Варіант з найбільшим рейтингом R буде оптимальним / рекомендованим.

Можливо, це один із найпопулярніших і широко відомих біологічних алгоритмів разом із штучними нейронними мережами.

Розглянемо детальніше етапи цього алгоритму:

1. Створення списку усіх можливих адрес, які є в БД. Стандартний генетичний алгоритм починається зі створення початкової популяції I_0 , яка є скінченною множиною допустимих розв'язків задачі. Ці розв'язки можуть бути

вибрані випадковим чином або отримані за допомогою простого алгоритму наближення. Вибір початкової популяції не пов'язаний зі збіжністю асимптотичного процесу, але створення "хорошої" початкової популяції (наприклад, з набору локальних оптимумів) може значно скоротити час досягнення глобального оптимуму. За відсутності припущень про місцезнаходження глобального оптимуму бажано, щоб особини початкової популяції були рівномірно розподілені по всьому простору розв'язків.

2. Оцінка кожної адреси окремо. Для оптимізації будь-якої структури за допомогою ГА необхідно визначити показник якості для кожної особини в просторі пошуку. Для цього використовується функція пристосованості. У задачах максимізації цільова функція часто виступає в ролі функції пристосованості. Якщо задача, що розв'язується, має алгоритмічно неконтрольовані обмеження, функція пристосованості зазвичай також містить штраф за невиконання обмежень (її значення зменшується).

3. Відбір (відтворення). Існують різні підходи до вибору батьківських пар. Найпоширенішими операторами вибору батьків є наступні.

1. Перемішування (Панміксія) є найпростішим оператором відбору. Відповідно, кожному члену популяції присвоюється випадкове ціле число з діапазону $[1; n]$. Ці числа розглядаються як кількість особин, що беруть участь у схрещуванні. При такому відборі деякі члени популяції спаровуються самі з собою і тому не беруть участі в процесі розмноження. Деякі члени популяції беруть участь у розмноженні з різними особинами популяції більше одного разу. Незважаючи на свою простоту, цей підхід є універсальним для розв'язання різних класів задач. Однак він дуже критичний до розміру популяції, оскільки ефективність алгоритмів, що реалізують цей підхід, зменшується зі збільшенням розміру популяції.

2. Інбридинг – метод, в якому перший батько вибирається випадковим чином, а другим батьком є популяція, найближча до першого батька. Під "найближчою" можна розуміти, наприклад, мінімальну відстань Хеммінга (у випадку двійкових рядків) або евклідову відстань між двома зваженими

векторами. Відстань Хеммінга дорівнює кількості розрізнених локусів (бітів) у двійковому рядку. Інбридинг також використовує поняття індивідуальної схожості. Однак шлюбні пари тепер формуються з найбільш віддалених особин.

Ці два методи по-різному впливають на поведінку генетичних алгоритмів. Так, внутрішній інбридинг характеризується властивістю концентрувати пошук на локальних вузлах, що фактично призводить до поділу популяції на окремі локальні групи навколо ділянок ландшафту, які вважаються надмірно заселеними. Зовнішній інбридинг, з іншого боку, має на меті запобігти збіжності алгоритму на вже знайдених рішеннях і зосередити його увагу на нових, недосліджених ділянках. Інбридинг та аутбридинг можуть бути генотиповими (коли відстань – це різниця між значеннями цільової функції кожної особини) або фенотиповими.

Два найпоширеніші оператори вибору – це оператори ймовірності: пропорційний вибір (рулетка) та турнірний вибір. У деяких випадках також використовується усічений вибір.

Пропорційний вибір. При пропорційному виборі ймовірність того, що на k -му кроці розв'язок i буде обрано як один з батьків, задається формулою $P\{i - \text{вибрано}\} = \frac{f(i)}{\sum_{j \in I_k} f(j)}$, у припущенні, що $f(i) > 0$ для всіх $i \in I_k$ (I_k -популяція на k -му кроці).

Простий пропорційний відбір турніру (рисунок 2.2, відбір за допомогою колеса рулетки) використовує n обертів колеса рулетки для відбору осіб. Колесо рулетки має один сектор для кожного i члена популяції. Розмір сектора i пропорційний відповідному значенню $P(i)$. При такому типі відбору особини з вищою пристосованістю в популяції мають більше шансів бути відібраними, ніж особини з нижчою пристосованістю.

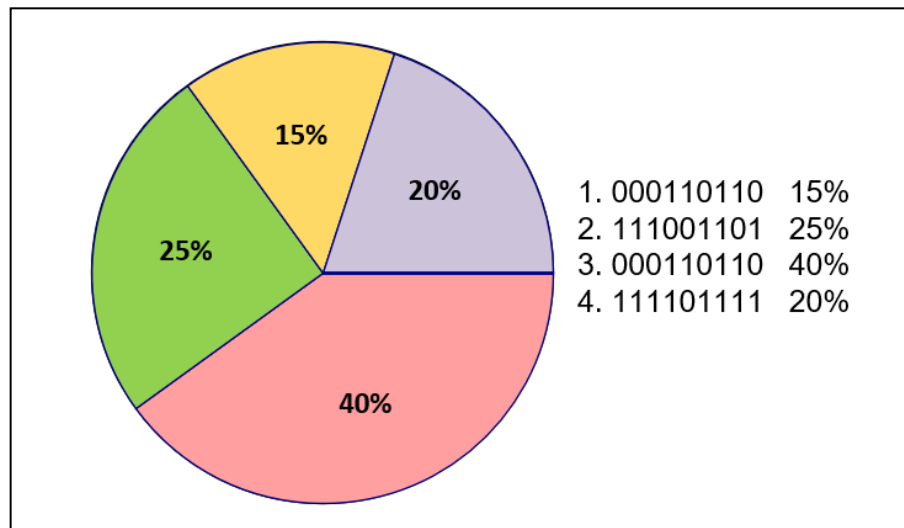


Рисунок 2.2 – Оператор селекції типу колеса рулетки

Вибір турніру. t -рядків випадковим чином вибираються з популяції з m -рядків (індивідів) і найкращий рядок записується на проміжний рядок (між обраними рядками грається турнір). Цей процес повторюється m разів. Рядки отриманого проміжного рядка використовуються для кросоверів. У цьому випадку він називається парним турніром. У загальному випадку t називається розміром турніру (Рисунок 2.3).

Усічений відбір. Ця стратегія використовує популяції, відсортовані в порядку спадання. Кількість особин для спарювання вибирається відповідно до порогового значення $T \in [0; 1]$. Поріг визначає, яка частка перших (найбільш сприятливих) особин буде брати участь у відборі. В принципі, поріг можна встановити рівним 1, в цьому випадку всі особини поточної популяції підлягатимуть відбору. З особин, допущених до спарювання, випадковим чином вибираються $m/2$ пар батьків, а їхні нащадки формують нову популяцію.

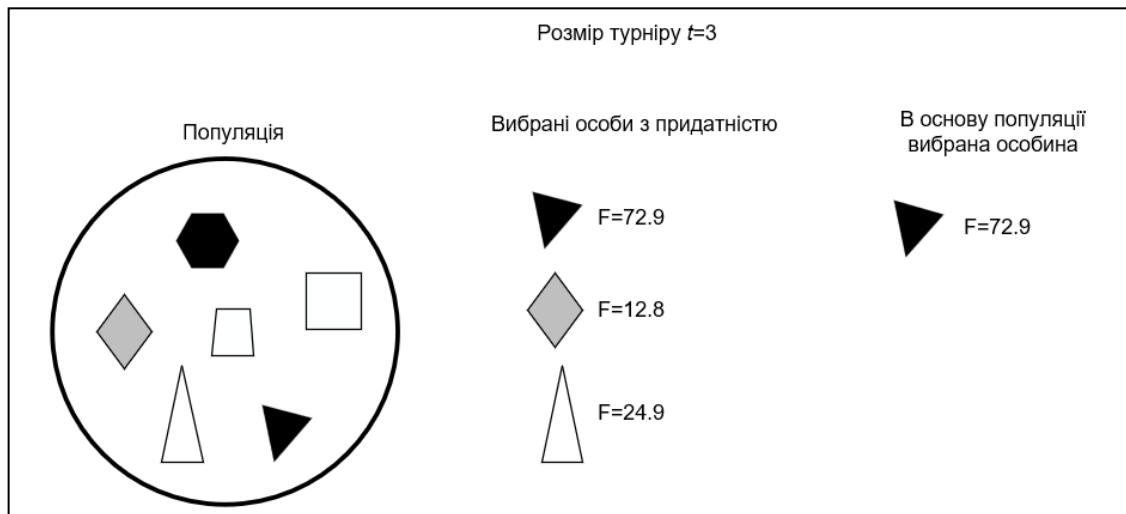


Рисунок 2.3 – Вибір особи методом турніру

Схрещення. Оператор рекомбінації (гібридизації) використовується для створення нових нащадків одразу після оператора селекції батьків. Мета рекомбінації полягає в тому, щоб нащадки успадкували генетичну інформацію своїх батьків. Розрізняють рекомбінацію та схрещування.

Моделі оптимізації є одним із чудових інструментів, які ми, дослідники даних, використовуємо для вирішення проблем: від розв'язання задач оптимізації до пошуку оптимальних гіперпараметрів моделі.

В основному, коли говориться про моделі оптимізації, їх можна класифікувати на основі:

1. Як проводиться пошук:

- Blind Search (сліпий пошук): він полягає у виконанні вичерпного пошуку всіх можливих варіантів без урахування того, який результат ми отримали раніше.

- Guided Search (керований пошук): складається з використання минулих результатів для скерування моделі в пошуку оптимуму. Так само існує два типи моделей керованого пошуку:

- Single State Search (пошук за одним станом): для пошуку результату використовує значення, подібні до першого рішення. У випадках невеликої складності (декілька локальних мінімумів) ця функція збігається (швидко знаходить p.litter).

- Population Based Search (пошук на основі сукупності): сукупність рішень, які вдосконалюються. У такий спосіб, хоча конвергенція займає більше часу, вона працює краще, ніж моделі локального керованого пошуку для найпоширеніших проблем.

2. Чи є пошук стохастичним чи детермінованим. Стохастичний пошук виконується стохастично (випадково), тому двічі запущений процес для тих самих даних може дати різні результати. Однак детерміновані моделі завжди повертатимуть однаковий результат.

2.2 Метод генетичного алгоритму для пошуку житла

Базовий (класичний) генетичний алгоритм складається з наступних кроків (рисунок 2.4) також схема роботи методу зображена на рисунку 2.5:

1. Ініціалізація або відбір початкової популяції хромосом;
2. Оцінка пристосованості хромосом у популяції;
3. Перевірка умов зупинки алгоритму;
4. Хромосомний відбір;
5. Застосування генетичних операторів;
6. Формування нової популяції;
7. Вибір «найкращої» хромосоми.

Ініціалізація, тобто формування початкової популяції, полягає у випадковому виборі заздалегідь визначеної кількості хромосом (особин), представлених бінарною послідовністю певної довжини.

Щоб оцінити пристосованість хромосоми в популяції, для кожної хромосоми в популяції обчислюють фітнес-функцію. Форма функції пристосованості залежить від характеру задачі, яку потрібно вирішити. Передбачається, що фітнес-функція завжди набуває невід'ємних значень і що для подальшого розв'язання оптимізаційної задачі цю функцію необхідно максимізувати. Якщо початкова форма фітнес-функції не задовольняє цим

умовам, виконується відповідне перетворення (наприклад, задача на мінімізацію функції може бути легко перетворена на задачу на максимізацію функції).

Перевірка умов зупинки алгоритму. Визначення умов зупинки генетичного алгоритму залежить від його конкретного застосування. Для задач оптимізації, якщо відоме максимальне (або мінімальне) значення фітнес-функції, алгоритм може зупинитися, коли буде досягнуто очікуваного оптимального значення.

Алгоритм також може бути зупинений, якщо його виконання не призводить до покращення вже досягнутих значень. Алгоритм може бути зупинений після закінчення певного часу виконання або після певної кількості ітерацій. Якщо умова зупинки виконана, алгоритм переходить до завершального етапу, на якому обирається "найкраща" хромосома. В іншому випадку відбувається наступний етап відбору.



Рисунок 2.4 – Стандартна блок-схема генетичного алгоритму

Відбір хромосом – це відбір (на основі значення фітнес-функції, розрахованої на другому етапі) хромосом, які братимуть участь у створенні наступної популяції, тобто наступного покоління нащадків. Цей відбір здійснюється за принципом природного добору, згідно з яким хромосома з

найбільшим значенням фітнес-функції з найбільшою ймовірністю візьме участь у створенні нових особин.

Вхідні дані безпосередньо залежать від споживача. Користувач нашого методу повинен на формі заповнити критерії оцінки житла. Для вибору критеріїв житла пропонуються такі основні параметри житла:

1. Категорія: пристосоване житло, не пристосоване житло;
2. Ціна житла: мінімальна вартість, максимальна вартість;
3. Квадратура: мінімальна величина, максимальна величина;
4. Поверх: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, >10;
5. Відстань від центру: <1км, 1км, 2 км, 3 км, 4 км, 5 км, 6 км, 7 км, 8 км, 9 км, 10 км, >10 км;
6. Тип будинку: цегла, каркасний, силікатна цегла, панель, піноблок, моноліт, інше;
7. Кількість кімнат: кухня-студія, 1, 2, 3, 4, 5;
8. Населений пункт: місто, СМТ, село.

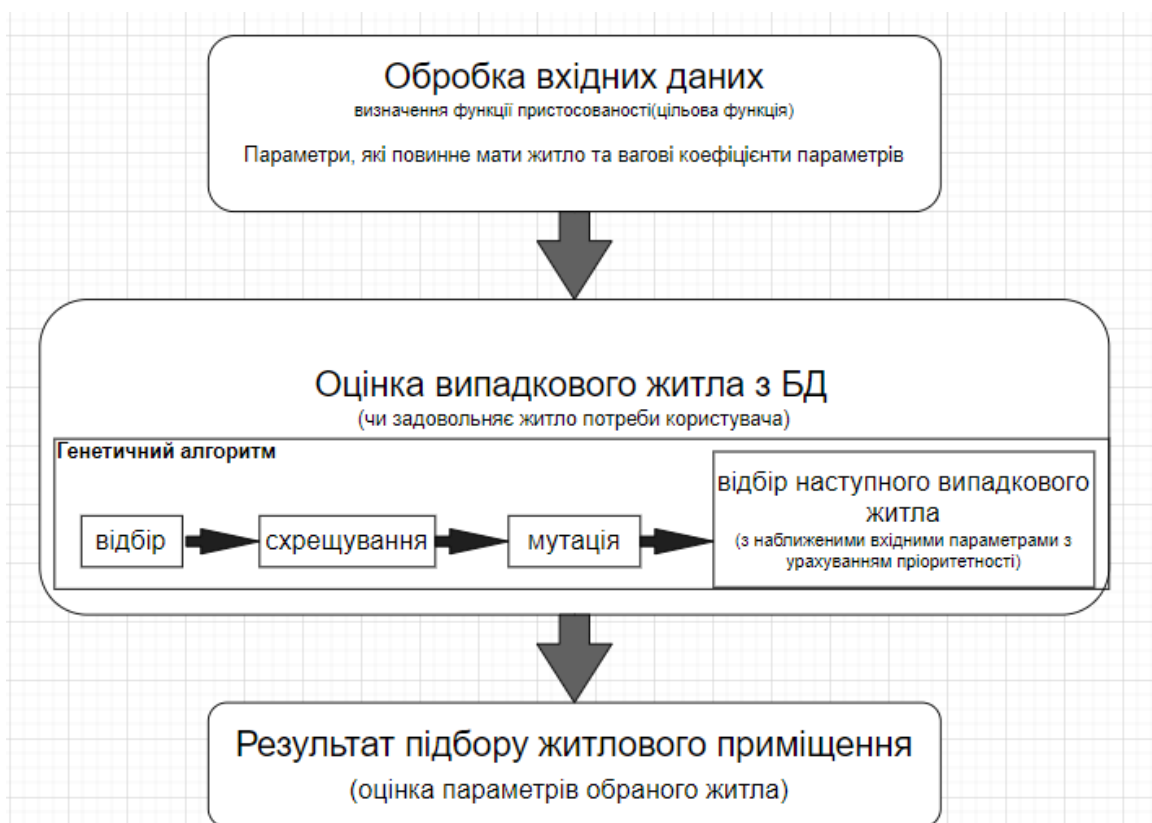


Рисунок 2.5 – Схема методу підбору житла за генетичним алгоритмом

Наступна форма пропонує внести додаткові параметри для детальнішого відбору житла:

1. Відстань від місцевого транспорту;
2. Графік відключення світла;
3. З тваринами;
4. Для людей похилого віку;
5. Для людей з обмеженими можливостями;
6. Опалення;
7. Плита;
8. Балкон;
9. Вид з вікон;
10. Ванна;
11. Меблі;
12. Інтернет;
13. Теплі ковдри;
14. Посуд;
15. Облаштований підвал тощо.

Також наш метод автоматизованого підбору житла має коефіцієнти значимості для вхідних параметрів. Користувач повинен оцінити параметри за своїми пріоритетами, а ПЗ за допомогою нашого методу врахує їх та результатом підбору стане житло, яке буде мати найвищі коефіцієнти. В іншому випадку, якщо такого житла з високими коефіцієнтами в нашій базі даних немає, то наш метод підбору буде враховувати менший коефіцієнт значимості параметра.

Нехай є така функція: $f(x) = 2x^2 + 1$.

Припустимо x – ціле число з інтервалом від 0 до 15. Завдання оптимізації даної функції зводиться до переміщення в просторі, який складається з 16 точок зі значеннями 0, 1, ..., 15 для виявлення точки, де функція приймає максимальне або мінімальне значення.

В нашому випадку змінна x приймає роль параметра задачі. Множина $\{0, 1, \dots, 15\}$ містить в собі простір пошуку та множину потенціальних розв'язків

задачі. Кожне з чисел, які налічуються в множині, називають точкою простору пошуку, розв'язком, фенотипом, значенням параметра. Потрібно зазначити, що розв'язок, який оптимізує функцію, називається оптимальним розв'язком. Кодуємо наступним чином значення параметра $x = 0, 1, \dots, 15$:

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

Цей розповсюджений спосіб двійкового кодування, можна порівняти із записом десяткових цифр у двійковій системі або вони можуть виступати в ролі генотипів. Ще надані кодові послідовності також називають ланцюгами або хромосомами. Хромосоми складають чотири гени (по іншому можна сказати, що двійкові послідовності складаються з чотирьох бітів). Значення гена в конкретній позиції називається алель, яка приймає в даному випадку значення 0 або 1.

Популяція складається з особин, відібраних з цих 16 хромосом. Прикладом популяції розміром у шість особин є, наприклад, набір хромосом $\{0010, 0101, 0111, 1001, 1100, 1110\}$, який кодує такі фенотипи: $\{2, 5, 7, 9, 12, 14\}$. Функція пристосованості у цьому прикладі задається рівнянням (формула 2.2):

$$f(x) = 2x^2 + 1 \quad (2.2)$$

Пристосованість окремих хромосом у популяції визначається значенням x , що відповідає цим хромосомам, тобто значенням цієї функції для фенотипу, що відповідає даному генотипу.

Приклад наступної задачі:

Розглянемо наступний приклад оптимізаційної задачі. У системі (зображеній на Рисунку 2.6) потрібно знайти (формула 2.3):

$$\min_{k_1, k_2} J = \int_0^T f(y_1, y_2, u) dt \quad (2.3)$$

де k_1 та $k_2 \in k_{min}$ та k_{max} .

Параметрами для цієї задачі є k_1 та k_2 . Простір пошуку повинен містити скінченну кількість точок, які можна закодувати як хромосоми. Параметри k_1 та k_2 дискретизовано, і множина їх значень у всьому діапазоні від мінімального значення k_{min} до максимального значення k_{max} відображається у відповідну послідовність двійкових кодів.

Значення k_{min} відповідає кодовій послідовності, що складається лише з нулів, а значення k_{max} відповідає кодовій послідовності, що складається лише з одиниць. Довжина цих кодових послідовностей залежить від значень k_1 і k_2 та частоти дискретизації інтервалу $[k_{min}, k_{max}]$.

Припустимо, що $k_{min} = -25$, а $k_{max} = 25$ і знаковий ряд довжиною 10 для кожного параметра k_1 і k_2 . Приклад популяції з 10 особин наведено в таблиці 2.1. Перші 10 генів кожного генотипу відповідають параметру k_1 , а останні 10 генів відповідають параметру k_2 . Таким чином, довжина хромосоми дорівнює 20.

