

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»



Факультет комп'ютерних наук та технологій
Кафедра «Комп'ютерні системи та мережі»

СОЛОГУБОВ ІЛЛЯ ЯРОСЛАВОВИЧ
Група КНТ-612м

КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ВИРІШЕННЯ І
ОПТИМІЗАЦІЇ ЗАДАЧ ЛОГІСТИКИ З
ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

АВТОРЕФЕРАТ

магістерської роботи на здобуття освітньо-кваліфікаційного
рівня «магістр» 123 «Комп'ютерна інженерія»
освітньої програми «Комп'ютерні системи та мережі»

2023 р.

Магістерська робота є рукопис.

Робота виконана в Національному університеті «Запорізька політехніка», на кафедрі комп'ютерних систем та мереж

Керівник кандидат технічних наук, доцент
Скрупський Степан Юрійович,
Національний університет «Запорізька
політехніка», доцент кафедри
комп'ютерних систем та мереж

**Офіційний
рецензент:** **Каплієнко Тетяна Ігорівна**,
доцента кафедри «Програмні засоби» НУ
«Запорізька політехніка»

Захист відбудеться "13" грудня 2023 р.

Секретар екзаменаційної комісії, старший викладач
кафедри комп'ютерних систем та мереж **Н.А. Куликовська**

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Весь світ будується на правильному використанні наявних ресурсів, вони можуть бути різними – товари масового збуту, сировина, їжа, зброя, енергоресурси, інформація та ін. Наявність ресурсів є лише одним із основних питань які ставить перед собою держава чи якесь підприємство, не менш важливим є питання доставки цих ресурсів до місць їх обробки, створення оптимальних ланцюгів виробництва.

Зі зростанням обсягів виробництва, глобалізацією та підвищенням транспортних можливостей, особливо актуальним стає питання знаходження оптимальних шляхів постачання ресурсів. Дана задача вирішується людьми ще з часів зародження цивілізації, однак чітко сформулювати її стало можливим після відкриття розділу математики – теорії графів і створення на її основі найвідомішої задачі – задача комівояжера (travelling salesman problem або TSP). Метою даної задачі є знаходження оптимального маршруту для всіх місць призначення з умовою їх досягнення хоча б один раз.

На поточний момент не існує чіткого алгоритму який би міг вирішити дану задачу за оптимальний час. Існує лише декілька різних варіантів оптимізації задачі з використанням різних підходів таких як обмеження вибірки рішень, зберігання часткових рішень чи перетворення на графах що можуть надати наближене рішення, який полегшує пошук. Однак всі ці алгоритми та підходи за своєю суттю представляють собою перебір.

Сучасними методами вирішення подібних задач є рішення з використанням методів штучного інтелекту, таких як – еволюційний, ройовий та інші. Еволюційний підхід представляється у вигляді генетичного алгоритму, яке імітує еволюційну поведінку всіх живих істот в природі з метою досягнення кращого результату. Ройовий представляється у вигляді мурашиного, який імітує поведінку живих мурах з метою знаходження оптимального шляху.

Особливість даних методів є те, що вони не є чіткими алгоритмами, а вказують на кроки які мають бути виконані, тому вони можуть мати абсолютно різні реалізації. Існує багато загальноприйнятих практик і підходів щодо їх реалізації чи

окремих кроків. Через що виникає багато питань, щодо конкретної роботи алгоритмів. Існує багато універсальних реалізацій з використанням базових алгоритмів для кожного етапу, наприклад для генетичних існує декілька готових алгоритмів схрещування, мутації, селекції та інших етапів. Зважаючи на це можна стверджувати що дані алгоритми мають значний потенціал для розвитку і їх оптимізації, через наявність такої гнучкості.

В роботі було сконцентровано увагу саме на генетичних алгоритмах. Проведений аналіз існуючих алгоритмів та підходів для пошуку оптимального шляху, що загальноприйняті реалізації не враховують особливостей конкретної предметної області, які потенційно можна використати для оптимізації алгоритму. Це пояснюється тим, що це може створити складнощі у майбутньому налаштуванні алгоритму і навчанні. Однак в свою чергу це сильно обмежує алгоритм, що в свою чергу веде до втрати продуктивності.