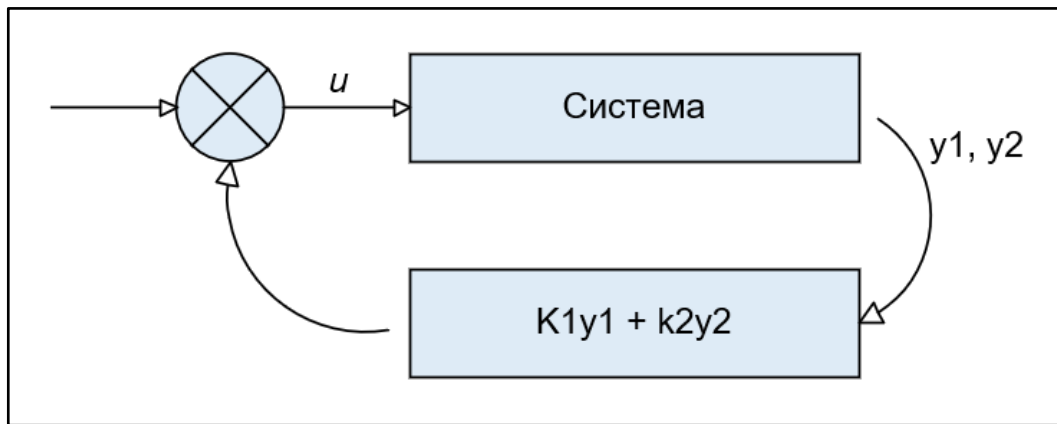


Рисунок 2.6 – Система оптимізаційної задачі

Таблиця 2.1 – Схема оптимізаційної двопараметричної системи

Генотипи	Фенотипи
00000000000000000000	-25,00 -25,00
10100010010011001011	6,72 -15,08

01101000101111010010	-4,57 22,8
11011010011110000111	17,67 19,13
00011011000000010001	-19,72 -24,17
00110000101011111010	-15,52 12,24
11111111111111111111	25,00 25,00

Існують різні методи відбору. Найпопулярнішим є так званий метод рулетки, названий на честь відомої азартної гри. Кожній хромосомі відводиться сектор на рулетці, розмір якого встановлюється пропорційно значенню фітнес-функції відповідної хромосоми. Таким чином, чим вище значення фітнес-функції, тим більший сектор на рулетці.

Все колесо рулетки відповідає сумі значень функції збігу для всіх хромосом цільової популяції. Кожна хромосома позначається через $i = 1, 2, \dots$ і позначається через ch_i для N (де N позначає чисельність популяції) і відповідає сектору колеса $V(ch_j)$, вираженому у відсотках (формула 2.4):

$$V(ch_j) = Ps(ch_j)100\% \quad (2.4)$$

де

$$Ps(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^n F(ch_i)} \quad (2.5)$$

причому, що $F(ch_i)$ – значення функції пристосованості хромосом, $Ps(ch_i)$ – вірогідність селекції хромосоми.

Відбір хромосом можна порівняти з результатом обертання колеса рулетки. Це пояснюється тим, що "виграшна" хромосома (тобто обрана хромосома) відноситься до сектора, який випав з рулетки.

Зрозуміло, що чим більша площа, тим більша ймовірність «виграшу» відповідної хромосоми. Отже, ймовірність вибору даної хромосоми пропорційна

значенню її функції відповідності. Якщо все колесо рулетки представлено у вигляді інтервалу чисел $[0, 100]$, то вибір хромосоми можна порівняти з вибором числа в інтервалі $[a, b]$, де a і b визначають відповідно початкову та кінцеву точки сегмент кола відповідає цій частині колеса; очевидно $0 < a < b < 100$. У цьому випадку вибір використання колеса зводиться до вибору числа в діапазоні $[0, 100]$, яке відповідає певній точці на колі колеса.

Селекція призводить до створення батьківської популяції, яку також називають пулом спарювання, з чисельністю N , що дорівнює чисельності поточної популяції.

Коли генетичний оператор застосовується до хромосом, відібраних селекцією з батьківської популяції, створеної на попередньому кроці, формується нова популяція нащадків.

У класичних генетичних алгоритмах використовуються два основні генетичні оператори: оператор кросинговеру та оператор мутації. Однак слід зазначити, що оператор мутації відіграє явно другорядну роль порівняно з оператором кросинговеру. Тобто в класичних генетичних алгоритмах майже завжди виконується кросинговер, а мутації зустрічаються досить рідко. Ймовірність кросинговеру зазвичай задається дуже високою (зазвичай $0.5 < p_c < 1$), а ймовірність мутації – дуже низькою (частіше за все $0 < p_m < 0.1$). Це впливає з аналогії з біологічним світом, де мутації трапляються вкрай рідко.

У генетичних алгоритмах хромосомні мутації можуть бути виконані на батьківській популяції до кросинговеру або на потомстві, що утворюється після кросинговеру.

На першому етапі спарювання з батьківської популяції (батьківського пулу) відбираються пари хромосом. Це тимчасова популяція, що складається з відібраних в результаті відбору хромосом, які в подальшому змінюються за допомогою операторів спарювання і мутацій для формування нової популяції нащадків. На цьому етапі хромосоми з батьківської популяції спаровуються.

Це робиться випадковим чином залежно від імовірності схрещування P_c . Потім для кожної обраної пари батьків малюється розташування гена (локусу) у

хромосомі, визначаючи так звану точку перехресного обміну. Якщо хромосоми кожного з батьків містять L генів, то очевидно, що точка перетину L_K є натуральним числом, меншим за L . Тому фіксація точки перетину є таким самим, як випадковий вибір числа в інтервалі $[1, L - 1]$. Результатом схрещування хромосом батьків буде наступна пара нащадків:

1. Нащадки, хромосоми яких від позиції 1 до L_K складаються з генів першого батька, а хромосоми від $L_K + 1$ до L , що складаються з генів другого батька.
2. Нащадки, хромосоми яких від позиції 1 до L_K складаються з генів другого батька, а хромосоми від $L_K + 1$ до L , що складаються з генів першого батька.

З імовірністю P_m оператор мутації змінює значення гена в хромосомі на протилежне (тобто з 0 на 1 або навпаки). Наприклад, якщо в хромосомі $[100110101010]$ ген у позиції 7 зазнає мутації, тобто його значення 1 зміниться на 0, що призведе до утворення хромосоми $[100110001010]$. Як згадувалося вище, ймовірність мутації, як правило, дуже низька і залежить від того, мутований ген чи ні. Наприклад, ймовірність мутації P_m може бути змодельована шляхом випадкового вибору числа в інтервалі $[0, 1]$ для кожного гена та вибору для цієї операції тих генів, число яких менше або дорівнює значенню P_m .

Формування нової популяції. Хромосоми, отримані в результаті застосування генетичного оператора до хромосом тимчасової батьківської популяції, включаються до нової популяції. Це буде так звана поточна популяція в поточній ітерації генетичного алгоритму.

На кожній черговій ітерації розраховуються значення функції пристосованості для всіх хромосом цієї популяції, потім перевіряється умова зупинки алгоритму і або зберігається хромосома з найвищою функцією пристосованості, або виконується наступний крок генетичного алгоритму – селекція. У класичному генетичному алгоритмі вся попередня популяція хромосом замінюється новою популяцією з такою ж кількістю нащадків.

Вибір "найкращої" хромосоми. Коли виконуються умови зупинки алгоритму, відображаються результати прогону, тобто шуканий розв'язок задачі. Оптимальний розв'язок – це хромосома з найбільшим значенням функції пристосованості.

Отже, слід визнати, що генетичні алгоритми з часом успадковують характеристики природних еволюційних процесів, що складаються з генетичної мінливості популяцій організмів.

2.3 Інформаційна модель методу пошуку житла

Інформаційна модель пошуку житла для користувачів використовує метод автоматизованого підбору житла за конкретними параметрами, які вводяться користувачем.

Важливо приділити увагу послідовності виконання методу. Дивлячись на Рисунок 2.7 можемо сказати, що першим кроком ми повинні налаштувати генетичний алгоритм під наш метод автоматизованого підбору житла. За допомогою алгоритму відбувається формування множини адрес разом з їхніми параметрами. Для підбору житла по індивідуальних параметрах користувача розглядається кожна підмножина, тобто житло зі списку всіх наявних. Ведеться перевірка усіх наявних параметрів і встановлюється пристосованість житла. Якщо житло не попадає під список параметрів, то воно відсіюється з множини. Цей процес відбору відбувається до тих пір, поки не залишається одне житло. Воно і буде найкращим вибором для користувача.

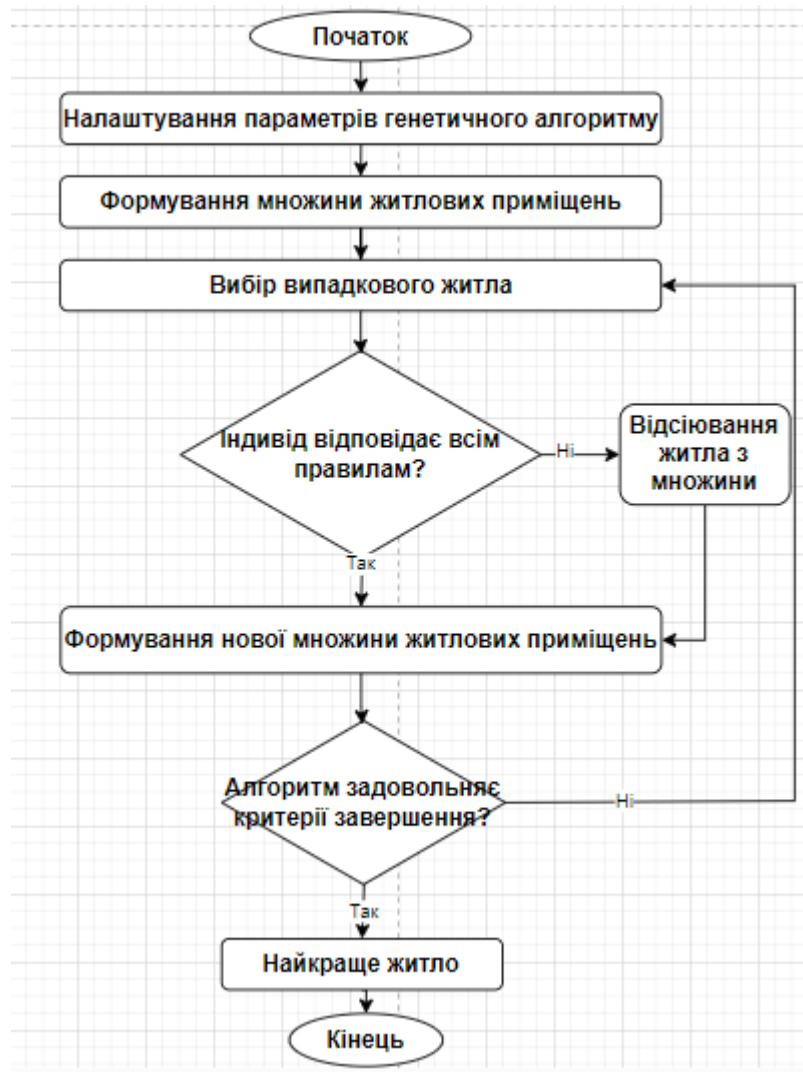


Рисунок 2.7 – Блок-схема підбору житла

Виходом програмного забезпечення буде адреса підбраного житла, місце розташування на карті та номер телефону власника. Такі параметри житла на виході має кожне житло, щоб користувач мав змогу подзвонити до власника і переглянути на карті місце розташування його майбутнього поселення.

Порівнюючи інші програми для пошуку житла з нашим програмним забезпеченням для підбору житла, можемо помітити одну відмінність між методами пошуку. База даних в інших програмах набагато менша від нашої бази даних. Наша база даних налічує в собі всі можливі варіанти житла, від готелю до сараю, від найдорожчих до безкоштовних. Особливістю ПЗ є можливість користувачу зазначити важливість певного критерію. Напроти кожного параметра користувач повинен ввести рівень значимості даного параметра.

Коефіцієнт пріоритетності – це безрозмірна відносна величина, що використовується для встановлення пріоритетності (важливості) певного об'єкта, процесу чи явища відносно інших об'єктів розгляду. Іншими словами, коефіцієнт пріоритетності показує, наскільки даний об'єкт є більш пріоритетним (важливим), ніж інші. Чим вище значення коефіцієнта – тим вищий пріоритет. Коефіцієнти пріоритетності можуть розраховуватися та використовуватися, наприклад:

1. При плануванні та розподілі ресурсів між різними проектами чи напрямками.
2. В теорії масового обслуговування для визначення пріоритетів обслуговування.
3. Для прийняття рішень щодо включення певних заходів у план за наявності ресурсних обмежень.
4. В оптимізаційних задачах з критерієм оптимальності у вигляді суми пріоритезованих показників.

Коефіцієнт пріоритетності – це відносний показник, який дозволяє кількісно оцінити рівень пріоритету певного процесу, явища чи об'єкта в конкретній задачі пріоритезації.

Якщо шукати житло на іншому сайті та нашим програмним забезпеченням враховуючи однакові критерії пошуку, то дві програми на виході покажуть різні результати.

Можна зауважити, що інша програма покаже кілька варіантів житла і користувачу прийдеться далі продовжити обирати і порівнювати ці варіанти. Наше ПЗ для підбору житла для категорій споживачів виведе лише один варіант, це є особливістю методу підбору. Наша база даних більша ніж звичайна, тому задоволеність клієнта від нашого підбору житла буде більшою. Ми встановили в ПЗ гнучкіший лист опитування і користувач зможе зазначити в програмі більше критеріїв. ПЗ методом підбору за допомогою ГА сама підбере житло, враховуючи коефіцієнт значимості критерію.

Висновки до Розділу 2

В даному розділі розроблено метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Описано особливості застосування генетичного алгоритму для підбору тимчасового житла для категорій споживачів. До особливостей нашого методу підбору житла можна віднести: великий список адрес для проживання в різних умовах; гнучкий опитувальник критеріїв житла; коефіцієнт значимості для кожного критерію; пропонується один варіант житла.

Створено схему методу для підбору тимчасового житла. Описано фактори, які впливають на ефективність підбору житла.

Опрацьовано метод на основі генетичного алгоритму, який базується на використанні біологічних залежностей та ітерації для знаходження найкращого рішення для підбору тимчасового житла.

Створено інформаційну модель автоматизованого підбору тимчасового житла, описано зв'язки між елементами методу в процесі опрацювання даних, подано фактори та показники, які використовуються для визначення найефективнішого способу підбору тимчасового житла: формується набір можливих варіантів житла.; кожен варіант оцінюється за певними показниками і визначається один оптимальний варіант.

Розділ 3. Програмна реалізація методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом

3.1 Вимоги до розробки програмного забезпечення

Для того, щоб наша програмне забезпечення могло працювати і виконувати свої завдання, а саме шукати помешкання за потребами користувачів чи автоматизовано підбирати житло, побудуємо її на основі генетичного алгоритму.

Одним з основних етапів створення методу автоматизованого підбору тимчасового житла за категоріями споживачів є проектування та розробка програмного забезпечення. При цьому важливо враховувати особливості генетичних алгоритмів, описані в попередньому розділі. Враховуючи ці особливості, ми обрали компонентно-орієнтовану архітектуру та мову програмування Python, яка надає як можливості об'єктно-орієнтованого програмування для створення незалежних, слабо пов'язаних між собою компонентів, так і можливості функціонального програмування для визначення математичних операцій.

У даному розділі проаналізовано основні функціональні та нефункціональні вимоги до програмного забезпечення методом автоматизованого підбору житла, його архітектуру та спосіб побудови.

Відповідно до структури генетичного алгоритму, описаної в попередньому розділі, доступні наступні компоненти (блоки):

- компонент генерації хромосом;
- компонент створення стартової популяції;
- компонент підбору пари;
- компонент відбору популяції;
- компонент кросинговеру;
- компонент мутацій;
- компонент, що визначає момент завершення роботи.

Основна функціональність системи побудована навколо компонента, який інтелектуально визначає правила для підбору, що нас цікавить, і створює початкову популяцію (підмножину житла).

Функціональні вимоги до програмного забезпечення:

- можливість задавати інтелектуальні правила формування для ранніх поколінь;
- функція генерації хромосом генетичного алгоритму;
- функція вибору методів селекції популяції;
- функція налаштування методів мутації;
- функція налаштування методів кросинговеру.

Нефункціональні вимоги до програмного забезпечення.

- швидкість роботи алгоритму;
- точність роботи;
- якість програмного забезпечення.

Для того, щоб гарантувати використання мови програмування Python для виконання коду, можна скористатися їх об'єктно-орієнтованими можливостями для опису інтерфейсів та компонентів додатку. Також слід враховувати функціональні можливості для створення математичної моделі, що дозволить забезпечити як функціональні, так і нефункціональні вимоги програмного продукту.

Надаючи визначення компонентного підходу до проектування системи і розкриваючи ключові компоненти генетичного алгоритму, ми можемо створити архітектуру системи, використовуючи ці компоненти. Окрім вже згаданих елементів, необхідно також детально описати компонент генетичного алгоритму, який буде визначати основний алгоритм функціонування системи і координувати взаємодію інших компонентів.

Отже, у центрі методу автоматизованого підбору тимчасового житла для категорій споживачів впроваджується компонент виконання генетичного алгоритму. Цей компонент залежить від реалізації елементів створення початкової популяції, селекції, кросовера та мутації, необхідних для повного

функціонування генетичного алгоритму. З метою підбору та автоматичного підбору тимчасового житла реалізовано метод, який відповідає визначеному контракту поведінки. Описавши компоненти генетичного алгоритму таким чином, можна застосовувати рішення для широкого спектру завдань. Навіть у випадку відсутності конкретної реалізації певного елементу генетичного алгоритму, можна легко визначити власний, відповідно до його контракту.

Згідно з описаними базовими компонентами генетичного алгоритму, загалом, необхідно розробити компоненти програмного забезпечення, що відображатимуть їхню функціональність. Визначимо компоненти програмного забезпечення генетичного алгоритму та побудуємо діаграму системи (рис. 3.1.)

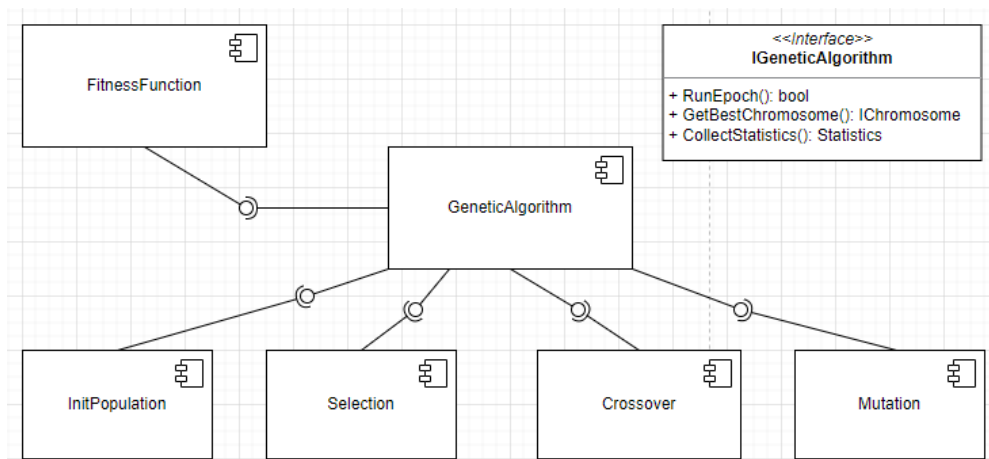


Рисунок 3.1 – Опис компонентів системи

Відповідно до раніше сформованої діаграми компонентів, розглянемо основні інтерфейси генетичного алгоритму. Таким чином, ключовими інтерфейсами системи є:

- IGeneticAlgorithm: Цей інтерфейс відповідає за реалізацію загальної роботи алгоритму. У даному випадку, реалізація компоненту залишається незмінною та залежить від інших компонентів.
- IFitnessFunction: Цей інтерфейс стосується реалізації функції відповідності.

- IPopulationInitializer: Цей інтерфейс відповідає за реалізацію методу генерації підбору тимчасового житла. У межах дослідження розглядаються дві реалізації – випадкова ініціалізація та інтелектуальна.
- ISelectionMethod: Цей інтерфейс визначає метод вибору підходящих для наступного покоління індивідів популяції.
- ICrossover: Цей інтерфейс описує метод кросинговеру.
- IMutation: Цей інтерфейс визначає метод мутації.

Повну діаграму класів можна знайти у додатку А. Система також забезпечить можливість налаштування генетичного алгоритму та методу формування підбору тимчасового житла. Далі описано Use-Case діаграму користувача системи (рис. 3.2).

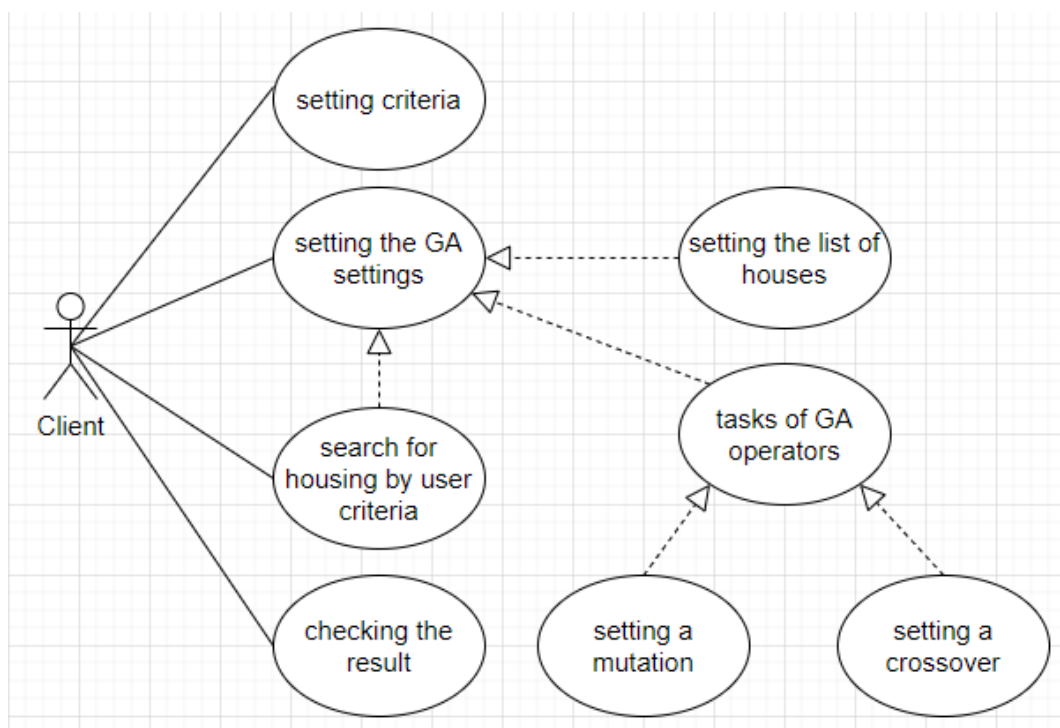


Рисунок 3.2 – UML діаграма використання методу

Ідею використання чітко визначених компонентів для представлення генетичного алгоритму можна успішно розширити до інструмента візуального програмування. У цьому випадку кожному компоненту відповідає користувацький інтерфейс (UI), з яким користувач може взаємодіяти. Такий підхід відкриває можливості налаштування генетичного алгоритму для

розв'язання конкретної задачі не лише за допомогою програмного коду та імплементації інтерфейсів, але і через графічний інтерфейс.

Імплементуючи такі компоненти як незалежні модулі, можна досягти їхнього незалежного розгортання. Це у свою чергу сприяє більш швидкому розробленню та внесенню змін до генетичних алгоритмів.

Важливим етапом у процесі розробки програмного забезпечення є її моделювання. Ми формуємо об'єкти та класи та створюємо зв'язки між ними, щоб система коректно виконувала наше завдання.

Моделювання програмного забезпечення — це важливий етап у процесі її розробки, який дозволяє аналізувати, проектувати та вдосконалювати систему перед її фактичним створенням. Для успішного моделювання програмного забезпечення важливо визначити та врахувати різні вимоги. Ось деякі з основних вимог для моделювання програмного забезпечення:

- Вимоги користувачів: збір та аналіз вимог, пред'явлених користувачами до системи; потрібно дати змогу користувачам підібрати фільтри для себе; визначення функціональних та нефункціональних вимог до системи.
- Аналіз поточних бізнес-процесів: вивчення та аналіз бізнес-процесів та вимог до їх оптимізації; визначення ключових етапів та взаємодій між складовими системи.
- Специфікація властивостей системи: визначення основних характеристик системи, таких як продуктивність, надійність, безпека та інші.
- Моделювання даних: створення моделі даних, яка включає в себе структуру та типи даних, зв'язки між ними та способи їх збереження.
- Архітектурне моделювання: розроблення архітектури системи, включаючи компоненти, їх взаємодію, розташування та інші аспекти.
- Моделювання процесів: визначення послідовності операцій та процесів в системі.
- Технічне моделювання: вибір технологій та інструментів, які будуть використовуватися для реалізації системи.

- Вимоги до безпеки та захисту даних: визначення вимог до захисту інформації та забезпечення безпеки системи.
- Тестування та верифікація: визначення вимог до тестування системи та перевірка відповідності їм під час розробки.

Ці вимоги допомагають забезпечити повноту, чіткість та стабільність моделі програмного забезпечення, що є важливим для успішного розробки та впровадження системи. Об'єктно-орієнтована модель програмного забезпечення – це підхід до проектування та розробки програмного забезпечення, який базується на концепції об'єктів. У цьому підході програмне забезпечення розглядається як набір взаємодіючих об'єктів, які мають свої властивості та поведінку.

Основні концепції об'єктно-орієнтованого програмування (ООП), які використовуються в ООМІС, включають:

1. Клас визначає структуру об'єкта, включаючи його властивості (змінні члені класу) та методи (функції члені класу). Об'єкт є конкретним екземпляром класу.
2. Інкапсуляція дозволяє об'єднувати дані та методи, які їх обробляють, у єдиному об'єкті. Це приховує деталі реалізації від зовнішнього середовища та сприяє зменшенню залежностей між різними частинами системи.
3. Наслідування дозволяє створювати нові класи, які успадковують властивості та методи вже існуючих класів. Це сприяє повторному використанню коду та покращенню організації коду.
4. Поліморфізм дозволяє використовувати об'єкти різних класів через спільний інтерфейс. Це сприяє гнучкості та розширюваності системи.

Використання об'єктно-орієнтованої моделі для методу програмного забезпечення сприяє полегшенню розробки, розумінню коду, підтримці та розширенню системи. ООМІС часто використовується в багатьох галузях програмування, включаючи розробку веб-додатків, баз даних, ігор та інших типів ПЗ.

Метод автоматизованого підбору тимчасового житла за допомогою генетичного алгоритму може бути корисним інструментом для оптимізації процесу пошуку житла для різних категорій споживачів. Генетичні алгоритми використовують природний відбір та еволюційні принципи для пошуку оптимальних рішень у просторі можливих варіантів. Щоб ефективно знаходити оптимальні рішення, алгоритм імітує еволюційний процес популяції, генеруючи набір кандидатів рішень для задачі оптимізації, дозволяючи рішенням відтворюватися та створювати нові рішення (відтворення), а також сприяючи конкуренції, щоб отримати найбільш підходящі рішення, тобто, оптимальний (найкращий) шанс вижити та заселити наступне покоління (природний відбір). Цей процес дозволяє ГА використовувати хороші рішення з часом для створення кращих рішень, і було показано, що він зближується до плато оптимізації.

Проблема ідентифікації оптимальної підмножини ознак також є складною проблемою оптимізації. Отже, якщо проблему вибору ознак можна сформулювати в термінах концепцій еволюційної біології, то механізм ГА можна застосувати для пошуку оптимальних підмножин ознак. У біології кожен суб'єкт у популяції представлений хромосомою, яка побудована з послідовності генів. Кожна хромосома оцінюється на основі критерію придатності. Чим краща відповідність, тим більша ймовірність того, що хромосому буде обрано для відтворення для створення наступного покоління хромосом. Етап відтворення включає процес кросинговеру та мутації, коли дві хромосоми обмінюються підмножиною генів. Невелика частка генів нащадків потім змінюється внаслідок процесу мутації.

Визначення цільової функції: Спочатку потрібно визначити цільову функцію або критерії, за якими буде відбуватися оцінка різних варіантів тимчасового житла. Це може включати в себе такі фактори, як вартість, розташування, розмір, безпеку тощо.

Представлення рішень у вигляді геномів: Кожен потенційний варіант тимчасового житла може бути представлений у вигляді геному або хромосоми.

Наприклад, кожен ген може відображати певну характеристику, таку як вартість оренди, кількість спалень, відстань до роботи тощо.

Популяція: Створюється випадкова початкова популяція різних варіантів тимчасового житла.

Селекція: Визначається метрика адаптивності (цільова функція), і на її основі відбувається відбір найкращих варіантів у популяції.

Схрещування (кросовер) та мутація: Найкращі варіанти обираються для створення нових хромосом шляхом схрещування чи мутації. Це допомагає генетичному алгоритму ефективно досліджувати простір можливих рішень.

Оцінка адаптивності і вибірка: Новостворені хромосоми оцінюються за цільовою функцією, і найкращі з них обираються для наступної ітерації.

Зупинка критерію: Процес продовжується до виконання критерію зупинки, такого як досягнення якості рішення або обмеження кількості ітерацій.

Вибір найкращого рішення: Після закінчення алгоритму вибирається найкращий знайдений варіант тимчасового житла.

Генетичні алгоритми можуть бути дуже потужними для оптимізації складних проблем, таких як вибір тимчасового житла, оскільки вони можуть враховувати багато параметрів і знаходити оптимальні рішення в складних просторах. Для використання цього методу нам, ймовірно, знадобиться програмне забезпечення, що реалізує генетичні алгоритми, а також система оцінки житла та інформація про потреби споживачів.

Слід використовувати зовнішню процедуру повторної дискретизації, щоб визначити, скільки ітерацій пошуку є відповідними. Це усереднює прогнози набору оцінок для всіх повторних вибірок, щоб визначити, як довго має тривати пошук, якщо його безпосередньо застосувати до всього заданого набору. По суті, кількість ітерацій знову є параметром налаштування.

Генетичний алгоритм поступово покращує ефективність прогнозування шляхом змін до підмножин функцій. Подібно до симуляції відбору, для вибору оптимальної кількості поколінь слід використовувати процедуру зовнішньої повторної вибірки. Зокрема, ГА виконуються в кожному зовнішньому циклі

передискретизації. Потім внутрішні повторні вибірки використовуються для вимірювання придатності, щоб різні дані використовувалися для моделювання та оцінки. Цей процес допомагає запобігти знаходженню генетичним алгоритмом підмножини функцій, яка переповнює доступні дані, що є законним ризиком для цього типу інструментів оптимізації.

Після встановлення кількості поколінь завершальний процес відбору виконується на навчальному наборі, а найкраща підмножина використовується для фільтрації предикторів для остаточної моделі.

На рисунку 3.3 показані криві, які відстежують розмір підмножини та прогнозні характеристики поточної найкращої підмножини кожного покоління для кожної зовнішньої повторної вибірки [17]. Крім того, цей рисунок ілюструє середню подібність підмножин у межах покоління до поточної найкращої підмножини для кожної зовнішньої повторної вибірки. З цієї фігури виділяється кілька характеристик. По-перше, незважаючи на значні варіації повторних вибірок у розмірі першого покоління, усі вони збігаються до розмірів підмножини, що містить від 61 до 85 предикторів. Діаграма подібності вказує на те, що підмножини в межах покоління стають більш схожими на найкраще рішення. У той же час площа під кривими ROC починає виходити на плато до середнього значення 0,843.

ROC-крива – графік, який дає оцінку якості бінарної класифікації і показує співвідношення між загальною кількістю об'єктів з ознакою, які правильно класифіковані, і загальною кількістю об'єктів, які помилково класифіковані як такі, що мають цю ознаку і не мають її. Також називається кривою помилок; аналіз класифікації за допомогою ROC-кривих називається ROC-аналізом.

Через загальну тенденцію до збільшення середньої площі під кривою ROC, покоління з найбільшим числовим значенням ROC (було 15). Оскільки загальна тенденція все ще триває, додаткові покоління можуть призвести до більш оптимального піднабору функцій у наступному поколінні.

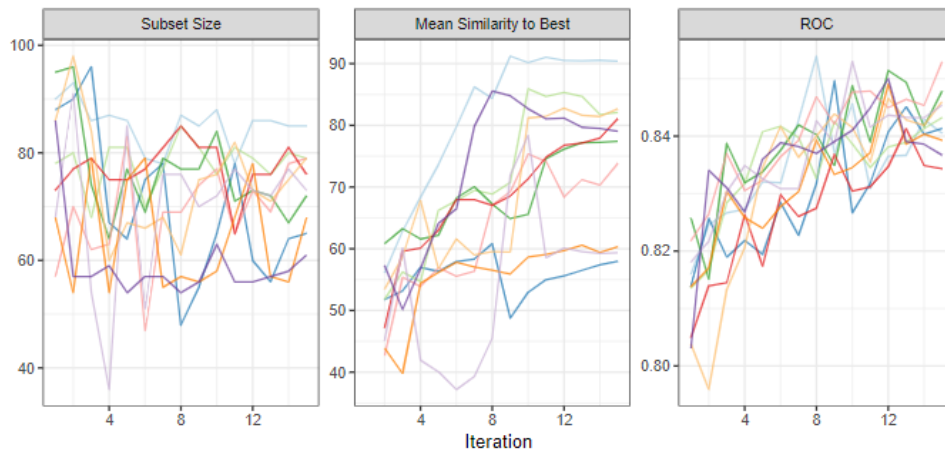


Рисунок 3.3 – Зміна розміру підмножини, подібності та продуктивності з часом для кожної зовнішньої повторної вибірки генетичного алгоритму

Остаточний пошук було проведено на всьому навчальному наборі і на рисунку 3.4 показано результати. Вони дуже схожі на внутрішні тенденції, показані раніше [17].

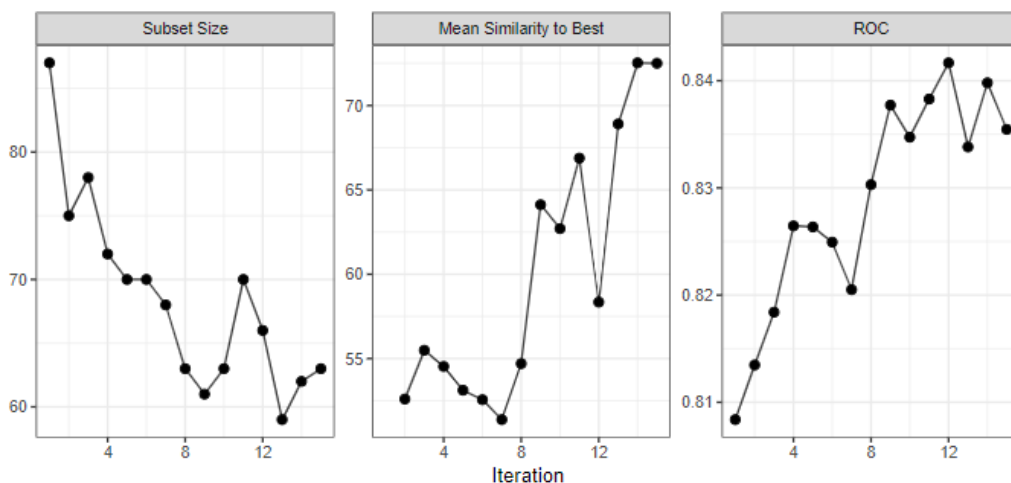


Рисунок 3.4 – Результати генетичного алгоритму для остаточного пошуку з використанням усього навчального набору

Моделі були побудовані з використанням оптимальних налаштувань параметрів для необмежених та обмежених ГА і застосовані до тестової вибірки. Площа під ROC-кривою для необмеженого ГА становила 0,831, тоді як відповідне значення для обмеженого ГА становило 0,804. Обидва ці значення є

більшими за аналогічні передискретизовані оцінки, але обидві моделі мають однаковий ранговий порядок ефективності. Рисунок 3.5 показує, що необмежена модель має рівномірно кращу чутливість і специфічність на всіх відсіченнях тестової вибірки, хоча важко сказати, чи суттєво ці результати відрізняються один від одного [17].

Щоб оцінити ефективність пошуку, було обрано 100 випадкових підмножин розміром 63 і для кожної розроблено наївну модель Байєса. Вибрана підмножина ГА показала кращі результати, ніж 100% випадково вибраних підмножин. Це вказує на те, що ГА знайшла корисну підмножину для прогнозування відповіді.

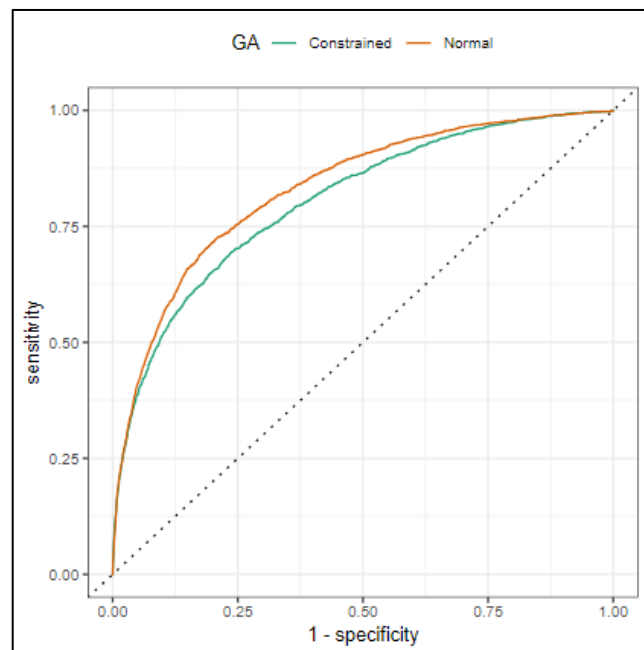


Рисунок 3.5 – Результати тестових наборів для двох моделей, отриманих за допомогою генетичних алгоритмів

Генетичні алгоритми часто вибирають більші підмножини ознак, ніж інші методи. Ймовірно, це пов'язано з побудовою алгоритму, згідно з яким, якщо ознака корисна для прогнозування, вона буде включена до підмножини ознак.

3.2 Програмна реалізація моделі методу автоматизованого підбору житла

Для нашого програмного забезпечення підбору житла потрібно створити базу даних для того щоб зберігати зареєстрованих користувачів.

Наша задача вимагає знайти оптимальний варіант житла для користувачів. Для початку створимо модель методу з об'єктами системи. Це головна сторінка системи з полями вводу реєстрації, початковий список усіх адрес з можливістю проживання, список усіх зареєстрованих клієнтів та взаємозв'язки між цими об'єктами.

Основні категорії житла бувають такі:

1. За формою власності:
 - приватне житло – належить фізичним особам на правах приватної власності;
 - державне житло – належить державі;
 - комунальне житло – належить територіальній громаді.
2. За призначенням:
 - житловий будинок – призначений виключно для проживання (гуртожиток, садибний будинок тощо);
 - багатоквартирний будинок – з декількома окремими квартирами.
3. За якістю:
 - елітне житло – високої якості, дороге;
 - масове / типове житло – стандартне, для широких верств населення;
 - соціальне житло – доступне житло для малозабезпечених.
4. За поверховістю:
 - багатоповерхове – висотка, 10 і більше поверхів;
 - середньоповерхове – від 5 до 9 поверхів;
 - малоповерхове – до 4 поверхів.

Першим питанням та важливим етапом для нашого завданням являється розподілити запити за категоріями. Категорії житла для споживачів за призначенням та пристосуванням:

1. Пристосоване житло – це житлові приміщення, спеціально обладнані під житло з нежитлових приміщень. Наприклад, квартири чи кімнати в перебудованих підвалах, горищах, складах тощо.

2. Спеціалізоване житло – призначене для певної категорії громадян. Наприклад, гуртожитки, будинки для літніх людей, сімейні дитячі будинки.

3. Соціальне житло – доступне житло, яке надається пільговим категоріям населення за рахунок держави чи місцевого бюджету.

4. Службове житло – житло в рамках установ чи підприємств, призначене для їхніх працівників. Наприклад, квартири військовослужбовців, вчителів, лікарів.

5. Житло комерційного використання – квартири та апартаменти, що здаються в оренду з метою отримання прибутку.

Для коректної роботи програми, користувачу потрібно вказати параметри житла, які будуть на вході в алгоритм. Першим етапом опитування будуть загальні показники житла:

1. Населений пункт (місто / селище міського типу / село). За адміністративним значенням населені пункти поділяють на:

- обласні центри – адміністративні центри областей;
- районні центри – центри районів;
- звичайні населені пункти.

2. Ціна;

3. Відстань від центру;

4. Тип будинку;

5. Поверховість будинку;

6. Квадратура;

7. Кількість кімнат;

8. Рік будівництва.

Важливо пам'ятати, що програмне забезпечення було розроблене для того, щоб розширити можливості вже існуючих програм для пошуку житла. Наступний етап для підбору житла – пройти опитування і зазначити особливі вимоги до житла. Під час війни потрібно забезпечити житлом різні категорії споживачів. Ось такі параметри повинно мати опитування:

- санітарні норми;
- балкон;
- централізоване опалення;
- індивідуальне опалення;
- підвал (пристосований для житла чи ні);
- пристосовані зали (школа, заводи тощо);
- близькість до військової частини;
- поштове відділення;
- супермаркети/магазини;
- квартира за правилом «двох стін»;
- відстань до зупинки місцевого транспорту;
- школа/ліцей;
- дошкільні навчальні заклади;
- наявність інтернету;
- графік відключень світла;
- електрична плита / газова плита;
- бойлер.

Для коректної роботи програми користувач повинен вказати точку на карті приблизного місця розташування помешкання, вартість, строк оренди, розмір житла, кількість кімнат, поверх, район та додаткові налаштування.

У генетичному алгоритмі популяція кандидатів на рішення (так звані індивідууми, істоти, організми або фенотипи) для оптимізаційної проблеми розвивається в напрямку кращих рішень. Кожне рішення-кандидат має набір властивостей (параметрів житла), які можна мутувати та змінювати; традиційно рішення представлені у двійковому вигляді (рядки з 0 і 1), але також можливі

інші кодування. Еволюція зазвичай починається з популяції випадково згенерованих особин і є ітеративним процесом, причому популяція в кожній ітерації називається поколінням. У кожному поколінні оцінюється придатність кожної особини в популяції; придатність зазвичай є значенням цільової функції в задачі оптимізації, що розв'язується. Більш придатні особини стохастично відбираються з поточної популяції, і геном кожної особини модифікується (рекомбінується та, можливо, випадково мутується), щоб сформувати нове покоління. Нове покоління варіантів рішень потім використовується в наступній ітерації алгоритму. Як правило, алгоритм припиняє роботу, коли або створено максимальну кількість поколінь, або досягнуто задовільного рівня придатності для популяції.

Типовий генетичний алгоритм має такі вимоги:

1. Генетичне представлення домену рішення;
2. Функція відповідності для оцінки області вирішення.