Таким чином розробка і створення нових підходів чи реалізації для таких алгоритмів як генетичний є актуальною і важливою. В свою чергу використання конкретизованого підходу і модифікації алгоритму за ним, окрім його безпосереднього використання для оптимізації задача, може також надавати нові ідеї і поштовх для майбутніх досліджень.

Мета і завдання дослідження. Мета магістерської роботи полягає у розробці та дослідженні комп'ютерної системи, яка виконувала б пошук оптимального шляху для доставки замовлень, використовуючи методи штучного інтелекту, а також автоматизацію обробки самих замовлень.

Для досягнення поставленої мети необхідно виконати такі основні завдання:

1. Провести аналіз вимог системи та аналіз наявних методів та алгоритмів пошуку оптимального шляху між точками. Виділити основні можливості алгоритмів та потенційно корисні дані які вони можуть надати;

2. Розробити структуру та алгоритм роботи системи відповідно до результатів першого пункту;

3. Реалізувати та протестувати систему, аби переконатися в її працездатності;

4. Провести аналіз розробленої системи, та дослідити ефективність алгоритму, за розробленим підходом.

Об'єктом дослідження – алгоритми пошуку оптимального шляху з використанням еволюційного методу та інших алгоритмів на графах.

Предмет дослідження – комп'ютерна система для автоматизації доставки товарів та модифікований генетичний алгоритм з використанням конкретизованого підходу.

Методи дослідження базуються на електронних вимірах основних характеристик системи, їх співставленні зі звичайними алгоритмами. Достовірність результатів досліджень забезпечується коректністю принципів параметрів, постановки задачі, обсягом вибірки для перевірки.

Наукова новизна отриманих результатів:

Запропоновано підхід пошуку оптимального шляху, який використовує комбінацію генетичного алгоритму та алгоритмів пошуку мінімального остовного дерева і обходів на графах, що надає можливості для оптимізації і калібрування алгоритму.

Практичне значення отриманих результатів:

Розроблена комп'ютерна система, яка реалізує запропонований підхід у генетичному алгоритмі та дозволяє пришвидшити виконання пошуку оптимального шляху майже на 30% у порівнянні з іншими загальноприйнятими підходами, з відносно мінімальною похибкою.

Структура та обсяг роботи. Магістерська робота складається зі вступу, п'яти розділів, висновків, списку використаних джерел, додатку. Основна частина містить 89 сторінок, 25 рисунків і 2 таблиць, список використаних джерел зі 51 найменування.

Апробація результатів: основні положення магістерської та результати дослідження подано до участі у конференції (Поточний статус - на стадії розгляду):

- 28th Conference on Automation - Innovations and Future Perspectives «automation 2024» у тематиці «Automation» - Łukasiewicz Research Network – Industrial Research Institute for Automation and Measurements PIAP – (м. Warsaw, Poland, May 8-10, 2024.).

ОСНОВНИЙ ЗМІСТ РОБОТИ

У першому розділі проведено аналіз існуючих алгоритмів та методів для пошуку оптимального шляху між вершинами, та підходів до оптимізації даної задачі. За результатами було визначено алгоритми та основні ідеї в оптимізації, які використовуються для вирішення задачі.

У результаті аналізу для розробки основного алгоритму було обрано еволюційний метод, у вигляді генетичного алгоритму. Визначено додаткові особливості інших алгоритмів і можливостей які вони надають для оптимізації алгоритму, впровадження яких потенційно могли сприяти підвищенню ефективності роботи алгоритму.

Також визначено основні вимоги до предметної області для розроблюваної системи.

У другому розділі проаналізовано сучасні підходи для створення архітектури системи, інструменти для розробки системи. В результаті аналізу були обрані необхідні інструменти та підходи, з якими було реалізовано систему. Схема наведена на рисунку 1.