Стандартним представленням кожного кандидатського рішення є масив бітів (також називається набором бітів або рядком бітів). Масиви інших типів і структур можна використовувати по суті таким же чином. Основна властивість, яка робить ці генетичні представлення зручними, полягає в тому, що їхні частини легко вирівнюються завдяки їх фіксованому розміру, що полегшує прості операції кросинговеру. Також можна використовувати представлення змінної довжини, але в цьому випадку реалізація кросинговеру є більш складною. Деревоподібні представлення досліджуються в генетичному програмуванні, а представлення у формі графа досліджуються в еволюційному програмуванні; поєднання як лінійних хромосом, так і дерев досліджується в програмуванні експресії генів.

Після визначення генетичного представлення та функції відповідності ГА починає ініціалізацію популяції рішень, а потім покращує її шляхом повторюваного застосування операторів мутації, кросинговеру, інверсії та відбору. Протягом кожного наступного покоління частина існуючої популяції відбирається для відтворення нового покоління. Індивідуальні рішення

вибираються за допомогою процесу, заснованого на пристосованості, де, як правило, з більшою ймовірністю будуть обрані кращі рішення (що вимірюється функцією пристосованості). Певні методи відбору оцінюють придатність кожного рішення та переважно вибирають найкращі рішення. Інші методи оцінюють лише випадкову вибірку генеральної сукупності, оскільки перший процес може займати багато часу.

Функція відповідності визначається через генетичне представлення та вимірює якість представленого рішення. Функція придатності завжди залежить від проблеми. Наприклад, у задачі про рюкзак потрібно максимізувати загальну вартість предметів, які можна покласти в рюкзак певної фіксованої місткості. Представленням рішення може бути масив бітів, де кожен біт представляє окремий об'єкт, а значення біта (0 або 1) представляє, чи є об'єкт у рюкзаку чи немає. Не кожне таке представлення є дійсним, оскільки розмір предметів може перевищувати місткість рюкзака. Придатність розв'язку – це сума значень усіх об'єктів у рюкзаку, якщо представлення дійсне, або 0 в іншому випадку.

Наступним кроком є створення другого покоління популяції рішень із вибраних за допомогою комбінації генетичних операторів: кросинговер (також називають рекомбінацією) і мутація.

Для кожного нового рішення, яке буде отримано, пара «батьківських» рішень вибирається для створення з пулу, обраного раніше. Створюючи «дочірні» рішення з використанням наведених вище методів кросинговеру та мутації, створюється нове рішення, яке зазвичай має багато характеристик своїх «батьків». Нові батьки вибираються для кожної нової дитини, і процес триває, доки не буде створено нову популяцію рішень відповідного розміру. Хоча методи розмноження, які базуються на використанні двох батьків, більше «натхненні біологією», деякі дослідження припускають, що більше ніж два «батьки» генерують хромосоми вищої якості.

Ці процеси зрештою призводять до популяції хромосом наступного покоління, яка відрізняється від початкового покоління. Загалом середня придатність підвищиться за допомогою цієї процедури для популяції, оскільки

для розведення відбираються лише найкращі організми з першого покоління разом із невеликою часткою менш придатних рішень. Ці менш придатні рішення забезпечують генетичне різноманіття в генетичному пулі батьків і, отже, забезпечують генетичне різноманіття наступного покоління дітей.

Цей процес генерації повторюється, доки не буде досягнуто умови завершення. Загальні умови припинення процесу:

- знайдено рішення, яке задовольняє мінімальні критерії;
- фіксована кількість досягнутих поколінь;
- виділений бюджет (час/гроші обчислень) досягнуто;
- відповідність рішення найвищого рангу досягає або досягла такого рівня (плато), що послідовні ітерації більше не дають кращих результатів;
- ручна перевірка;
- комбінації вищевказаного.

Генетичний алгоритм простий у реалізації, але його поведінку важко зрозуміти. Зокрема, важко зрозуміти, чому цим алгоритмом часто вдається генерувати рішення високої придатності при застосуванні до практичних завдань. Гіпотеза будівельного блоку (ГББ) складається з таких елементів:

1. Опис евристики, яка виконує адаптацію шляхом ідентифікації та рекомбінації «будівельних блоків», тобто схем низького порядку з низькою визначальною довжиною з придатністю вище середньої.

2. Гіпотеза про те, що генетичний алгоритм виконує адаптацію, неявно та ефективно реалізуючи цю евристику.

Девід Едвард Голдберг описує евристику наступним чином: Короткі схеми низького порядку та схеми з високою відповідністю відбираються, рекомбінуються (перетинаються) і повторно відбираються, щоб утворити рядки потенційно вищої відповідності. У певному сенсі, працюючи з цими конкретними схемами (будівельними блоками), ми зменшили складність нашої проблеми; замість того, щоб будувати високопродуктивні рядки, випробовуючи всі можливі комбінації, ми створюємо все кращі рядки з найкращих часткових рішень минулих вибірок.

Мова програмування Python має потужні можливості для програмування баз даних. Python підтримує різні бази даних, такі як SQLite, MySQL, Oracle, Sybase, PostgreSQL тощо. Python також підтримує мову визначення даних (DDL), мову маніпулювання даними (DML) і оператори запиту даних. Стандартом Python для інтерфейсів баз даних є Python DB-API. Більшість інтерфейсів баз даних Python відповідають цьому стандарту.

Для нашого завдання використаємо базу даних SQLite мовою програмування Python. Це робиться за допомогою вбудованого в Python модуля sqlite3. Спочатку створюємо об'єкт підключення, який представляє базу даних, а потім створюємо декілька об'єктів курсору для виконання операторів SQL.

Наступний код Python показує, як підключитися до існуючої бази даних. Якщо бази даних не існує, вона буде створена, і, нарешті, буде повернено об'єкт бази даних:

```
import sqlite3
db = sqlite3.connect('server1.bd')
sql = db.cursor()
```

Якщо базу даних створено успішно, вона відобразить таке повідомлення: «Opened database successfully».

Генетичні алгоритми широко використовуються для розв'язання задач оптимізації з обмеженнями та без обмежень. Генетичні алгоритми використовують три основні типи правил: правила відбору, правила кросинговеру та правила мутації.

Існує три основні групи методів вирішення проблем пошуку:

- математичне програмування;
- евристичні методи;
- еволюційні методи пошуку.

До еволюційних методів пошуку належать генетичні алгоритми та методи генетичного програмування (Рисунок 3.6). Ці методи вдало поєднують переваги евристики (швидкий пошук близьких до оптимальних рішень) та забезпечують вихід з локальних екстремумів.

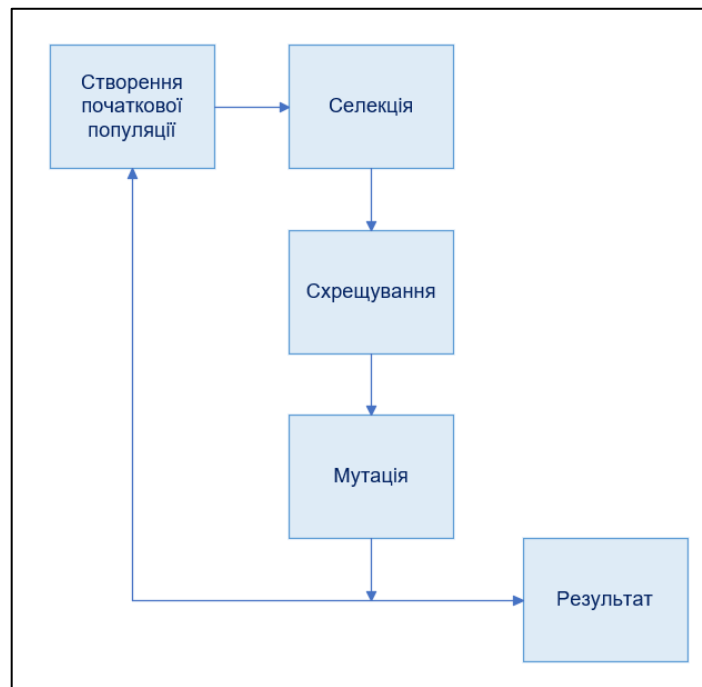


Рисунок 3.6 – Схема роботи генетичного алгоритму

У класичних генетичних алгоритмах початкова популяція генерується випадковим чином. Розмір популяції (N – загальна кількість хромосом) фіксований і не змінюється під час роботи алгоритму. Кожна особина генерується як послідовність з L бітів, де L – довжина індивідуального коду. Кожна особина є рішенням поточної задачі. Найбільш підходяща особина є можливим рішенням. Ці властивості дозволяють генетичним алгоритмам досягати найкращих результатів, які перевершують інші алгоритми оптимізації.

Алгоритм складається з трьох етапів: створення проміжних популяцій шляхом відбору з поточного покоління, відтворення проміжних поколінь шляхом кросинговеру та мутації.

Як правило, генетичні алгоритми роблять лише 5-10% або менше "помилки", що гарантує високу швидкість у комбінаторних задачах. На практиці генетичні алгоритми часто використовують у поєднанні з іншими методами, які можуть підвищити їхню ефективність.

Генетичний алгоритм у інтелектуальному аналізі даних є передовим методом класифікації даних. Класифікація даних складається з двох етапів, тобто

етапу навчання та етапу класифікації. Модель класифікації будується на етапі навчання, а на етапі класифікації модель передбачає вихід для наданого вхідного.

Генетичний алгоритм застосовує ту саму техніку в інтелектуальному аналізі даних – він інтерактивно виконує процес відбору, кросинговеру, мутації та кодування, щоб розвивати послідовне покоління моделей.

Компоненти генетичних алгоритмів бувають такі:

- популяція, що включає особин;
- механізм кодування або декодування індивідів;
- цільова функція та відповідний критерій оцінки придатності;
- процедура відбору;
- генетичні оператори, такі як рекомбінація або кросинговер, мутація;
- імовірність виконання генетичних операцій;
- техніка заміни;
- комбінація припинення.

На кожній ітерації алгоритм надає модель, яка успадковує свої риси від попередньої моделі та конкурує з іншими моделями, доки не виживе найбільш прогностична модель. Якщо ми порахуємо фази генетичного алгоритму, то розподілимо весь алгоритм на декілька фаз.

Отже, генетичний алгоритм – це метод класифікації, який є адаптивним, надійним і використовується в усьому світі в ситуаціях, коли область класифікації велика. Алгоритми оптимізують функцію відповідності на основі критеріїв, яким віддає перевагу аналіз даних, щоб отримати оптимальне рішення, наприклад:

- система виявлення знань;
- система MASSON.

Проте програма інтелектуального аналізу даних, заснована на генетичному алгоритмі, не така багата, як програма, заснована на нечітких наборах. Розглянемо класифікацію деяких систем на основі генетичного алгоритму, що використовується для аналізу даних.

Правила асоціації. Багатоцільовий ГА має справу з проблемами з кількома цільовими функціями та обмеженнями, щоб визначити оптимальний набір рішень. Жодне з рішень із цього набору не повинно існувати в просторі пошуку, яке може домінувати над будь-яким членом цього набору. Такі алгоритми використовуються для аналізу правил із великим простором пошуку з багатьма атрибутами та записами. Щоб отримати оптимальні рішення, багатоцільовий ГА виконує глобальний пошук із кількома цілями. Наприклад, поєднання таких факторів, як точність прогнозування, зрозумілість і цікавість.

Переваги:

- легко зрозумілий, оскільки він базується на концепції природної еволюції;
- класифікує оптимальне рішення з набору рішень;
- ГА використовує інформацію про житло замість похідної, щоб отримати оптимальне рішення;
- ГА підтримує багатоцільову оптимізацію;
- ГА – адаптивний алгоритм пошуку;
- ГА також працює в шумному середовищі.

Недоліки:

- неправильна реалізація може призвести до рішення, яке не є оптимальним;
- ітераційне впровадження фітнес-функції може призвести до обчислювальних проблем;
- ГА займає багато часу, оскільки має справу з великою кількістю обчислень.

Визначення найкращого варіанту житла є ключовим для проведення цього дослідження. Тому розглядається та розробляється програмна реалізація декількох компонентів для автоматизованого підбору тимчасового житла для категорій споживачів. UML діаграма користувача при налаштуванні алгоритму автоматизованого пошук оптимального житла має такий вигляд як на рис. 3.7:

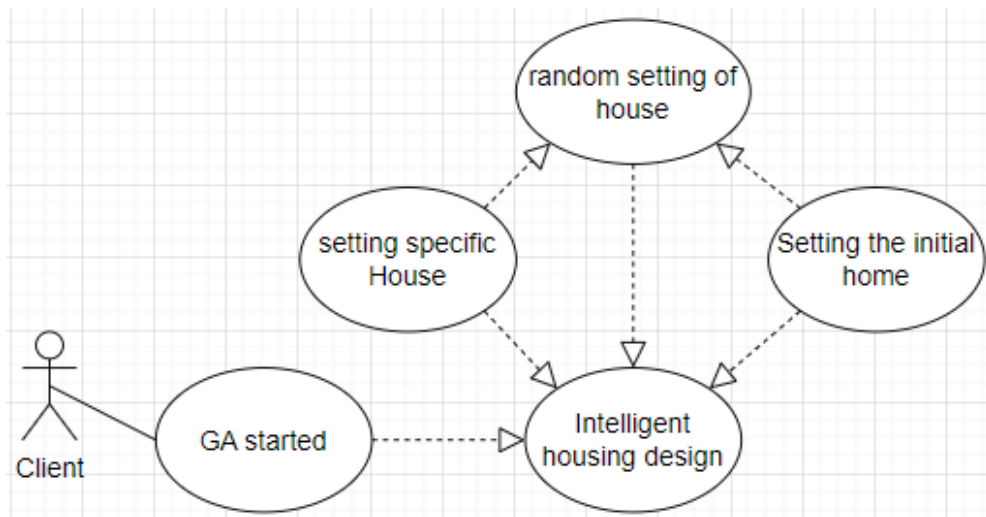


Рисунок 3.7 – UML діаграма вибору алгоритму підбору тимчасового житла

Зазвичай для початкової ініціалізації використовують метод випадкового генерування генів у популяції. У мові програмування Python реалізація такого підходу може бути за допомогою такої функції (див. у додатку В):

```
– def with_random_genotype(self).
```

Для даної реалізації опишемо інтерфейс компонента генерації можливих рішень з підбору тимчасового житла у загальному випадку (див. у додатку В):

Клас PopulationFactory з функціями:

```
– def __init__(self, individual_factory: IndividualFactory);
```

```
– def with_random_individuals(self, size: int);
```

```
– def with_individuals(self, individuals);
```

```
– def with_minimal_fitness_individuals(self, size: int).
```

У цьому випадку метод приймає типізовані функції для створення генів та популяцій, що дозволяє представляти їх не лише у числовому вигляді, але й у формі будь-якого складного типу. Реалізація компонента для автоматизованого підбору тимчасового житла відповідатиме визначеному контракту, але на вхід буде приймати набір допустимих значень, а саме адрес житлових приміщень.

Під час кожної ітерації алгоритму випадковим чином вибирається індивід з заданого переліку допустимих значень, після чого перевіряються всі критерії. Якщо індивід задовольняє всім критеріям, він включається до початкової

популяції. Алгоритм роботи зображено на Рисунку 3.8. Повний код реалізації компоненту наведений в Додатку В.

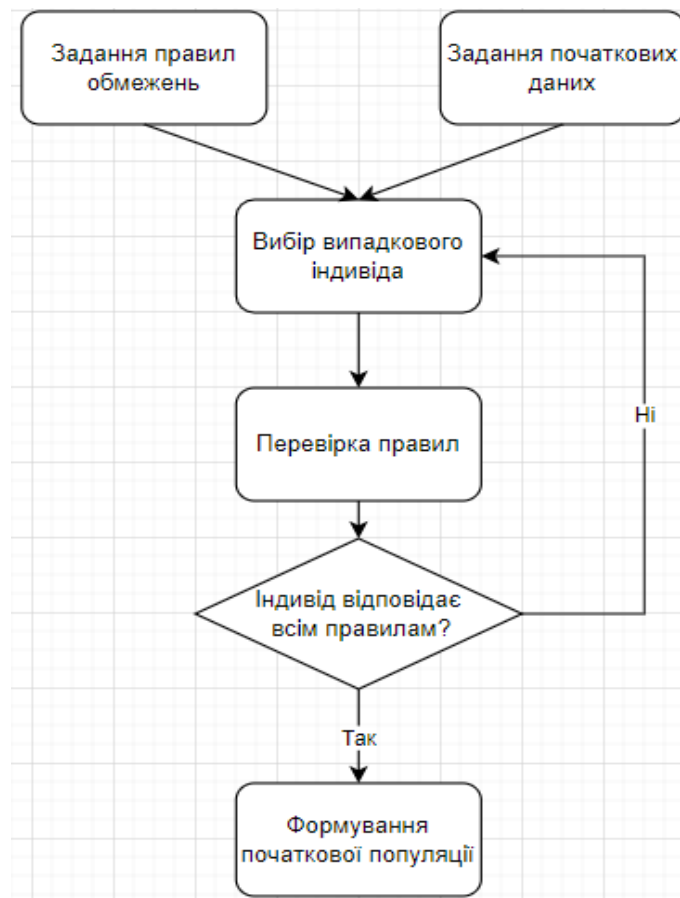


Рисунок 3.8 – Алгоритм інтелектуального формування початкової популяції

В системі, яку ми розробляємо, компонент визначення функції відповідності буде представлений інтерфейсом, що визначає лише один метод – Evaluate (див. у додатку В).

Клас `FitnessEvaluator` з функціями:

- `def __init__(self, genotype_decoder: GenotypeDecoder);`
- `def evaluate(self, genotype: str).`

Компонент селекції відповідає за вибір наступного покоління, базуючись на раніше обчислених значеннях в компоненті визначення відповідності. Раніше були описані різні варіанти реалізації алгоритмів селекції, проте важливо визначити структуру компоненту у вигляді контракту для визначення та підтримки нових реалізацій (див. у додатку В).

Клас ParentSelector з функцією:

- def select_parents(self, population: Population).

У цьому випадку, компонент на вході отримує набір усіх хромосом з обчисленими значеннями відповідності та розмір популяції в конкретному поколінні. В залежності від обраної реалізації, алгоритм буде базуватися на виборі (зазвичай випадковому) індивідів для наступної популяції, які відсортовані за значеннями фітнес-функції.

Компонент кросинговеру разом з компонентом мутації формують математичну основу роботи генетичного алгоритму. Вони допомагають забезпечити різноманітність індивідів у популяції, тому правильний вибір реалізації компонентів є важливим. Компонент кросинговеру повинен приймати на вхід дві хромосоми та змінювати значення їх генів (див. у додатку В).

Клас SinglePointCrossover з функціями:

- def __init__(self, individual_factory: IndividualFactory);
- def crossover(self, parent_1: Individual, parent_2: Individual);
- def __new_genotype(self, crossover_point: int, parent_1: Individual, parent_2: Individual).

Мутація додає нову інформацію у випадковий спосіб до процесу генетичного підбору і, зрештою, допомагає уникнути зациклення на локальних оптимумах. Реалізація програмної компоненти виглядає так (див. у додатку В).

Клас Mutator з функціями:

- def __init__(self, individual_factory: IndividualFactory);
- def mutate(self, individual: Individual).

Генерування підбору тимчасового житла випадковим чином, які розв'язуються за допомогою генетичних алгоритмів підходить для реалізації задачі в швидкому порядку. Проте іноді такий метод може бути не оптимальним, тому вводиться концепція автоматичного підбору тимчасового житла. У цьому випадку початкові покоління може бути сформоване відповідно до завдань алгоритму. Загальний результат підбору найкращого варіанту житла виводиться на екран (див. Рисунок 3.9 та Рисунок 3.10).

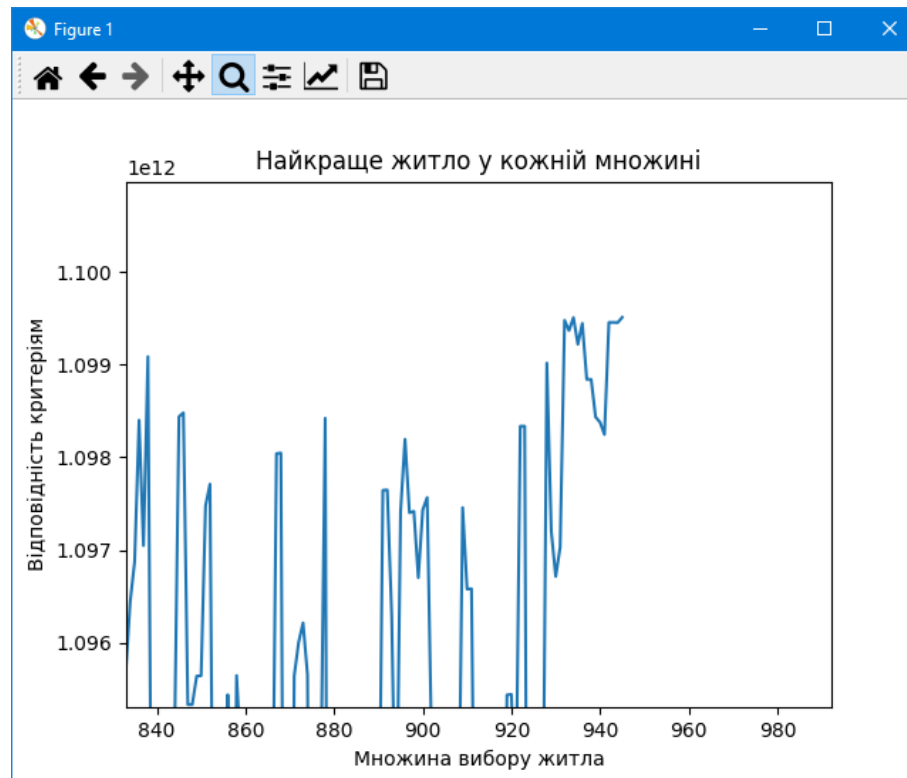


Рисунок 3.9 – Пошук найкращого рішення (житла)

```

Оболонка x
>>> %Run ex_gen_alg.py

Win!
Зупинилися на поколінні 944. Найпристосованіша особина:
Individual/genotype = 11111111111111111111 Fitness = 1099509530625

```

Рисунок 3.10 – Вивід результату підбору (житла)

Перевага використання ГА полягає в його простоті та швидкості. Як надійний алгоритм підбору, генетичний алгоритм швидко знаходить найкраще рішення в складній задачі високої розмірності.

3.3 Обґрунтування вибору середовища розробки

Для реалізації генетичного алгоритму було обрано мову програмування Python. Алгоритм може бути використаний для синхронізації, розподілу частот,

мікропроцесорних регістрів, обчислення похідних та чисельного розпаралелювання. Це пов'язано, головним чином, з його популярністю у сфері науки про дані та автоматизації даних, де обрані завдання є найбільш затребуваними; Python – мова з відносно простим та інтуїтивно зрозумілим синтаксисом і є вигідною у випадках, коли необхідно модифікувати будь-який з параметрів алгоритму або функцій, таких як кросовери, мутації тощо; а ГА – гнучкий інструмент, який необхідно адаптувати під кожну конкретну задачу.

Алгоритм написаний мовою Python. Однак використано бібліотеку операторів (для сортування словників), функції `random()` та `choice()` у модулі `random`, `perf_counter()` у модулі `time` та модуль `json` для роботи з файлами.

Як вже було сказано, хромосома – це список, що містить кольори її вершин. У Python кількість кольорів можна легко отримати, перетворивши список на множину (у множині немає двох однакових елементів): `len(set(gen))`, де `gen` – це хромосома. Вносити зміни в Python дуже просто, але важливо пам'ятати, що списки є змінюваними структурами даних. Реалізований алгоритм не використовує жодних сторонніх бібліотек. Наявність об'єктно-орієнтованої структури та коментарів дозволяє легко розширювати функціонал за потреби.

Модуль `random` – це вбудований модуль для генерації псевдовипадкових величин. Його можна використовувати для виконання випадкових дій, таких як отримання випадкового числа, вибір випадкового елемента зі списку, випадкове перемішування елементів тощо.

Модуль `pyGAd`, за допомогою якого можна створювати, запускати, зберігати та завантажувати екземпляри генетичного алгоритму. Можна розв'язувати одно- та багатоцільові оптимізаційні задачі.

Методи батьківського вибору. Клас `ParentSelection` у `pyGAd.utils.parent_selection` модуля має кілька методів для вибору батьків, які спаровуватимуться для отримання потомства. Усі ці методи приймають однакові параметри, а саме:

- `fitness`: значення відповідності рішень у поточній популяції;
- `num_parents`: кількість батьків, яку потрібно вибрати.

Ці методи повертають масив вибраних батьків.

- `steady_state_selection()` – вибирає батьків за допомогою методу стаціонарного відбору;
- `rank_selection()` – вибирає батьків за технікою рангового відбору;
- `random_selection()` – вибирає батьків випадковим чином;
- `tournament_selection()` – вибирає батьків методом турнірного відбору;
- `roulette_wheel_selection()` – вибирає батьків за допомогою техніки вибору колеса рулетки;
- `stochastic_universal_selection()` – вибирає батьків за допомогою техніки стохастичного універсального відбору;
- `nsGA2_selection()` – вибирає батьків для алгоритму NSGA-II для розв'язування задач багатоцільової оптимізації. Він вибирає батьків, ранжуючи їх на основі сортування без домінування та відстані скупчення;
- `tournament_selection_nsGA2()` – вибирає батьків для алгоритму NSGA-II для розв'язування задач багатоцільової оптимізації. Він вибирає батьків, використовуючи методику турнірного відбору, застосовану на основі сортування без домінування та відстані скупчення;

Клас `Crossover pyGAd.utils.crossover` модуля підтримує кілька методів для застосування кросинговеру між вибраними батьками. Усі ці методи приймають однакові параметри, а саме:

- `parents`: батьки, яких треба спарувати для отримання потомства;
- `offspring_size`: розмір потомства для виробництва.

Усі ці методи повертають масив отриманих нащадків. Далі наведено підтримувані методи кросинговеру.

- `single_point_crossover()` – застосовує одноточковий кросовер. Він випадковим чином обирає точку, де буде схрещування між парами батьків;
- `two_points_crossover()` – застосовує 2-точковий кросовер. Випадково вибирає 2 точки, в яких відбувається схрещування між парами батьків;
- `uniform_crossover()` – застосовує рівномірний кросовер. Для кожного гена випадковим чином вибирають одного з двох батьків, які спаровуються, і з нього копіюють ген;

– `scattered_crossover()` – застосовує розсіяний кросовер. Він випадковим чином вибирає ген від одного з 2 батьків.

Клас `Mutation` у `pyGAd.utils.mutation` модуля підтримує кілька методів застосування мутації. Усі ці методи приймають той самий параметр:

– `offspring`: нащадки мутують.

Ці методи повертають масив мутованих нащадків.

Методи мутації:

– `random_mutation()` – Застосовує випадкову мутацію, яка змінює значення деяких генів випадковим чином. Кількість генів вказується відповідно до атрибутів `mutation_num_genes` або `mutation_percent_genes`. Для кожного гена вибирається випадкове значення відповідно до діапазону, визначеного 2 атрибутами `random_mutation_min_val` та `random_mutation_max_val`. Випадкове значення додається до вибраного гена;

– `swap_mutation()` – Застосовує мутацію обміну, яка змінює значення 2 випадково вибраних генів;

– `inversion_mutation()` – Застосовує інверсійну мутацію, яка вибирає підмножину генів і інвертує їх;

– `scramble_mutation()` – Застосовує мутацію `scramble`, яка вибирає підмножину генів і переміщує їх порядок випадковим чином;

– `adaptive_mutation()` – Застосовує адаптивну мутацію, яка вибирає підмножину генів і випадковим чином переміщує їх порядок;

– `best_solution()` – Повертає інформацію про найкраще рішення, знайдене генетичним алгоритмом. Він приймає такі параметри: `pop_fitness=None`: необов'язковий параметр, який приймає список значень відповідності рішень у популяції. Якщо `None`, то `cal_pop_fitness()` метод викликається для обчислення значень відповідності `self.population`. Застосовуємо `GA_instance.last_generation_fitness` для використання останнього значення придатності та пропуску повторного розрахунку придатності населення.

Він повертає наступне:

– `best_solution`: Найкраще рішення в поточній популяції;

- `best_solution_fitness`: Фітнес-цінність найкращого рішення;
- `best_match_idx`: Індекс найкращого рішення в поточній популяції;
- `plot_fitness()` – Цей метод, який раніше називався `plot_result()`, створює, показує та повертає цифру, яка показує, як значення придатності змінюється за поколінням. Він працює лише після завершення принаймні 1 покоління. Якщо жодне покоління не завершено (принаймні 1), виникає виняток.
- `plot_new_solution_rate()` – Метод `plot_new_solution_rate()` створює, показує та повертає цифру, яка показує кількість нових рішень, досліджених у кожному поколінні. Цей метод працює лише `save_solutions=True` в конструкторі класу `pyGAd GA`. Він працює лише після завершення принаймні 1 покоління. Якщо жодне покоління не завершено (принаймні 1), виникає виняток;
- `save()` – Зберігає екземпляр генетичного алгоритму.

Приймає наступний параметр:

- `filename`: ім'я файлу для збереження примірника. Розширення не потрібне.

Крім методів, доступних у `pyGAd` класі, у обговорюються функції, доступні в `pyGAd`. До цього часу існує лише одна функція під назвою `load()`.

- `pyGAd.load()` – Читає збережений екземпляр генетичного алгоритму. Це не метод, а функція, яка має відступ під `pyGAd` модулем. Отже, модуль `pyGAd` може викликати його таким чином: `pyGAd.load(filename)`. Приймає наступний параметр – `filename`: ім'я файлу без розширення, що містить збережений екземпляр генетичного алгоритму. Повертає екземпляр генетичного алгоритму.

Мова `Python` стала гарним вибором для розробки програми пошуку житла. Вона має простий синтаксис, велику кількість бібліотек та фреймворків, що полегшують розробку. Існують деякі аспекти використання `Python` для розробки програми пошуку житла:

Веб-фреймворки: `Flask` та `Django` – це популярні веб-фреймворки для розробки веб-додатків на `Python`. `Flask` є легшим та більш гнучким, в той час як `Django` надає повноцінну архітектуру для швидкої розробки.

Бібліотеки для роботи з базами даних: SQLAlchemy – це потужна бібліотека для роботи з реляційними базами даних. Peewee – це ще одна легка бібліотека для взаємодії з базами даних.

Web-скрапінг: Для отримання даних про житло з інтернет-ресурсів можна використовувати бібліотеки, такі як BeautifulSoup чи Scrapy.

Робота з API: Для взаємодії з різними API, наприклад, сервісами нерухомості, використовується бібліотека requests.

Мова програмування для аналізу та обробки даних: Python часто використовується для аналізу даних, і для цього можна використовувати бібліотеки, такі як Pandas, NumPy та Matplotlib.

Можливості розширення: Python легко поєднується з іншими мовами програмування, що може бути корисно для виконання конкретних операцій, які можуть бути краще реалізовані на інших мовах.

Використання Python для програми пошуку житла може полегшити розробку завдяки великій спільноті, яка підтримує мову, та великій кількості доступних бібліотек.

Висновки до Розділу 3

У цьому розділі описано реалізацію веб-додатку методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

Створено модель методу автоматизованого підбору житла, яка складається з ініціалізації, оцінки пристосованості, селекції, формування нової множини (популяції) і вибору найкращого житла (хромосоми), які базується на основі генетичного алгоритму.

Обґрунтовано вибір середовища розробки програмного забезпечення. Нашу задачу можливо вирішити на багатьох мовах програмування, але ми обрали мову Python. Вона має усі потрібні нам бібліотеки, широко використовується в теперішній час, підтримується великою спільнотою.

Розділ 4. Дослідження методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом

4.1 Підготовка даних для програмних компонентів методу та прикладне тестування методу

Для вирішення проблем з автоматизованим пошуком житла для категорій споживачів було створено програмне забезпечення на основі генетичного алгоритму. Під час функціонального дослідження методу підбору тимчасового житла для категорій споживачів було створено програмне забезпечення. Воно складається з чотирьох форм для заповнення. Вікна програми можна розглянути на рисунках 4.1-4.4.

Метод автоматизованого підбору пристосований для категорій споживачів з різним достатком, або для допомоги людям, які примусово покинули свої домівки. Користувачу потрібно обрати свою категорію, для детальної інформації див табл. 4.1:

Таблиця 4.1 – Категорії споживачів

Категорія	Опис
ВОР (Base of the Pyramid)	Люди, які живуть менше ніж на 100 грн. в день. Це бідні верстви населення в країнах, що розвиваються.
Малозабезпечені	Люди з низькими доходами в межах прожиткового мінімуму. У розвинутих країнах це ті, хто живе на соціальну допомогу.
Середній клас	Люди із середнім рівнем достатку, які становлять основу споживчого ринку.
Заможні	Люди з високими доходами, які можуть собі дозволити дорогі товари і послуги преміум класу.
МННWI (Millionaires next door)	Мільйонери і мультимільйонери, еліта суспільства за рівнем багатства.

	Можуть дозволити собі наддорогі речі.
--	---------------------------------------

Для всіх категорій було створено однакові параметри підбору. Кожному параметру потрібно вказати коефіцієнт важливості. Коефіцієнт – це числовий показник, що використовується для відображення відносної важливості або пріоритетності певного фактору чи параметру в рамках аналізу, оцінки, прийняття рішень тощо. Перша форма з параметрами житла вимагає від споживача заповнити поля відповідним чином (див табл. 4.2):

Таблиця 4.2 – Форма введення параметрів житла

Параметр	Опис
Ціна	Потрібно вказати мінімальну ціну та максимальну ціну.
Квадратура	Потрібно вказати мінімальний розмір приміщення та максимальний розмір приміщення.
Поверх	Вказати від 1 до 10.
Відстань від центру	Вказати від 1 км до 10 км.
Тип будинку	Цегляний; Каркасний; Силікатної цегли; Панелі; Піноблоку; Моноліту; та інші.
Кількість кімнат	Вказати від 1 кімнати до 5 кімнат.
Населений пункт	Обрати – місто, СМТ або село.
Кількість днів оренди	Вказати: менше тижня; менше місяця; менше півроку; менше року; більше року.

Після введення звичайних критеріїв житла потрібно детальніше описати властивості житла, бо наш метод відрізняється від звичайних методів підбору, а

саме в стані війни повинні бути доступні такі розширені параметри, що й описується в таблиці 4.3. Тут достатньо залишити помітку наявності параметру та визначити пріоритет критерію:

Таблиця 4.3 – Розширені параметри житла

Розширені параметри	Опис
Графік відключення світла	Є окремі райони, де графік відключень світла не активний, наприклад біля лікарень.
Облаштований підвал	Наявність підвалу з можливостями тимчасового проживання, наявність води та їжі в підвалі.
Опалення	Індивідуальне, Централізоване.
Плита	Газова, електрична
Балкон	Наявність (є/немає)
Ванна	Наявність (є/немає)
Душова кабіна	Наявність (є/немає)
Бойлер	Наявність (є/немає)
Меблі	Наявність (є/немає)
Теплі ковдри	Наявність (є/немає)
Інтернет	Наявність (є/немає)
Їжа для критичних умов	Наявність (є/немає)
Військова частина	Поруч (є/немає)
Лікарня	Поруч (є/немає)
Школа/ліцей	Поруч (є/немає)
Продуктові магазини	Поруч (є/немає)
Поштове відділення	Поруч (є/немає)
ДНЗ	Поруч (є/немає)
Заводи хімічної промисловості	Поруч (є/немає)

База даних зі списком житлових адрес містить в собі усі можливі адреси. Від самих дорогих до самих дешевих і безкоштовних. Приклад множини з адресами можна розглянути в таблиці 4.4. Усі параметри житла було описано детальніше в підрозділі 2.2.

Таблиця 4.4 – Список адрес з параметрами в базі даних

Адреса	Ціна	Квадратура	Кількість кімнат	Тип будинку	Населений пункт
вулиця Івана Павла II	5 000 грн	20 м ²	2	Цегла	Місто
вулиця 2-а Комунарська 2-а	7 900 грн	30 м ²	1	Панель	Село
вулиця Атерлея	18 500 грн	40 м ²	3	Піноблок	Село
вулиця 2-а Лугова	24 600 грн	60 м ²	3	Каркасний	Місто
вулиця 2-а Нова	31 800 грн	50 м ²	2	Цегла	СМТ
1-й провулок Ентузіастів	50 000 грн	100 м ²	5	Моноліт	Місто
вулиця Аеродромна	Безкоштовно	30 м ²	2	Силікатна цегла	СМТ

Друга БД створена для користувачів щоб зберегти їхні контактні дані. Також, за їхньою поштою та номером телефону закріплюються вибрані критерії житла. БД для користувачів містить інформацію про користувача та параметри підбору (див в табл. 4.5).

Таблиця 4.5 – БД для користувачів.

Пошта	Номер тел.	Ціна	Квадратура	Кількість кімнат	Тип будинку	Населений пункт
poshta@gmail.com	380964042452	2 000 грн.	30 м ²	1	Цегла	Місто
poshta2@gmail.com	0964240254	Безкоштовно	50 м ²	2	Піноблок	Село

Рисунок 4.1 – Форма Реєстрації

Рисунок 4.2 – Форма з стандартними параметрами житла

Інформаційна система пошуку житла

Розширені параметри

0.6 Відстань від зупинки місцевого транспорту <1 км

0.3 Графік відключень світла

0.8 Можна з тваринами

Опалення: 0.4 Централізоване 0.5 Індивідуальне

Плита: 0.5 Газова 0.2 Електрична

Наявність:

0.1 Балкон 0.6 Ванна

0.8 Бойлер 0.7 Меблі

0.9 Підвал 0.4 Інтернет

0.2 Душова кабіна 0.2 Теплі ковдри

Поруч:

0.5 Військова частина 1.0 ДНЗ

0.2 Поштове відділення 0.8 Заводи

0.9 Супермаркет 0.6 Лікарня

1.0 Школа/ліцей

Почати підбір Назад

Рисунок 4.3 – Форма з розширеними параметрами житла

Детальну інформацію можна розглянути натиснувши «Детальніше» (рисунок 4.4).

Інформаційна система пошуку житла

Підбір житла завершився

Адреса:
вулиця Пекарська, 10, Львів, Львівська область, 79000

Контактний номер:
+(380)96 404 2452

[Детальніше](#)

Рисунок 4.4 – Результат підбору житла

Метод підбору використовує генетичний алгоритм, він допомагає обрати один варіант житла в кінцевому результаті. Якщо користувача не влаштовує результат, можна повернутися назад і почати підбір ще раз.

4.2 Функціональне дослідження та визначення ефективності методу підбору тимчасового житла для категорій споживачів

Для перевірки коректності роботи програмного забезпечення методом автоматизованого підбору тимчасового житла для категорії споживачів, було проведено п'ять тест-кейсів. Перший тест-кейс описано в Таблиці 4.6.

Таблиця 4.6 – Тест-кейс №1

Перевірка успішної реєстрації	
Послідовність виконання	Обробка події
<ol style="list-style-type: none"> 1. Запуск додатку; 2. Ввести свою електронну адресу; 3. Ввести свій контактний номер телефону; 4. Натиснути «Далі». 	<p>Відкриття першого вікна з полями для вводу даних. Після натискання на кнопку «Далі» електронний адрес та номер телефону зберігається в БД. Якщо реєструватися повторно, то програма повідомить користувача, що такий запис вже існує.</p>
Результат тестування: пройдено успішно	

Другий тест-кейс описано в Таблиці 4.7. Метод автоматизованого підбору тимчасового житла використовує габаритну БД, тому потрібно перевірити як вона працює. Є ймовірність того, що підбір буде працювати повільно.

Таблиця 4.7 – Тест-Кейс №2

Перевірка БД житлових адрес

Послідовність виконання	Обробка події
<ol style="list-style-type: none"> 1. Запуск додатку; 2. Введення даних для реєстрації; 3. Натиснути «Далі»; 4. Обрати категорію та встановити їхню пріоритетність; 5. Натиснути «Далі»; 6. Заповнити форму «Параметри житла стандартного набору»; 7. Натиснути «Розширені параметри»; 8. Заповнити форму «Розширені параметри»; 9. Натиснути «Почати підбір»; 10. Зачекати до завершення підбору тимчасового житла; 11. Для детальної інформації про житло натисніть кнопку «Детальніше» 	<p>Якщо реєстрація пройшла успішно, додаток по черзі запускає форми, які потрібно заповнити користувачу. Після вводу усіх критеріїв, кнопка «Почати підбір» відкриває БД з житловими адресами та починає підбирати відповідне житло по параметрам. Після підбору запускається вікно з підбраною адресою.</p>
<p>Результат тестування: пройдено успішно (є ймовірність того, що підбір житла змусить користувача зачекати на декілька секунд більше ніж інша програма)</p>	

Третій тест-кейс описано в Таблиці 4.8. Тестуємо ПЗ тоді, коли потрібно підібрати тимчасове житло для людини з обмеженими можливостями.

Таблиця 4.8 – Тест-кейс №3

Підбір житла з нетиповими вхідними параметрами	
Послідовність виконання	Обробка події
<ol style="list-style-type: none"> 1. Запуск додатку; 2. Реєстрація; 3. Натиснути «Далі»; 	<p>Після успішної реєстрації користувач обирає категорію пристосованого житла, адже для його випадку йому потрібні комфортні умови. Вводимо</p>

<p>4. Обираємо категорію житла «Пристосоване»;</p> <p>5. Натиснути «Далі»;</p> <p>6. Підбираємо параметри з особливими умовами;</p> <p>7. Натиснути «Розширені параметри»;</p> <p>8. Додаємо додаткові умови комфортного житла для людини з обмеженими можливостями;</p> <p>9. Натиснути «Почати підбір»;</p>	<p>низьку ціну житла. На формі з розширеними можливостями обираємо особливі параметри, такі як: доступність для інвалідних візків (широкий вхідний прохід, ліфт, облаштований підвал, спеціальне ліжко, додаткові технічні особливості для людей з обмеженими можливостями). Запуск вікна з результатом підбору.</p>
<p>Результат тестування: 2 випадки – не знаходило житла з такими умовами, 4 випадки – пройдено успішно.</p>	

Розглянемо наступний тест-кейс №4, наприклад, дано такий набір параметрів (таблиця 4.9). і враховуючи ці параметри починаємо підбір житла в трьох додатках, результат тест-кейсу в таблиці 4.10:

Таблиця 4.9 – Перелік вимог до житла

Параметри	Вибрані параметри
Категорія	пристосоване
Ціна	від 3000 до 6000
Квадратура	від 40 м ² до 70 м ²
Поверх	Тільки 1
Тип будинку	Цегла
Кількість кімнат	1
Додаткові параметри	<ul style="list-style-type: none"> – можливе житло з власником; – від зупинки менше ніж 1 км; – можна з тваринами;

	<ul style="list-style-type: none"> – для людей похилого віку; – для людей з обмеженими можливостями; – лікарня поряд; – облаштований підвал.
--	--

Таблиця 4.10 – Тест-кейс №4

Підбір житла в трьох програмах з однаковими вхідними параметрами	
Послідовність виконання	Обробка події
<ol style="list-style-type: none"> 1. Запуск ПЗ підбору житла (ГА); 2. Запуск сайту booking.com; 3. Запуск сайту dom.gia.com; 4. Налаштування вимог житла в трьох системах підбору по параметрам як в табл. 4.4; 	<p>Після запуску трьох програм, вводимо однакові вхідні параметри і починаємо підбір.</p>
<p>Результат тестування: метод автоматизованого підбору житла знайшов тимчасове житло в центрі міста, разом з власником, поверховість складає один поверх, умови доступні для людей похилого віку та з обмеженими можливостями та іншими параметрами. Програма booking знайшла лише однокімнатне житло 50м², можна з тваринами та облаштований підвал. Програма dom.gia знайшла житло за 5500 грн. 40 м², 1 поверх, цегляний будинок, але житло не пристосоване для людей з обмеженими можливостями.</p>	

Якщо враховувати час підбору житла, підбір нашим методом не підходить. Але якщо витратити часу трохи більше ніж на звичайні способи підбору, то наше програмне забезпечення покаже кращий варіант житла.