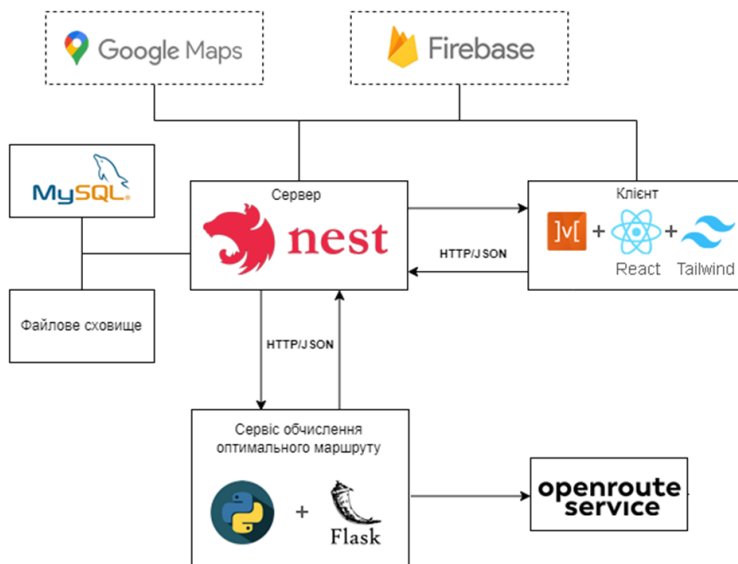


Рисунок 1 – Схема реалізації системи

Дана система представляє собою розподілену веб системою, яка в основі представляє собою моноліт, хоча в ній так само наявний додатковий малий сервіс для обчислення оптимального шляху. На рисунку 1 продемонстровані всі інструменти які були використані для реалізації кожного компоненту системи, а також зовнішні сервіси які надавали змогу полегшити реалізацію таких елементів як авторизації (Firebase) та роботою з мапами і отримання даних для обчислення (Google Maps, Openroute Service).

На рисунку 2 наведено опис основного алгоритму системи, який виконує автоматично обробку замовлень на основі існуючих даних. Ця система автоматично приймає рішення щодо виконання замовлень, групування їх у один маршрут для кур'єра з побудовою самого оптимального маршруту. Адміністратор системи лише має вести контроль і за необхідності реєструвати нові дані, такі як працівники та нові відділення, та редагування усіх інших даних за необхідністю, замовлення, товару та попередньо описаних.

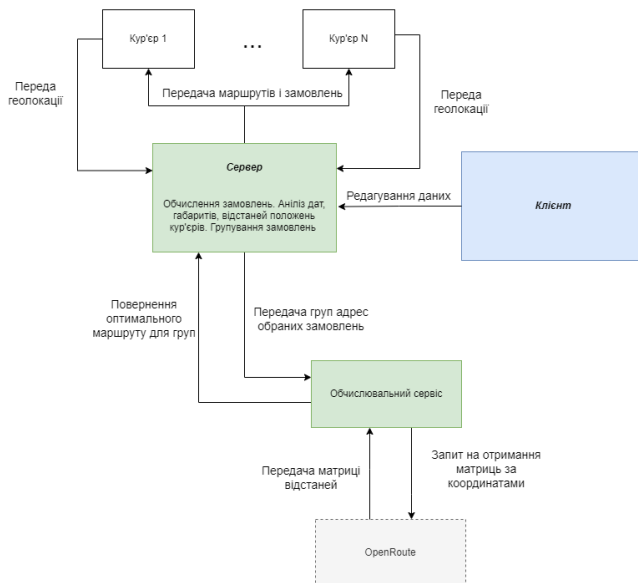


Рисунок 2 – Схема алгоритму обробки замовлень в системі

У третьому розділі описано розробку підходу до для використання генетичних алгоритмів з використанням додаткових алгоритмів на графах. Надано опис ідеї, реалізації модифікованих етапів та проаналізовано ефективність виконання модифікованого алгоритму у порівнянні з іншими реалізаціями, які базуються на загальноприйнятих підходах.

Першим алгоритмом і ідеєю оптимізації стало використання мінімального остовного дерева. Мінімальним можливим рішенням задачі Комівоєжера по довжині шляху - буде шлях який пройде по $N-1$ найкоротшим ребрам, де N кількість вершин. Мінімальне остовне дерево - це граф який об'єднує всі вершини графа без циклів і довжина всіх ребер мінімальна і їх кількість $N-1$. Отже в кращому випадку дане дерево і буде оптимальним рішенням, а в гіршому якщо спробувати обійти граф по всім його точкам - довжина шляху буде максимум в двічі більше, оскільки кожне ребро в гіршому разі доведеться пройти двічі. Дане твердження працює для будь-якого остовного дерева. Завдяки цим даним стають зрозумілі межі пошуку рішень від найкоротшого до найдовшого, які можна використати в генетичному алгоритмі.