Тест-кейс №5 продемонструє на скільки відсотків відповідності метод автоматизованого підбору житла зможе підібрати житло. Припустимо, якщо метод підбере житло по всім параметрам, які ввів користувач, то відсоток відповідності буде дорівнювати 100. Наприклад, створимо таблицю з декількома параметрами та розділимо 100% для всіх критеріїв (таблиця 4.11).

Таблиця 4.11 – Відсоток відповідності для декількох параметрів житла.

Коефіцієнт значимості	Параметр	Оцінка параметру	Відсотки
1.0	Ціна	13 000 грн.	25%
0.5	Квадратура	70 м ²	25%
1.0	Кількість кімнат	4	25%
8.0	Населений пункт	СМТ	25%

Коефіцієнт значимості параметра враховується після розподілу 100% для всіх обраних параметрів, бо деякі параметри можуть бути вилучені з множини. Тепер дивимось, ціна має дорівнювати максимум 13 000 грн. та кількість кімнат має бути – 4. Генетичний алгоритм тепер не може вилучити з множини квартири які, мають ці два параметри разом в одні адресі.

Припустимо, метод підібрав житло, в якому присутні такі чотири параметри з цієї таблиці: ціна – 13 000 грн, кількість кімнат – 4, населений пункт – СМТ, квадратура – 60 м². Висновок: 75% параметрів житла відповідає введеним критеріям користувача, та 25% не відповідає бажаним критеріям користувача.

Повторимо такі ж підрахунки рівня відповідності параметрам з наступними сайтами підбору житла: dom.ria.com, booking.com. Таблиця з введеними параметрами буде мати інший вигляд параметрами тому, що існуючі системи підбору житла не дають можливості користувачу встановити коефіцієнт значимості параметру (див табл. 4.12).

Таблиця 4.12 – Відсоток відповідності для декількох параметрів житла для dom.ria.com та booking.com.

Параметр	Оцінка параметру	Відсотки
Ціна	13 000 грн.	25%
Квадратура	70 м ²	25%

Кількість кімнат	4	25%
Населений пункт	СМТ	25%

Результатом підбору dom.ria.com виступає два варіанти квартири, в такому випадку користувач обирає квартиру, яка має наближені параметри до введених: ціна – 9 500 грн; квадратура – 64 м²; кількість кімнат – 4; населений пункт – місто. Співпали два параметри, ціна та кількість кімнат, тому рівень відповідності параметрам дорівнює 50%.

Результатом системи підбору житла booking.com маємо список з 8-ми варіантів апартаментів. Сайт розрахований на проживання у готелях та апартаментах під час відпочинку або перельотів. Мала БД не розрахована для людей переселенців, та не зможе підібрати житло для різних категорій споживачів в критичних умовах. З підібраних варіантів розглянемо одну адресу: ціна – 19 000 грн., квадратура – 40 м²; кількість кімнат – 2; населений пункт – СМТ. Так як співпадає лише один параметр, рівень відповідності параметрам буде дорівнювати 25%.

Отже, побудуємо таблицю для порівняння рівня відповідності параметрам для трьох систем підбору житла (таблиця 4.13).

Таблиця 4.13 – Порівняльна таблиця рівня відповідності параметрів.

Система підбору житла	Відсоток відповідності
Метод автоматизованого підбору житла для категорії споживачів за генетичним алгоритмом	75%
dom.ria.com	50%
booking.com	25%

Зважаючи на приблизний розрахунок рівня відповідності параметрам розробленого методу автоматизованого підбору житла та двом існуючим середовищам для пошуку житла, можемо побудувати діаграму(рисунк 4.5).

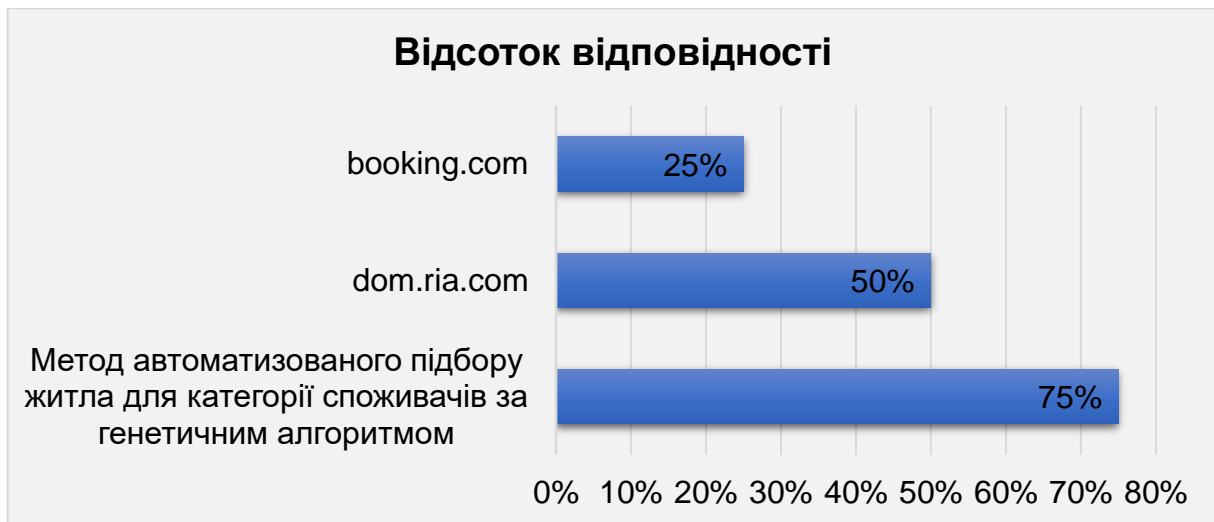


Рисунок 4.13 – Діаграма відповідності параметрів

Порівнюючи ефективність трьох систем підбору житла, можна зробити висновок. Існуючі програми пошуку житла не визначають пріоритетність параметрів, мають обмежену БД адрес та не розраховані для категорії споживачів. Середня відповідність параметрів існуючих сайтів, з якими ми працювали (формула 4.1):

$$R_{\text{сер.}} = \frac{50\% + 25\%}{2} = 62.5\% \quad (4.1)$$

Ефективність методу автоматизованого підбору житла для категорій користувачів вища ніж ефективність двох інших програм, в середньому на 12,5% (формула 4.2).

$$R_p = 75\% - 62.5\% = 12.5\% \quad (4.2)$$

Після проведення п'яти тест-кейсів та розрахунку ефективності методу можемо зробити висновок, що метод автоматизованого підбору житла для користувачів за генетичним алгоритмом був покращеним та наразі може використовуватися в цілях підбору житла будь-ким із нас.

Висновки до розділу 4

В даному розділі проведено дослідження автоматизованого методу підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

Було розроблено набір тестових сценаріїв для верифікації роботи системи. Перевірку пройшли коректність інтерфейсу, функціональність основних модулів, а також відповідність результатів роботи генетичного алгоритму теоретичним очікуванням.

З метою порівняльного аналізу ефективності запропонованого підходу було проведено підбір житла за методом генетичного алгоритму, dom.gia.com та booking.com. Проведено експерименти на тестових даних для дослідження переваг та недоліків розробленого методу.

Отримані результати демонструють переваги використання генетичного алгоритму для автоматизованого підбору тимчасового житла для категорій споживачів.

Загальні висновки

Результатом виконання кваліфікаційної роботи магістра є розроблений метод автоматизованого підбору тимчасового житла для категорії споживачів за генетичним алгоритмом.

При цьому були вирішені наступні задачі:

- Проведено дослідження предметної області підбору житла для категорій споживачів. Проведено аналіз проблеми пошуку житла за різними категоріями та параметрами. Визначено основні характеристики, на які будемо орієнтуватися при підборі тимчасового житла, та пріоритетні показники кожного параметру помешкання.

- Досліджено існуючі методи підбору житла та програмне забезпечення для вирішення подібних задач, описано їх переваги та недоліки. Обґрунтовано необхідність розробки нового методу підбору тимчасового житла для категорій споживачів на основі генетичного алгоритму.

- Розроблено метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Запропонований метод підбору тимчасового житла враховує усі впливові фактори. Генетичний алгоритм легко застосовувався до нашого завдання кваліфікаційної роботи. Концепція комп'ютерних алгоритмів, заснованих на еволюції організмів, є добре дослідженою та часто застосованою. Багато задач, будь то комерційні, освітні чи наукові, все частіше вирішуються на основі генетичних алгоритмів. Корисність і витонченість генетичного алгоритму у вирішенні проблем зробила його більш улюбленим вибором серед традиційних методів, його можна застосовувати, коли розробник не має точних знань про предметну область, а генетичні алгоритми мають здатність досліджувати і вчитися у своїй предметній області.

- Розроблено веб-додаток, що втілює запропонований метод оптимізації підбору тимчасового житла на мові Python з використанням сучасних фреймворків та бібліотек. Реалізовано процедури реєстрації, візуалізації, оцінки

рівня задоволеності користувача від запропонованого житла. Проведено тестування на адекватність результатів.

– Проведено валідацію розробленого методу на тестових наборах даних. Для перевірки адекватності результатів роботи алгоритму здійснено його порівняння з альтернативними методами. Визначено основні характеристики та проведено дослідження підбору житла в залежності від пріоритетних вимог до помешкання.

– Експериментально підтверджено переваги генетичного алгоритму для даної задачі. Він дозволяє знаходити близький до оптимального розв'язок (житло) за сукупністю критеріїв та з врахуванням додаткових побажань користувачів. Ефективність автоматизованого підбору тимчасового житла для категорій споживачів за розробленим методом на основі генетичного алгоритму збільшується в середньому на 12,5%.

Розроблений метод і програмне забезпечення можуть застосовуватися на практиці для реального підбору тимчасового житла.

Отже, у кваліфікаційній роботі магістра запропоновано і розроблено ефективний метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

Перелік посилань

1. Realtor. [Електронний ресурс] – Режим доступу: <https://www.realtor.com/> (дата звернення: 24.09.2023);
2. Zillow. [Електронний ресурс]. – Режим доступу: <https://www.zillow.com/> (дата звернення: 24.09.2023);
3. Airbnb. [Електронний ресурс]. – Режим доступу: https://www.airbnb.com.ua/?_set_bev_on_new_domain=1699806650_OTUyMjdmMjZlZmMx (дата звернення: 24.09.2023);
4. Booking. [Електронний ресурс]. – Режим доступу: https://www.booking.com/index.uk.html?label=gen173nr-1BCAEoggI46AdIM1gEaOkBiAEBmAEpuAEXyAEM2AEB6AEBiAIBqAIDuALQ-KirBsACAdICJDikNjFmMGE2LTIwYzctNGQ5Yy1hMWU3LWI5YzA0ZGE0Y2Q5MdgCBeACAQ&sid=ded02472ff84608533ba08cb1ee5863d&keep_landing=1&sb_price_type=total& (дата звернення: 24.09.2023);
5. Wohnraum für schutzbedürftige Menschen aus der Ukraine, [Електронний ресурс]. – Режим доступу: <https://www.helfendewaende.de/> (дата звернення: 24.09.2023);
6. Про забезпечення прав і свобод внутрішньо переміщених осіб : Закон України від 20.10.2014 р. № 1706-VII : станом на 3 серп. 2022 р. [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1706-18#Text> (дата звернення: 20.09.2023);
7. Вербицька К. В. Актуальні питання забезпечення потреб вимушених переселенців на території України / Вербицька К. В. Карпенко Р. В. // *Право і суспільство* 2022. № 6. С 124-129;
8. Не можна виселяти ВПО із закладів освіти без надання альтернативного прихистку / Міністерство з питань реінтеграції тимчасово окупованих територій України. [Електронний ресурс]. – Режим доступу: <https://minre.gov.ua/news/ne-mozhna-vyselyaty-vpo-iz-zakladiv-osvity-bez->

nadannya alternatyvnoho

pryhystku?fbclid=IwAR1051c37IFwhMhBtBUo6RH8d5aL8GewIRJnumDjIJ2sZ
9tZgs2d3b-hB68 (дата звернення: 20.09.2023);

9. Тимченко Л. М., Сєдих Ю.О. Правова природа та особливості договору купівлі-продажу земельних ділянок. Молодий вчений. № 11 (99), 2021. С. 323-326. [Електронний ресурс]. – Режим доступу: <https://molodyivchenyi.ua/index.php/journal/article/view/2570> (дата звернення: 20.09.2023);

10. Krystyna Rezvorovych World experience of the land sale market reform : a temporal legal measure. Philosophy, economics and law review. Volume 1, 2021. P. 76-85. [Електронний ресурс]. – Режим доступу: https://phelr.dduvs.in.ua/wp-content/uploads/2021/12/1r/Pherl%20Review_2_1-22.12.2021-6-76-85.pdf (дата звернення: 20.09.2023);

11. K.A.F.A. Samah, Optimization of house purchase recommendation system (HPRS) using genetic algorithm / K.A.F.A. Samah, I.M. Badarudin, E.E. Odzaly, K.N. Ismail, N.I.S. Nasarudin, N.F. Tahar, M.H. Khairuddin // *Indonesian Journal of Electrical Engineering and Computer Science* Vol. 16, No. 3, December 2019, pp. 1530-1538 ISSN: 2502-4752, DOI: 10.11591/ijeecs.v16.i3.pp1530-1538;

12. Alfred, R., Yu, H.F. (2020). Automated Scheduling of Hostel Room Allocation Using Genetic Algorithm. In: Sharma, N., Chakrabarti, A., Balas, V. (eds) *Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing*, vol 1042. Springer, Singapore. https://doi.org/10.1007/978-981-32-9949-8_11;

13. Сортування сусідів по кімнаті за допомогою генетичного алгоритму / Шелдон Реґо, Політехнічна школа [Електронний ресурс]. – Режим доступу: <https://community.wolfram.com/groups/-/m/t/2608922> (дата звернення 05.12.2023);

14. Chih-Ching Chang & Che-Chern Lin / Dormitory Assignment Using a Genetic Algorithm, // *Applied Artificial Intelligence*. 2021 - Issue 15 Volume 35. pp 2276-2297;

15. Applying Genetic Algorithm for Allocating Students to the Dormitory Rooms / Hossein Eslami, Maryam Afzali, Iran Data Mining Conference 2007 [Електронний ресурс]. – Режим доступу: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=90cc2d37700a1071ffaecafb986a18646ac579e2> (дата звернення 05.12.2023);
16. Furnished apartments, homes, temporary housing - rent on wunderflats - wunderflats. [Електронний ресурс]. – Режим доступу: <https://wunderflats.com/en> (дата звернення: 24.09.2023);
17. Feature Engineering and Selection: A Practical Approach for Predictive Models – Розділ 12.3 Genetic Algorithms. [Електронний ресурс] – Режим доступу: <https://bookdown.org/max/FES/genetic-algorithms.html> (date of access: 23.10.2023);
18. Генетичні алгоритми як обчислювальні методи скінченновимірної оптимізації. [Електронний ресурс] – Режим доступу: <http://dspace.nbuiv.gov.ua/handle/123456789/181346> (дата звернення: 12.10.2023);
19. Machine Learning Mastery. Simple Genetic Algorithm From Scratch in Python [Електронний ресурс]. – Режим доступу: <https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/> (дата звернення: 15.10.2023);
20. Реалізація генетичного алгоритму шляхом застосування продукційних правил. Репозиторій :: Репозитарій КНУБА :: Головна. [Електронний ресурс]. – Режим доступу: <https://repository.knuba.edu.ua/items/bc64f68d-3d83-45d2-8cb1-6123d35841b9> (дата звернення: 15.10.2023);
21. AI з python – генетичні алгоритми [Електронний ресурс]. – Режим доступу: <https://coderlessons.com/tutorials/python-technologies/izuchit-iskusstvennyi-intellekt-s-python/ai-s-python-geneticheskie-algoritmy> (дата звернення: 15.10.2023);
22. AlgoDaily [Електронний ресурс]. – Режим доступу: <https://algodaily.com/lessons/introduction-to-genetic-algorithms-in-python> (date of access: 16.10.2023);

23. Genetic algorithm - Wikipedia. [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Genetic_algorithm (date of access: 16.10.2023);

24. ElAr :: Головна. [Электронный ресурс]. – Режим доступа: <https://openarchive.nure.ua/server/api/core/bitstreams/0d035f63-bd86-43d1-8785-78f2c6c77602/content> (дата звернення: 16.10.2023);

25. GAd A. F. Different genetic algorithm representations in python | paperspace blog. Paperspace Blog. [Электронный ресурс]. – Режим доступа: <https://blog.paperspace.com/working-with-different-genetic-algorithm-representations-python/> (date of access: 18.10.2023);

26. Genetic algorithm in python - ander fernández. Ander Fernández. [Электронный ресурс]. – Режим доступа: <https://anderfernandez.com/en/blog/genetic-algorithm-in-python/> (date of access: 18.10.2023);

27. Geneticalgorithm. PyPI. [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/geneticalgorithm/> (date of access: 18.10.2023);

28. Practical genetic algorithms in python and MATLAB - video tutorial - yarpiz. Yarpiz. [Электронный ресурс]. – Режим доступа: <https://yarpiz.com/632/ypГА191215-practical-genetic-algorithms-in-python-and-matlab> (date of access: 23.10.2023);

29. PyGAd - Python Genetic Algorithm! — PyGAd 3.2.0 documentation. PyGAd, [Электронный ресурс]. – Режим доступа: <https://pyGAd.readthedocs.io/en/latest/> (date of access: 23.10.2023);

30. Simple genetic algorithm from scratch in python . [Электронный ресурс]. – Режим доступа: <https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/> (date of access: 23.10.2023);

31. Simple Genetic Algorithm by a Simple Developer (in Python). [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/simple-genetic-algorithm-by-a-simple-developer-in-python-272d58ad3d19> (date of access: 23.10.2023);

32. О. В. Бондаренко, о. В. Устиненко, р. В. Протасов, і. Є. Клочков, б. С. Воронцов, м. В. Матюшенко, п. М. Калінін огляд сучасного використання генетичних та еволюційних алгоритмів. Стратегії, можливості. Вісник національного технічного університету «ХПІ». Серія: машинознавство та сапр. № 2. 2022 с.6-16;

33. Gerges Firas; Zouein Germain, Azar Danielle. Genetic Algorithms with Local Optima Handling to Solve Sudoku Puzzles. Proceedings of the 2018 International Conference on Computing and Artificial Intelligence. ICCAI 2018. New York, NY, USA: Association for Computing Machinery, 2018. P. 19–22.;

34. Musatafa Abbas Albadr, Sabrina Tiun, Masri Ayob, Fahad ALDhief. Genetic algorithm based on natural selection. Theory for Optimization Problems. Symmetry, 2020. No. 12. P. 1–31.;

35. Погорілий С. Д., Білоус Р. В., Білоконь І. В. Застосування генетичних алгоритмів у комп'ютерних системах. Київ: Видавничо-поліграфічний центр «Київський університет», 2014. 319 с.;

36. Silva F. T., Silva M. X., Belchior J. C. A new genetic algorithm approach applied to atomic and molecular cluster studies. Frontiers in Chemistry. 2019. Vol. 7, Article 707. P. 1–21;

37. Fedorchenko I., Oliinyk A., Stepanenko A., Zaiko T., Korniienko S., Burtsev N. Development of a genetic algorithm for placing power supply sources in a distributed electric network. EasternEuropean journal of enterprise technologies. 2019. Vol. 5, No. 3 (101). P. 6–16;

38. Гук Н. А., Гук М. К., Шаповал І. П. Налаштування параметрів генетичного алгоритму для розв'язання задачі оптимізації топології сенсорної мережі. Питання прикладної математики та математичного моделювання. 2018. Вип. 18. С. 57–65;

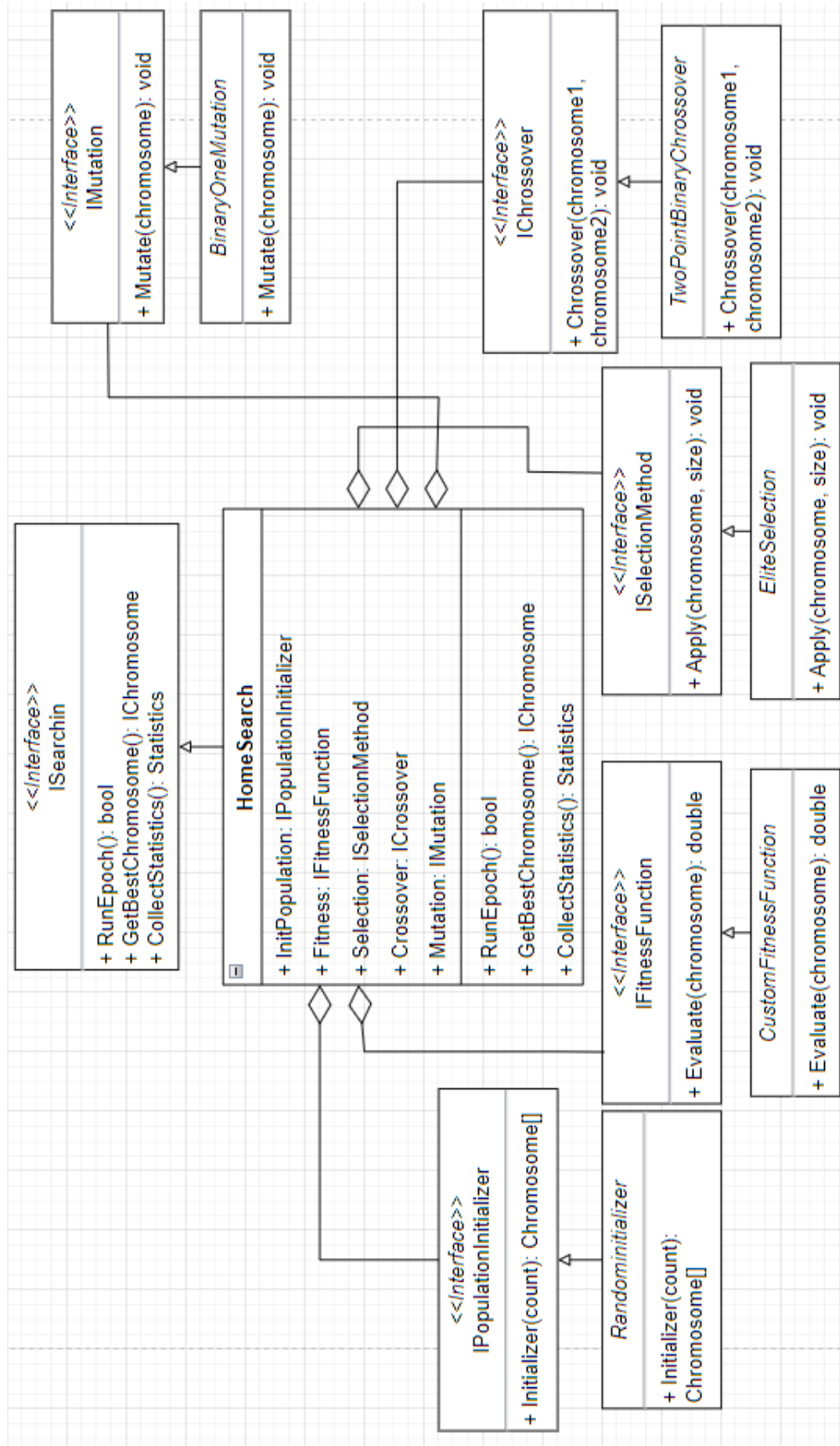
39. Ahiakwo G. N., Braide S. L., Ojuka O. E. Evolutionary algorithm based approach for power quality enhancement. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE). 2022. Vol. 17, Is. 2, Ser. I. P. 36–43;

40. Макаришкін Д. А., Ковтун Л. О., Онишко О. Г., Борис А. В. Еволюційний алгоритм для автоматизованого структурнопараметричного синтезу НВЧ транзисторних підсилювачів. Вісник Хмельницького національного університету. 2016. № 6, С. 238–247;
41. Шаповалова Н. Н., Рибальченко О. Г., Куропятник Д. І. Порівняльний аналіз методів оптимізації функціоналу якості моделей машинного навчання. Вісник Криворізького національного університету. 2018. Вип. 46, С. 104–111;
42. Федорченко Є. М., Олійник А. О., Корнієнко С. К., Харченко А. С., Гончаренко Д. А. Особливості побудови рішень генетичного алгоритму в задачі розпізнавання образів захворювання пневмонії. Реєстрація, зберігання і обробка даних, 2020. Т. 22, № 3. С. 76–95;
43. Налапко О. Л., Козлов В. Г. Метод кластеризації даних на основі еволюційної оптимізації котячих зграй. International scientific conference «New development areas of digitalization at the beginning of the third millennium». RiGA, Latvia: «Baltija Publishing», 2021. P. 21–24;
44. Aponsoa G. C. A. L., Tennakonb T. M. T. I., Arampathc A. M. C. B., Kandeepand S., AmaratunΓAe H. P. K. K. S. Database optimization using genetic algorithms for distributed. DatabasesInternational Journal of Computer. 2017. Vol. 24, No. 1. P. 23–27;
45. Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К. Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». -Хмельницький, 2023. - С. 286-288.

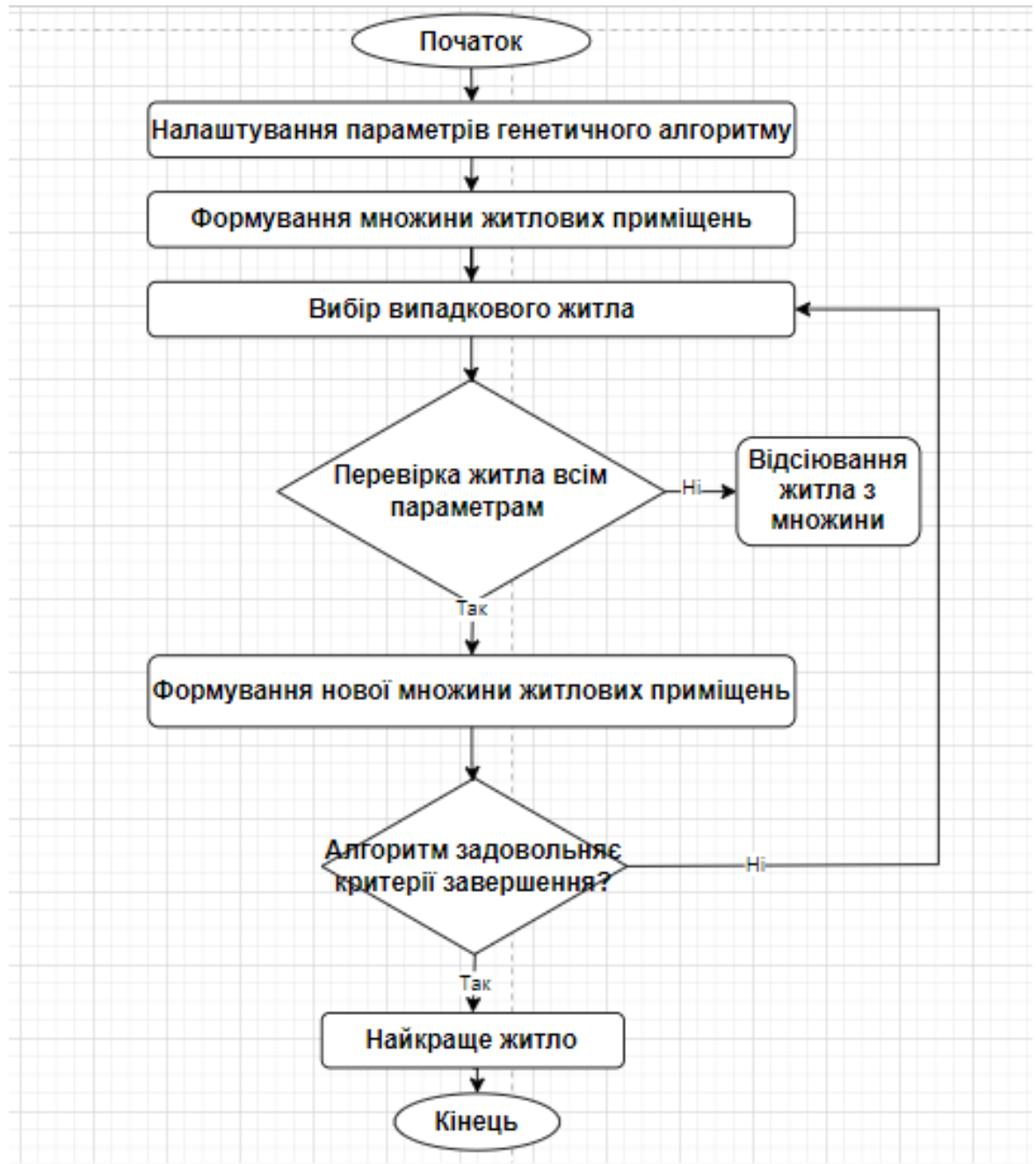
ДОДАТКИ

ДОДАТОК А

Діаграма класів



ДОДАТОК Б
Модель методу



ДОДАТОК В

Програмні коди

```

from numpy.random import randint
from numpy.random import rand
import random

class GenotypeDecoder:
    def decode(self, genotype: str):
        return int(genotype, 2) ** 2

class FitnessEvaluator:
    def __init__(self, genotype_decoder: GenotypeDecoder):
        self.genotype_decoder = genotype_decoder
    def evaluate(self, genotype: str):
        return self.genotype_decoder.decode(genotype)

class Individual:
    def __init__(self, genotype: str, fitness: int):
        self.genotype = genotype
        self.fitness = fitness
    def __repr__(self):
        return "Individual/genotype = " + self.genotype + " Fitness = " +
str(self.fitness)

class IndividualFactory:
    def __init__(self, genotype_length: int, fitness_evaluator:
FitnessEvaluator):
        self.genotype_length = genotype_length
        self.fitness_evaluator = fitness_evaluator
        # E.g. {:032b} to format a number on 32 bits with leading zeros
        self.binary_string_format = '{:0' + str(self.genotype_length) +
'b}'
    def with_random_genotype(self):
        genotype_max_value = 2 ** self.genotype_length
        random_genotype =
self.binary_string_format.format(random.randint(0, genotype_max_value))
        fitness = self.fitness_evaluator.evaluate(random_genotype)
        return Individual(random_genotype, fitness)
    def with_set_genotype(self, genotype: str):
        fitness = self.fitness_evaluator.evaluate(genotype)
        return Individual(genotype, fitness)
    def with_minimal_fitness(self):
        minimal_fitness_genotype = self.binary_string_format.format(0)
        fitness =
self.fitness_evaluator.evaluate(minimal_fitness_genotype)
        return Individual(minimal_fitness_genotype, fitness)

class Population:
    def __init__(self, individuals):
        self.individuals = individuals
    def get_the_fittest(self, n: int):
        self.__sort_by_fitness()
        return self.individuals[:n]

```

```

    def __sort_by_fitness(self):
        self.individuals.sort(key = self.__individual_fitness_sort_key,
reverse = True)
    def __individual_fitness_sort_key(self, individual: Individual):
        return individual.fitness

class PopulationFactory:
    def __init__(self, individual_factory: IndividualFactory):
        self.individual_factory = individual_factory
    def with_random_individuals(self, size: int):
        individuals = []
        for i in range(size):
            individuals.append(self.individual_factory.with_random_genotype())
        return Population(individuals)
    def with_individuals(self, individuals):
        return Population(individuals)
    def with_minimal_fitness_individuals(self, size: int):
        individuals = []
        for i in range(size):
            individuals.append(self.individual_factory.with_minimal_fitness())
        return Population(individuals)

class ParentSelector:
    def select_parents(self, population: Population):
        total_fitness = 0
        fitness_scale = []
        for index, individual in enumerate(population.individuals):
            total_fitness += individual.fitness
            if index == 0:
                fitness_scale.append(individual.fitness)
            else:
                fitness_scale.append(individual.fitness +
fitness_scale[index - 1])
            mating_pool = []
            number_of_parents = len(population.individuals)
            fitness_step = total_fitness / number_of_parents
            random_offset = random.uniform(0, fitness_step)
            current_fitness_pointer = random_offset
            last_fitness_scale_position = 0
            for index in range(len(population.individuals)):
                for fitness_scale_position in
range(last_fitness_scale_position, len(fitness_scale)):
                    if fitness_scale[fitness_scale_position] >=
current_fitness_pointer:
                        mating_pool.append(population.individuals[fitness_scale_position])
                        last_fitness_scale_position = fitness_scale_position
                        break
                current_fitness_pointer += fitness_step
            return mating_pool

class SinglePointCrossover:
    def __init__(self, individual_factory: IndividualFactory):
        self.individual_factory = individual_factory
    def crossover(self, parent_1: Individual, parent_2: Individual):
        crossover_point = random.randint(0, len(parent_1.genotype))
        genotype_1 = self.__new_genotype(crossover_point, parent_1,
parent_2)

```

```

        genotype_2 = self.__new_genotype(crossover_point, parent_2,
parent_1)
        child_1 = self.individual_factory.with_set_genotype(genotype =
genotype_1)
        child_2 = self.individual_factory.with_set_genotype(genotype =
genotype_2)
        return child_1, child_2
    def __new_genotype(self, crossover_point: int, parent_1: Individual,
parent_2: Individual):
        return parent_1.genotype[:crossover_point] +
parent_2.genotype[crossover_point:]

class Mutator:
    def __init__(self, individual_factory: IndividualFactory):
        self.individual_factory = individual_factory
    def mutate(self, individual: Individual):
        mutated_genotype = list(individual.genotype)
        mutation_probability = 1 / len(individual.genotype)
        for index, gene in enumerate(individual.genotype):
            if random.random() < mutation_probability:
                mutated_genotype[index] = '0' if gene == '1' else '1'
        return self.individual_factory.with_set_genotype(genotype =
"".join(mutated_genotype))

class Breeder:
    def __init__(self,
        single_point_crossover: SinglePointCrossover,
        mutator: Mutator):
        self.single_point_crossover = single_point_crossover
        self.mutator = mutator
    def produce_offspring(self, parents):
        offspring = []
        number_of_parents = len(parents)
        for index in range(int(number_of_parents / 2)):
            parent_1, parent_2 = self.__pick_random_parents(parents,
number_of_parents)
            child_1, child_2 =
self.single_point_crossover.crossover(parent_1, parent_2)
            child_1_mutated = mutator.mutate(child_1)
            child_2_mutated = mutator.mutate(child_2)
            offspring.extend((child_1_mutated, child_2_mutated))
        return offspring
    def __pick_random_parents(self, parents, number_of_parents: int):
        parent_1 = parents[random.randint(0, number_of_parents - 1)]
        parent_2 = parents[random.randint(0, number_of_parents - 1)]
        return parent_1, parent_2

class Environment:
    def __init__(self,
        population_size: int,
        parent_selector: ParentSelector,
        population_factory: PopulationFactory,
        breeder: Breeder):
        self.population_factory = population_factory
        self.population =
self.population_factory.with_random_individuals(size = population_size)

```

```

        self.parent_selector = parent_selector
        self.breeder = breeder
    def update(self):
        parents = self.parent_selector.select_parents(self.population)
        next_generation = breeder.produce_offspring(parents)
        self.population =
population_factory.with_individuals(next_generation)
    def get_the_fittest(self, n: int):
        return self.population.get_the_fittest(n)

TOTAL_GENERATIONS = 1000
POPULATION_SIZE = 50
GENOTYPE_LENGTH = 20

current_generation = 1

genotype_decoder = GenotypeDecoder()
fitness_evaluator = FitnessEvaluator(genotype_decoder)
individual_factory = IndividualFactory(GENOTYPE_LENGTH,
fitness_evaluator)
population_factory = PopulationFactory(individual_factory)
single_point_crossover = SinglePointCrossover(individual_factory)
mutator = Mutator(individual_factory)
breeder = Breeder(single_point_crossover, mutator)
parent_selector = ParentSelector()
environment = Environment(POPULATION_SIZE, parent_selector,
population_factory, breeder)

highest_fitness_list = []
while current_generation <= TOTAL_GENERATIONS:
    fittest = environment.get_the_fittest(1)[0]
    highest_fitness_list.append(fittest.fitness)
    if "0" not in fittest.genotype:
        print("Win!")
        break
    environment.update()
    current_generation += 1

print("Зупинилися на множині " + str(current_generation - 1) + ".
Найпристосованіший вибір: ")
print(fittest)

import matplotlib.pyplot as plt

generations = range(1, len(highest_fitness_list) + 1)
plt.plot(generations, highest_fitness_list)
plt.title('Найкраще житло у кожній множині')
plt.xlabel('Множина вибору житла')
plt.ylabel('Відповідність критеріям')
plt.show()

```

ДОДАТОК Г
Світлини наукових публікацій,
виконаних при роботі над кваліфікаційною роботою магістра

Перелік наукових публікацій:

1. Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К. Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». - Хмельницький, 2023. - С. 286 - 288.

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2023»

17-18 листопада 2023

Хмельницький 2023

Слутяк Є.І., Радельчук Г.І., Балицький Б.І. Удосконалення передачі даних у мережі інтернет з використанням алгоритму верифікації повідомлень.....	274
Смірнов О.П., Поплавський С.Ю., Ковальчук В.К., Лутюк Л.І. Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні.....	278
Смолієнко Д.В., Петровський С.С. Метод прогнозування забруднення громадських доріг на основі підходу глибокого активного навчання	280
Собко В.В. Метод поєднання технологій Redux-Toolkit та Redux-Saga для роботи з API веб-ресурсів	284
Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К. Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом	286
Стецюк Ю.В. Методи обробки даних з обмеженим доступом в мультикомп'ютерних системах із застосуванням хмарних технологій їх зберігання	289
Тоцький О.П. Методи обробки кардіограм	293
Уваров В.С., Чабан О.Р., Манзюк Е.А. Метод діагностики захворювань серця на основі аналізу зображень, отриманих методом магнітно-резонансної томографії.....	296
Федоренко В.В., Пасічник О.А., Скрипник Т.К. Технологія блокчейн у сфері реєстрації майнових прав	300
Хміль О.О., Праворська Н.І. Веб-сайт біржі фрілансу	304
Швайко В.К., Ільчишина Ю.В. Метод вибору виду спорту на основі морфофункціональних показників людини	308
Шебетко О.В., Кліменко В.І., Мазурець О.В. Метод адаптивного тестування з використанням продукційних правил.....	311

УДК 004.4

Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К.

*Хмельницький національний університет***МЕТОД АВТОМАТИЗОВАНОГО ПІДБОРУ ТИМЧАСОВОГО ЖИТЛА ДЛЯ КАТЕГОРІЙ СПОЖИВАЧІВ ЗА ГЕНЕТИЧНИМ АЛГОРИТМОМ**

Розглянуто метод автоматизованого підбору тимчасового житла для категорій споживачів. Досліджено суть класифікації методів оптимізації на аналітичні та чисельні методи і показано, що схема генетичного алгоритму може бути виражена у вигляді схеми чисельного методу прямого пошуку. Наведено спосіб зведення заданої оптимізаційної задачі до такої, що може бути розв'язана генетичним алгоритмом, та окреслено клас задач, які можуть бути розв'язані подібним чином. Запропоновано деякі способи підвищення ефективності реалізації цієї процедури.

The method of automated selection of temporary housing for categories of consumers is considered. The essence of the classification of optimization methods into analytical and numerical methods is studied and it is shown that the scheme of the genetic algorithm can be expressed in the form of the scheme of the numerical method of direct search. A method of reducing a given optimization problem to one that can be solved by a genetic algorithm is presented, and a class of problems that can be solved in a similar way is outlined. Some methods of increasing the effectiveness of the implementation of this procedure are proposed.

Розв'язується задача забезпечення автоматизованого пошуку житла для споживачів за критеріями. За основний метод вирішення завдання вибрано метод генетичного алгоритму (ГА).

Для ілюстрації принципів роботи генетичного алгоритму розглядається задача необмеженої оптимізації, в нашому випадку, задача максимізації:

$$\text{Maximize } f(x), x_i^l \leq x_i \leq x_i^u, i = 1, 2, \dots, N, \quad (1)$$

де, x_i^l та x_i^u - нижня та верхня межі, яких може набувати змінна x_i .

Для вирішення проблем з автоматизованим пошуком житла створили на мові Python програмне забезпечення системи пошуку житла для категорії споживачів на основі генетичного алгоритму. Першим кроком будується база даних з усіма користувачами, які мають бажання підібрати собі житло за деякими параметрами. Якщо користувач з таким логіном вже був зареєстрованим у базі даних, то про це повідомляється користувачу.

Наступний крок можливий лише для зареєстрованих осіб. Запускається вікно з полями вводу даних, які користувач повинен заповнити, зокрема, номер телефону, орієнтовну вартість оренди/купівлі житла, район міста чи відстань від міста та інше. Вікно для користувача зображено на Рисунку 1.

Автоматизований підбір житла для категорії споживачів

Введіть, будь-ласка, комфортну для вас відстань(км) від дому до центра міста та ціну(грн) житла!