В контексті генетичного алгоритму, можна модифікувати за допомогою даного критерію такі кроки як – мутацію та вибірку наступного покоління. В поточній системі вони обидва були модифіковані. Ідея вибірки кращого покоління для його розмноження полягає у видаленні з колонії заздалегідь відоме неоптимальне рішення, з поступовим з наближенням верхньої межі до нижньої.

Також вводиться спеціальний лічильник. У випадку якщо не вдалося зсунути межу з поточною генерацією - збільшується лічильник. У разі свого переповнення він надає сигнал що колонія більше не здатна отримати кращу величину і потенційно можна вважати що її еволюція дійшла до піку, що є кінцем виконання алгоритму.

Алгоритм мутації так само зав'язаний на цьому лічильнику. Чим ближче стає лічильник до свого переповнення тим більше стає вірогідність мутації. Ідея полягає у тому, що у випадку коли колонія не здатна дати нове, то мутація дозволяла б додати в

колонію новий генетичний матеріал і могла побороти проблему нестачі генетичного матеріалу.

Наступною оптимізацією алгоритму є створення початкової оптимальної колонії, чії значення входять у межі і потенційно є найкращими. Ідея полягає у трансформації знайденого мінімального дерева у шлях який обходить всі вершини.

На рисунку 3 проілюстрована побудова оптимального шляху використовуючи мінімальне остовне дерево та використовуючи обхід у глибину. Обхід графа у глибину є одним з найпростіших алгоритмів обходу, але він дозволяє обходити дерево поступово не полишаючи гілку до тих пір доки вона не буде повністю відвідана.

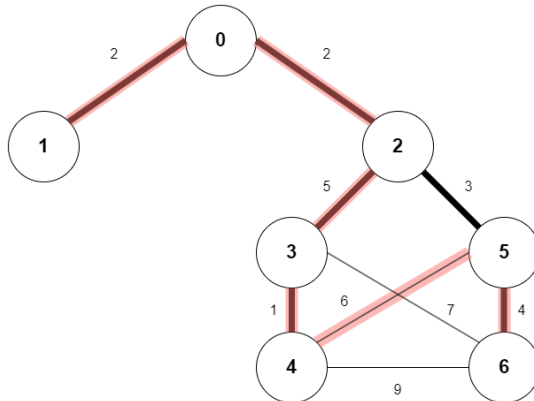


Рисунок 3 – Здобуття початкового шляху алгоритмом обходу із мінімального остовного дерева

Основна цікавість і користь даного шляху полягає в тому, що воно не тільки є потенційно оптимальним і може бути використано як індивід у колонії, а й у тому, що даний шлях може бути використаний як основа, для побудови інших індивідів в популяції.

Ідея побудови першої колонії з такої основи полягає, полягає в наступному спостереженні. На рисунку 4 продемонстровано початковий шлях який можна отримати з мінімального остовного дерева. З якого можна побачити, що деякі підграфи (сукупність точок між якими прокладено шлях) є вже готовими оптимальними шляхами. Очевидними проблемами є наявність петлі у шляху, та

недоцільний розворот, які очевидно, не можуть існувати в остаточному рішенні. Виходячи з цього, доцільним може бути генерації першої колонії не з випадкової послідовності точок, а з шматків різною довжини поточного графу, оскільки перемішування за шматками може залишити оптимальні шматки і лише по різному побудувати між ними так звані мости. Міст у графі – це ребро, яке поєднує два підграфи і його відсутність розриває сам граф.

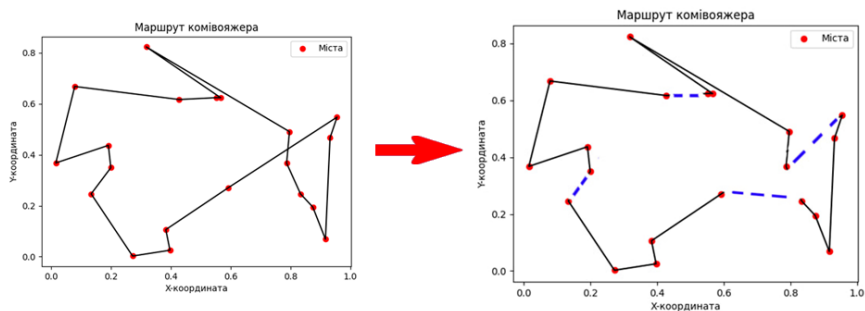


Рисунок 4 - Здобуття шляху обходу графу із 20 вершин і потенційна генерація нового індивіда перемішуючи підграфи

Можна побачити що використовуючи подібні маніпуляції – можна вирішити деякі неоптимальні моменти у графі і створити багато навіть більш оптимальних рішень. При такому перемішуванні також доцільно ще проводити певні маніпуляції з початковим шляхом. Наприклад доцільно проводити циклічний зсув шляху, аби постійно брати різні шматки та випадковим чином розвертати шматки, щоб генерувати більше різних варіантів.