Номер телефону:
380976299433

Відстань:
5

Вартість:
20000

Пошук

Рисунок 1 – Початкове вікно для вводу даних

Після того як користувач введе номер телефону, відстань та вартість в поля для вводу, його дані будуть збережені в таблицю (Рисунок 2) та відкриється повідомлення про те, що користувача збережено (Рисунок 3):

	A	B	C	D
1	Номер телефону користувача	Відстань	Вартість	
2	380621459356	2	61000	
3	380982234562	1,2	30000	
4				
5				

Рисунок 2 – Збережені споживачі з параметрами житла

Автоматизований підбір житла для категорії споживачів

Введіть, будь-ласка, комфортну для вас відстань(км) від дому до центра міста та ціну(грн) житла!

Номер телефону:
380982234562

Відстань:
1,2

Вартість:
30000

Пошук

Повідомлення

Користувача успішно збережено

OK Скасувати

Рисунок 3 – Повідомлення про успішне збереження даних

Для прикладу візьмемо створену таблицю з переліком вулиць. В таблиці міститься лише дві критерії: вартість та відстань (Рисунок 4).

Опираючись на метод ГА, проводимо пошук оптимального за вказаними критеріями рішення, отримуємо в результаті рядок 6 із таблиці.

	A	B	C	D
1	Назва вулиці	Відстань від центра(км)	Вартість(грн)	
2	вулиця Європейська	2,10	9 000,00 ₴	
3	вулиця 4-а Нова	9,10	10 000,00 ₴	
4	вулиця 1-а Лугова	2,30	12 000,00 ₴	
5	вулиця Академіка Чеботарьова	9,00	12 000,00 ₴	
6	2-й провулок Ентузіастів	2,00	13 000,00 ₴	
7	вулиця Академіка Заболотного	6,30	13 000,00 ₴	
8	вулиця 2-а Комунарська 2-а	7,70	14 000,00 ₴	
9	вулиця Атерлея	8,10	14 000,00 ₴	
10	вулиця 2-а Лугов	8,00	15 000,00 ₴	
11	вулиця 2-а Нова	5,60	22 000,00 ₴	
12	1-й провулок Ентузіастів	1,00	23 000,00 ₴	

Рисунок 4 – Таблиця критеріїв

Отже, запропонований метод на основі генетичного алгоритму довів свою здатність і може бути застосований до автоматизованого підбору тимчасового житла для категорій споживачів.

Перелік посилань

1. Beasley, D., Bull, D. R. and Martin, R. R., «An overview of genetic algorithms: Part 2, research topics», University Computing, Vol. 15, pp. 170–181. citeseer.nj.nec.com/article/beasley93overview.html.
2. Chambers, L., Practical handbook of genetic algorithms: Applications, Vol. I, CRC Press, Boca Raton, Florida.
3. Charbonneau, P., «An introduction to genetic algorithms for numerical optimization». <http://www.hao.ucar.edu/public.research/si/pikaia/tutorial.html>.

ДОДАТОК Д
Презентаційний матеріал

Кваліфікаційна робота магістра

«Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом»

Виконала: студентка 2 курсу, група КНм-22-1
СОБКОВА ЮЛІЯ ВАСИЛІВНА

Керівник: к.ф. -м.н., доцент кафедри КН
МІХАЛЕВСЬКИЙ ВІТАЛІЙ ЦЕЗАРІЙОВИЧ

Мета кваліфікаційної роботи магістра полягає у розробці методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

Об'єкт дослідження – це процес підбору тимчасового житла для категорій споживачів.

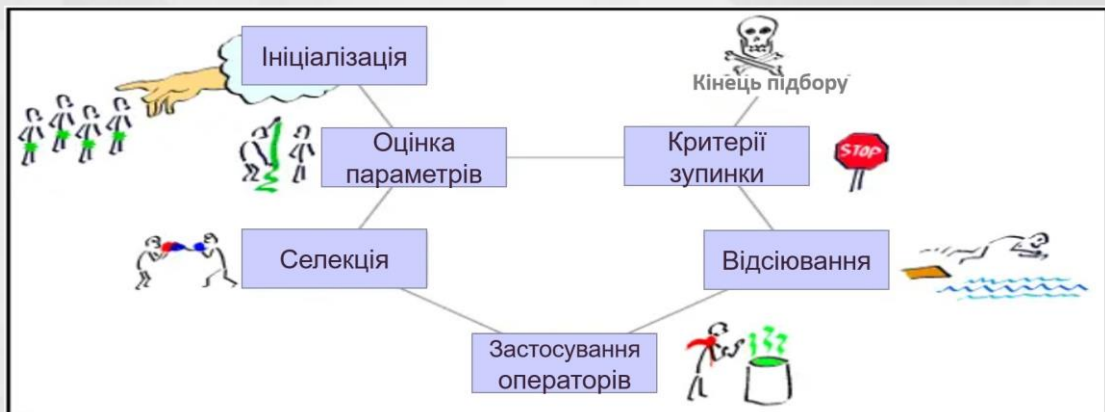
Предмет дослідження – це моделі, методи та засоби автоматизації процесу підбору тимчасового житла для категорій споживачів.



Завдання роботи

- провести аналіз предметної області та відомих підходів до організації підбору тимчасового житла;
- вдосконалити інформаційну модель підбору тимчасового житла для категорій споживачів;
- розробити метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом;
- підготувати набір даних для застосування у генетичному алгоритмі;
- використати генетичний алгоритм для автоматизованого підбору тимчасового житла для категорій споживачів;
- провести функціональне та прикладне дослідження ефективності розробленого методу, в тому числі для тестового аналізу автоматизованого підбору тимчасового житла для категорій споживачів.

Генетичний алгоритм



Модель методу



Вхідні дані

- Строк оренди;
- Графік відключень світла;
- Електрична плита / Газова плита;
- Централізоване опалення;
- Індивідуальне опалення;
- Підвал(приспосований для житла чи ні);
- Заводи;
- Відстань від військової частини;
- Поштове відділення;
- Супермаркети/магазини;
- Квартира за правилом «двох стін»;
- Відстань до зупинки місцевого транспорту;
- Школа/Ліцей;
- ДНЗ;
- Санітарні норми;
- Балкон;
- Наявність інтернету;
- Бойлер;

Особливість методу

Користувач повинен на проти кожного критерію обрати рівень пріоритетності для свого вибору.

Тоді програма буде звертати увагу на скільки відсотків користувачу важливий той чи інший критерій вибору.

Населений пункт:

0.1 Місто

0.6 СМТ

0.3 Село

0.0

0.1

0.2

0.3

0.4

0.5

0.6

0.7

0.8

0.9

Програма автоматизованого підбору житла для категорії споживачів

1

Електронна адреса: poshta@gmail.com
 Номер телефону: 380964042452
 Далі Закрити

2

Категорія:

Пристосоване
 Непристосоване
 Закрити

0.1

0.2

0.3

0.4

0.5

0.6

0.7

0.8

0.9

1.0

4

Розширені параметри

0.6 Відстань від зупинки місцевого транспорту <1 км

0.3 Графік відключення світла

0.8 Можна з тваринами

Опалення: 0.4 Централізоване 0.5 Індивідуальне

Плита: 0.6 Газова 0.2 Електрична

Наванчність:

0.1 Балкон 0.6 Ванна

0.8 Бойлер 0.7 Меблі

0.9 Підвал 0.4 Інтернет

0.2 Душова кабина 0.2 Теплі коври

Поруч:

0.5 Військова частина 1.0 ДНЗ

0.2 Поштове відділення 0.8 Заводи

0.9 Супермаркет 0.6 Лікарня

1.0 Школа/лицей

Почати підбір Назад

3

Солити Example

Параметри житла стандартного набору

Ціна: від 5000 до 15000

Квадратура: від 50 до 70

Поверх: 0.0 1 0.9 2 0.8 3 0.7 4 0.3 5 0.3 6 0.3 7 0.1 8 0.1 9 0.0 10 0.0

Відстань від центру: 1 км 2 км 3 км 4 км 5 км 6 км 7 км 8 км 9 км 10 км

Тип будинку: 0.9 Цегла 0.0 Каркасний 0.0 Сипкатна цегла 0.1 Панель 0.1 Монолит 0.4 Інше

Кількість кімнат: 0.0 1 0.0 2 0.0 3 0.7 4 1.0 5 0.0 >5

Населений пункт: 0.3 Місто 0.5 СМТ 0.1 Село

0.4

0.5

0.6

0.7

0.8

0.9

1.0

На скільки днів плануєте знайти житло? Місяць

Розширені параметри Назад


5

Підбір житла завершився

Адреса: вулиця Лепорська, 10, Львів, Львівська область, 79000

Контактний номер: +38096 404 2452

Детальніше



Наукова новизна

В результаті роботи були отримані наступні результати:

- вдосконалено інформаційну модель підбору тимчасового житла для категорій споживачів, яка відрізняється тим, що містить формальне подання всіх необхідних сутностей для операцій підбору тимчасового житла для категорій споживачів;
- розроблено метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом, що дозволяє за множиною наявних параметрів житла, початкових та проміжних вимог до житла, апаратно-технічних засобів автоматично визначати множину можливих варіантів тимчасового житла та визначити оптимальний, що збільшує ефективність підбору тимчасового житла і забезпечує рівень задоволення від отриманого результату в середньому на 12,5%.

Функціональне дослідження та визначення ефективності методу

Таблиця 1 – Параметри житла

Параметр	Оцінка параметру	Відсотки
Ціна	13 000 грн.	25%
Квадратура	70 м ²	25%
Кількість кімнат	4	25%
Населений пункт	СМТ	25%

Таблиця 2 – Рівень відповідності 75%.

Параметр	Оцінка параметру	Відсотки
Ціна	13 000 грн.	25%
Квадратура	60 м ²	25%
Кількість кімнат	4	25%
Населений пункт	СМТ	25%

Таблиця 3 – Рівень відповідності 50%.

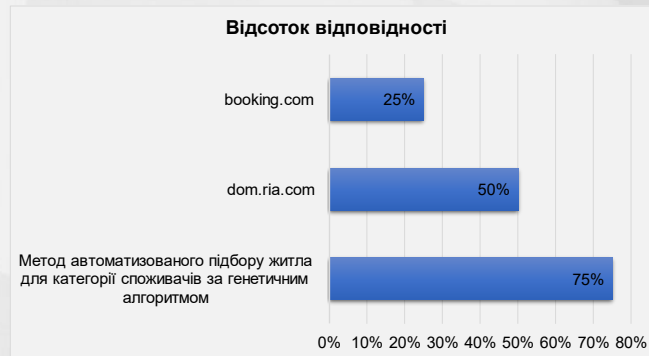
Параметр	Оцінка параметру	Відсотки
Ціна	9 500 грн.	25%
Квадратура	64 м ²	25%
Кількість кімнат	4	25%
Населений пункт	Місто	25%

Таблиця 4 – Рівень відповідності 25%.

Параметр	Оцінка параметру	Відсотки
Ціна	19 000 грн.	25%
Квадратура	40 м ²	25%
Кількість кімнат	1	25%
Населений пункт	СМТ	25%

Таблиця 5 – Порівняльна таблиця рівня відповідності параметрів.

Система підбору житла	Відсоток відповідності
Метод автоматизованого підбору житла для категорії споживачів за генетичним алгоритмом	75%
dom.ria.com	50%
booking.com	25%



Висновки

Результатом виконання кваліфікаційної роботи магістра є розроблений метод автоматизованого підбору тимчасового житла для категорії споживачів за генетичним алгоритмом.

- Проведено дослідження предметної області підбору житла для категорій споживачів.
- Досліджено існуючі методи підбору житла та програмне забезпечення для вирішення подібних задач, описано їх переваги та недоліки.
- Експериментально підтверджено переваги генетичного алгоритму для даної задачі.

Розроблений метод і програмне забезпечення можуть застосовуватися на практиці для реального підбору тимчасового житла.



Дякую за увагу!



Ім'я користувача:
Кафедра КН

ID перевірки:
1015996634

Дата перевірки:
12.12.2023 11:04:30 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.12.2023 11:07:01 EET

ID користувача:
100005671

Назва документа: КНн-22-1 Собкова

Кількість сторінок: 96 Кількість слів: 17637 Кількість символів: 134035 Розмір файлу: 2.63 MB ID файлу: 1015679475

14.6% Схожість

Найбільша схожість: 2.7% з Інтернет-джерелом (http://pravoisuspistvo.org.ua/archive/2022/6_2022/6_2022.pdf)

13.1% Джерела з Інтернету 899 Сторінка 98

1.67% Джерела з Бібліотеки 98 Сторінка 105

0% Цитат

Вилучення цитат виключено

Вилучення списку бібліографічних посилань виключено

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 26

Tue Dec 12 11:25:59 EET 2023, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 6%**

ID: 122701 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА Тема: Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом Додано в БД: 2023-12-12 Автора: Ю.В. Собюва Керівники: В.Ц. Міхалевський Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	105125	1658	4063 (4%)	59 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

ВРІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА
РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

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

Назва: Метод автоматизованого підбору житла для категорії споживачів за генетичним алгоритмом

Автор: Собкова Ю.В., група КНм-22-1

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: комп'ютерні науки

Науковий керівник: к.ф.-м.н., доц. кафедри КН Міхалевський 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, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 2% запозичень, що підтверджує авторство дослідження.

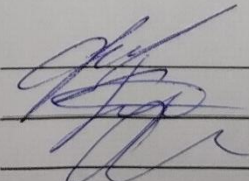
2) За програмою UNICHECK виявлені 14,6% запозичень є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 2% і 14,6% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН



В. Ц. Міхалевський

Р. О. Багрій

О. В. Бармак



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

групи КНМ-22-1 Собкової Юлії Василівни за темою: «Метод автоматизованого підбору житла для категорій споживачів за генетичним алгоритмом»

1. Актуальність обраної теми.

Сьогодні онлайн-платформи полегшують обмін інформацією про ринок житла та розширюють читабельність ринку житла для продавців, покупців, орендодавців та орендарів. Такі платформи можуть демократизувати доступ до інформації та урізноманітнити інформаційне забезпечення шукачів житла. Це, у свою чергу, може розширити параметри вибору, збільшити радіус підбору, зменшити витрати на пошук і відійти від традиційних підходів, щоб допомогти шукачам житла реалізувати ефективніший підбір з кращим результатом. У цій кваліфікаційній роботі розробляється та досліджується метод підбору житла для категорій користувачів.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт.

Обрана тема розробки методу автоматизованого підбору житла для категорій споживачів за генетичним алгоритмом та створення відповідного програмного забезпечення повною мірою відповідає предметній області спеціальності 122 – Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра, оскільки охоплює питання моделювання складних систем, проектування та реалізації алгоритмів і програм, проведення обчислювальних експериментів. Розроблений в роботі метод та програмні засоби дозволяють автоматизувати та оптимізувати процес підбору житла для категорій споживачів.

3. Повнота розкриття мети та завдань дослідження.

В кваліфікаційній роботі автор повністю розкриває мету дослідження та поставленні в межах теми завдання. Результати дослідження ясно визначають основну мету. Розроблено метод для автоматизації підбору житла для категорій споживачів за генетичним алгоритмом. Даний метод дозволяє отримувати різні множини житла з урахуванням наборів параметрів та формувати оптимальний варіант житла.

4. Наявність наукової новизни.

В кваліфікаційній роботі представлена наукова новизна, що відповідає спеціальності 122 «Комп'ютерні науки» в межах обраної області дослідження. Продемонстровано й обгрунтовано результати, які мають наукове та інноваційне значення.

Основні наукові й практичні результати кваліфікаційної роботи магістра доповідались на XV Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2023» та у Збірнику наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023».

5. Зміст кожного розділу роботи.

Робота містить чотири розділи. У першому розділі провели аналіз предметної області, моделей, існуючим наукових публікацій та рішень. Сформували постановку задачі. В другому розділі розробили інформаційну модель методу автоматизованого підбору житла для категорій споживачів за генетичним алгоритмом. В третьому розділі сформували програмну реалізацію методу автоматизованого підбору, обґрунтували вибір середовища розробки методу та висвітлили вимоги до розробки програмного забезпечення. Четвертий розділ присвячений функціональному дослідженню та визначенню ефективності методу підбору тимчасового житла для категорій споживачів за генетичним алгоритмом.

6. Ступінь розкриття теми роботи.

Тема кваліфікаційної роботи повною мірою розкрита та обґрунтована, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та проведено аналіз результатів прикладного застосування запропонованих методу і засобів.

7. Якість оформлення кваліфікаційної роботи

Оформлення роботи відповідає необхідним нормам та вимогам, які ставляться до оформлення кваліфікаційних робіт.

8. Недоліки кваліфікаційної роботи

Явних недоліків в роботі не виявлено. Доцільно було б навести приклади тестування запропонованого методу на результатах підбору житла в нестандартних умовах або з врахуванням національно-культурних особливостей споживачів.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота

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

Опонент _____ д.фіз.-мат.н., проф. Бедратюк Л.П.



**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ**



кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНм-22-1 Собкової Юлії Василівни за темою: Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом

1. Актуальність теми

Актуальність теми обґрунтована в достатній мірі: на сьогоднішній день існує ряд рішень, що допомагають вирішувати задачу підбору тимчасового житла, проте вони мають ряд недоліків. Враховуючи те, що апаратно-технічне забезпечення постійно вдосконалюється і, крім того, розвивається парк програмного забезпечення, то розробка нових методів і засобів підбору тимчасового житла для категорій споживачів є актуальною.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

В кваліфікаційній роботі магістра було розроблено метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом. Детально досліджені процеси підбору тимчасового житла, методи отримання рішень щодо вибору оптимального житла та проведено аналіз ряду програмних продуктів для підбору тимчасового житла для категорій споживачів.

Тема кваліфікаційної роботи магістра відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра: 1) Виконавцем сплановано і реалізовано метод автоматизованого підбору житла для категорій споживачів за генетичним алгоритмом та програмного забезпечення, проведено тестування та порівняльний аналіз. 2) Виконавець обрав інформаційне середовище розробки та дослідження, що дозволило знайти правильне і ефективне рішення, а запропонований метод на основі генетичного алгоритму дозволяє обрати найоптимальніший варіант житла для категорій споживачів.

Проаналізовано, оцінено та порівняно різні технології процесів для встановлення пріоритетів у відповідності з критеріями продуктивності та якості, що визначені завданням. Проведено функціональне та прикладне дослідження ефективності розроблених засобів, у тому числі для тестового аналізу випадків підбору тимчасового житла для категорій споживачів.

3. Професійні та особистісні якості магістранта

Магістрант володіє в достатній мірі професійними якостями дослідника: 1) Має здатність збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до методу, що розробляється. 2) Має здатність формалізувати предметну область проекту у вигляді відповідної інформаційної моделі.

Серед особистісних якостей магістранта слід виділити відповідальність, цілеспрямованість, здатність навчатися, нестандартність мислення.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Студент більшу частину роботи виконав самостійно. Особисто магістрантом досліджено предметну область, проведено порівняльний аналіз переваг та недоліків існуючих методів для підбору житла; підібрано алгоритм для опрацювання даних; розроблено нову інформаційну систему автоматизованого підбору тимчасового житла для

категорій споживачів, проведено функціональне та прикладне дослідження ефективності розробленого методу, у тому числі для тестового аналізу випадків підбору житла.

5. Наукова новизна та оригінальність запропонованих підходів

У виконаній роботі наукова новизна присутня в достатній мірі. Інноваційний підхід проявлено в розробці методу автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом, що дозволяє за множиною наявних параметрів автоматизовано визначати множину житла для здійснення підбору.

Результати дослідження доповідались на 1-й конференції та оприлюднені в 1-х тезах.

6. Ступінь оволодіння методами дослідження

Магістрант в достатній мірі оволоділа методами дослідження, які були використані у роботі: порівняння, аналізу, класифікації, узагальнення.

7. Повнота та якість розкриття теми роботи

Тема роботи розкрита достатньо в рамках поставлених завдань: спроектовано структуру і розроблено метод та прикладне програмне забезпечення автоматизованого підбору житла для категорій споживачів за множиною житла та підбору оптимального варіанту. Інформаційна система включає в себе базу даних, базу знань і п'ять функціональних модулів, що мають різне призначення: модуль роботи користувача з експертними даними, модуль роботи користувача з каталоговими даними, модуль взаємодії користувача з оперативними даними, модуль формування множини можливих маршрутів та модуль оптимізації за критеріями часу, відстані та ваги замовлення і виведення користувачу результатів роботи.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Магістрант матеріал виклав логічно, послідовно, аргументовано. Наводилися наявні розробки, ставилося завдання та послідовно розв'язувалося. Для аргументації отриманих рішень проводилося теоретичне обґрунтування та порівняльний аналіз експериментів.

Літературна та граматична якість матеріалу на достатньому рівні.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

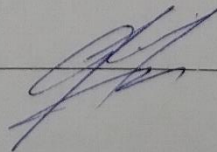
Результати кваліфікаційної роботи магістра можуть застосовуватися на практиці після налаштувань під конкретно запропоновану модель проведення підбору тимчасового житла для категорій споживачів. Практична цінність роботи полягає в тому, що при застосуванні методу оптимального підбору житла виконується не тільки відкидання параметрично неприйнятних варіантів, а й запити користувачу на встановлення параметрів житла, що дозволяє в процесі роботи методу розширити функціональну картину та взяти до розгляду відпочатку безпідставні варіанти.

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

Рекомендую допустити кваліфікаційну роботу магістра до захисту.

Робота заслуговує на оцінку « добре ».

Науковий керівник _____



к.фіз.-мат.н., доц. Віталій Міхалевський