Алгоритм селекції також модифікується за допомогою такої генерації. Очевидним стає той момент що подібна генерація може створити проблему, яка в біології відома як нестача генетичного різноманіття. Обмеженість генів може не дати можливість знайти достатньо оптимальне рішення. Тому пропонується модифікувати селекцію таким чином щоб у випадку, коли залишається мало індивідів після зсуву і відсіювання за допомогою меж, в колонію додавалися додаткові індивіди які генеруються за алгоритмом генерації вже з нового найкращого рішення.

Для аналізу і тестування був побудований алгоритм використовуючи подібний підхід. Алгоритми модифікованих кроків відповідають попередньому опису, кросовер - одноточковий, схрещування через попарне схрещування сусідніх індивідів які відсортовані за зростанням довжини шляху.

Щодо конкретних параметрів для мутації визначено коефіцієнт 0.1 як базовий шанс мутації який ріс із лічильником до 0.2. Межа лічильнику 20% від максимуму генерацій які виконує алгоритм. Шматки перемішування для генерації надавалися масивом $[2^0, 2^1, 2^2 \dots 2^k, N/4]$, де N – кількість вершин і $2^k < N/4$.

Було проведено два дослідження – порівняння алгоритмі з використанням основи у вигляді оптимального маршруту і без, та порівняння з іншими генетичними алгоритмами.

В таблиці 3.1 результати у часі і відстані для першого дослідження. Наведено параметр дельта який визначає на скільки сильно відрізняється рішення двох рішень. Позитивне значення – кращий результат, від'ємне – гірший. Який розраховувався як відношення алгоритму з основою до алгоритму без.

Таблиця 1 – Результати тестування алгоритму з основою та без

№	Алгоритм з осн.		Алгоритм без осн.		Δ (відс.)	Δ (час)
	Відст.	Час(сек)	Відст.	Час(сек)		
1	3.985	8.592	4.041	10.595	0.014	0.189
2	3.592	6.129	3.592	9.666	0	0.366
3	4.169	8.636	4.169	11.103	0	0.222
4	3.753	7.627	3.769	10.849	0.004	0.297
5	4.546	7.291	4.429	11.486	-0.026	0.365
6	3.617	10.345	3.798	13.586	0.048	0.239
7	3.316	9.337	3.305	11.837	-0.003	0.211
8	3.697	7.061	3.682	9.527	-0.004	0.259
9	4.074	6.811	4.101	8.235	0.007	0.173
11	3.925	5.367	3.925	9.502	0	0.435
12	3.557	9.505	3.486	9.563	-0.021	0.006
13	3.486	9.563	4.587	9.969	0.240	0.041
14	3.592	9.969	3.591	10.268	0	0.029
15	4.327	9.573	4.411	5.246	0.019	-0.825

За даними з таблиці можна побачити що алгоритм без основи працює в середньому повільніше, а в деяких випадках стільки ж часу скільки й алгоритм з основою. Обидва алгоритми видають значення що відрізняють один від одного не більше ніж на 5%, при чому алгоритм з основою частіше знаходить більш оптимальне значення. Однак слід зазначити що іноді відіграє роль фактор випадковості значень і алгоритм без використання основи отримує випадковим чином краще рішення ще на етапі генерації значень. Таким чином іноді стаються випадки коли він спрацьовує навпаки швидше чи отримує більш оптимальне значення. За експериментальною оцінкою таких випадків не більше 10%.

На рисунку 5 наведено графік залежності часу роботи генетичних алгоритмів від кількості вершин. Для тестування був обраний стандартний алгоритм який використовує фітнес функцію як критерію для проведення мутацій та відбору батьків з критерієм зупинки через перевірку наявності зміни популяції. Даний алгоритм є стандартним і універсальним для будь якої задачі. В якості іншого алгоритму були переглянуті найкращі реалізації з готовими коефіцієнтами

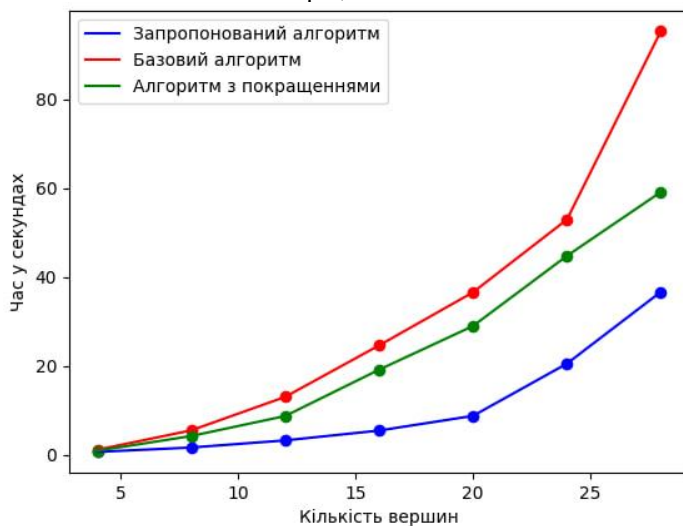


Рисунок 5 – Залежність часу роботи від кількості вершин в різних алгоритмах

Графіки зображені зверху вниз – базовий алгоритм, з покращеннями та запропонований відповідно. З графіку можна побачити що алгоритм з основою для вирішення задачі комівояжера працює так само швидше. Однак слід зазначити, що зі зростанням кількості вершин – зростає і вірогідність помилки в розрахунку алгоритмом з основою. Алгоритм з покращеннями видає більш часто оптимальніший варіант маршруту на великій кількості вершин. В гірших випадках алгоритм з основою починає мати похибку приблизно до 4%, що з великою вірогідністю пояснюється недостатньою навченістю алгоритму.

У таблиці 2 наведено ці результати тестування за часом та точністю відповіді. Точність визначається параметром дельта, який визначає у скільки разів рішення гірше відносно реального рішення. В якості дослідження параметру дельта було взято одні з найгірших показників, для визначення потенційної помилки.

Таблиця 2 – Результати роботи трьох алгоритмів

N (верш.)	Час виконання (t, сек)			Похибка рішення (Δ , разів)		
	t _{з осн.}	t _{базовий}	t _{з покращ.}	Δ з осн.	Δ базовий	Δ з покращ.
4	0.672	1.154	0.924	0	0	0
8	1.655	5.510	4.243	0	0	0
12	3.257	13.037	8.735	0	0	0
16	5.468	24.615	22.153	0	0.002	0
20	7.767	36.551	29.972	0.002	0.008	0
24	17.511	52.904	40.736	0.01	0.012	0
28	30.645	95.227	52.096	0.019	0.016	0.01

З таблиці можна побачити що при збільшенні кількості міст, може почати зростати також і похибка рішення, однак на 28 вершина вона сягає не більше 3%. Навіть більш, менш навчений алгоритм може почати надавати невелику помилку у рішенні. Основним фактором в цьому є саме підбір оптимальних параметрів, зміна яких безпосередньо впливає на точність та час виконання. Неправильне налаштування є основною причиною, за якою точність двох алгоритмів – базового і з основою, стає значно

гіршою з ростом вершин ніж у навченого алгоритму, а також причиною тривалого виконання алгоритму, при досягненні тих самих результатів.

Розглядаючи конкретно отримані дані можна побачити, що даний модифікований алгоритм дозволяє пришвидшити пошук рішення у декілька разів на малих кількостях вершин, та приблизно у середньому на 30% коли їх кількість стає досить високою. Однак треба зазначити, що ціною за таку швидкість в деякій мірі стає втрата точності. Якщо спробувати правильно налаштувати параметри, наприклад за допомогою методів машинного навчання, можна спробувати досягти точності як навченого універсального алгоритму. При цьому може знизитися швидкість, але зважаючи на динаміку росту та поточні цифри, пришвидшення все одно може залишитися на рівні 30%.

Даний підхід до створення генетичного алгоритму з використання допоміжних методів і ідеї пов'язаних з конкретною задачею, як генерація оптимальної початкової колонії, можуть пришвидшити виконання алгоритму. Однак ціною використання даного підходу – є втрата універсальності генетичного алгоритму. Розглядаючи дану реалізацію, стає зрозуміло що алгоритм став конкретизованим під задачу комівояжера і більше не може бути використаний у інших задачах.

Втрата універсальності може стати критичним чинником і ускладнити налаштування алгоритму. Алгоритм використовує конкретизований підхід може мати додаткові, унікальні для нього параметри, чи унікальний підхід для будь-якого з етапів його роботи, що в свою чергу вимагає додаткових корегувань, складнішого тестування та навчання алгоритму.

Досліджуючи поточний алгоритм можна визначити декілька унікальних для генетичного алгоритму параметрів – величина підграфів які піддаються випадковому перемішуванню, частота для групи підграфів, швидкість зсуву межі та вплив нестачі генетичного різноманіття.

Для швидкості сходження популяцій, дуже сильний вплив має правильність перемішування підграфів. Даний параметр не є простим коефіцієнтом, а визначає собою масив чисел який відповідає шматкам які перемішуються, що ускладнює підбір правильного показника, зважаючи на те що для різною кількості

вершин він має динамічно змінюватися. Експериментально було визначено, що одним з найкращих способів розбиття є розбиття за ступенем двійки, від 1 до $N / 4$, де N кількість вершин, з додаванням у кінець числа $N / 4$, якщо воно відсутнє. До того ж коротші шматки в першій половині масиву мають перемішуватися вдвічі частіше.

Вплив нестачі генетичного різноманіття, також сильно впливає на остаточне вирішення. Однією з основних проблем алгоритму, є помилкове визначення моменту сходження популяції через те, що один з наборів генів швидше витісняє інші і більшість індивідів стають ідентичним, доходячи до точки коли перестають еволюціонувати. Пошук правильного коефіцієнту і алгоритму мутації може сприяти вирішенню цієї проблеми.

Одним з найвпливовіших вирішень проблеми нестачі генетичного різноманіття - є саме алгоритм селекції – дана реалізації суттєво пришвидшує алгоритм і допомагає знайти швидше оптимальне рішення з мінімальною похибкою.

Додатковими параметрами даного алгоритму можна визначити швидкість зсуву межі, та алгоритм з якого отримуються оптимальні рішення. Зсув межі не є унікальним варіантом для даного алгоритму, а є аналогом певного критерію для динамічної корекції показників таких як мутації, селекції та ін. Підбираючи алгоритм для пошуку початково рішення можна пробувати розв'язувати петлі за оптимальний час, знаходити оптимальні підграфи та ін. Дані параметри також можуть вплинути на роботу алгоритму.

Виходячи з цих даних можна стверджувати що в даному алгоритму існує один новий параметр що налаштовується безпосередньо – масив шматків та один з параметрів що має значно більшу вагу ніж у тривіальній версії алгоритму – коефіцієнт мутації.

Таким чином можна зробити висновок, що конкретизувавши алгоритм можна пришвидшити його для виконання певних задач, однак за рахунок ускладнення його навчання через ведення додаткових критерії та зміну ваги існуючих.

ВИСНОВКИ

Результати, отримані в магістерській роботі, є рішенням практичної задачі підвищення ефективності генетичних алгоритмів для пошуку оптимального шляху, використовуючи розроблений підхід, який передбачає модифікацію загальноприйнятого підходу та використання додаткових алгоритмів на графах.

Отримано такі теоретичні та практичні результати:

1. Проведено аналіз алгоритмів пошуку оптимального шляху на графах та оптимізації задачі. Досліджені основні методи штучного інтелекту які використовуються для вирішення даної задачі, виявлено ключові недоліки і потенційні можливості для удосконалення алгоритмів.

2. Проаналізовано сучасні підходи та інструменти для створення розподілених систем. Реалізовано систему та її алгоритми роботи згідно поставлених вимог. Розроблена система дозволяє автоматично оброблювати замовлення, групувати їх та автоматично надавати працівникам на обробку з розрахунком оптимального шляху доставки.

3. Розроблено підхід використання генетичних алгоритмів та реалізовано алгоритм, який дозволяє пришвидшити виконання пошуку оптимального шляху. Проведені дослідження демонструють що даний підхід використання цих алгоритмів дозволяє підвищити ефективність їх роботи майже на 30% з відносно мінімальною похибкою у порівнянні з іншими підходами.

Таким чином, всі поставлені в роботі задачі виконані й мета досягнута.