МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КОМПЛЕКСНА ВИПУСКНА РОБОТА

на тему:

«Інформаційна система формування туристичних маршрутів в Ферганській долині. Модуль формування одноденних маршрутів.»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Шелехов І.В.**

Студента групи ІН-73 — **Абдумухторов Б.А.**

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра комп'ютерних наук

Затверджую_____

Зав. кафедри Довбиш А.С.

"_____" _____20__р.

ЗАВДАННЯ

до випускної роботи

Студента п'ятого курсу, групи ІН73 спеціальності "Комп'ютерні науки" денної форми навчання Абдумухторов Б.А.

Тема: «Інформаційна система формування туристичних маршрутів в Ферганській долині. Модуль формування одноденних маршрутів.»

Затверджена наказом по СумДУ

№_____ от _____20__р.

Зміст пояснювальної записки: 1) аналіз проблеми та постановка задачі; 2) вибір методів розв'язання задачі; 3) розробка інформаційного і програмного забезпечення системи

Дата видачі завдання "_____"_____20__р.

Керівник випускної роботи _____Шелехов І.В.

Завдання прийняв до виконання _____ Абдумухторов Б.А.

РЕФЕРАТ

Записка: 33 стор., 7 рис., 5 табл., 1 додаток, 26 бібліографічних джерел.

Об’єкт дослідження – процес планування туристичних маршрутів .

Мета роботи — розробка алгоритму планування туристичних маршрутів.

Методи дослідження — методи багатовимірної глобальної оптимізації з використанням генетичних алгоритмів.

Результати — В роботі була спроектовано і реалізовано систему планування туристичних маршрутів в Ферганській долині. Вибір оптимального маршруту проводився на базі модифікованого генетичного алгоритму шляхом вирішення задачі декількох комівояжерів. Програмна реалізація планувальника виконана в середовищі для наукових і інженерних розрахунків MATLAB.

ГЕНЕТИЧНИЙ АЛГОРИТМ, ХРОМОСОМИ, ГЕН,
КРОСИНГОВЕР, МУТАЦІЯ, ФІТНЕС-ФУНКЦІЯ,
ЗАДАЧА ДЕКІЛЬКОХ КОМІВОЯЖЕРІВ

ЗМІСТ

ВСТУП

Успішне функціонування фірми на ринку туристичного бізнесу практично немислимо без використання сучасних інформаційних технологій. Специфіка технології розробки і реалізації турпродукту вимагає таких систем, які в найкоротші терміни надавали б відомості про доступність транспортних засобів та можливості розміщення туристів, забезпечували б швидке резервування і бронювання місць, а також автоматизацію рішення допоміжних завдань при наданні туристичних послуг (паралельне оформлення таких документів, як квитки, рахунки і путівники, забезпечення розрахункової і довідковою інформацією). Особливо актуальні системи, що дозволяють користувачеві самому вносити зміни в кінцевий продукт, тим самим персоналізуючи його за своїм бажанням. Це можна досягти за умови широкого використання в туризмі сучасних комп'ютерних технологій обробки і передачі інформації. Тому розробка спеціалізованих інформаційних систем в сфері туризму в наші дні особливо актуальна. Метою даної роботи є автоматизація процесу формування туристичного маршруту.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Формування індивідуальних маршрутів в сфері туризму

Сучасний темп розвитку суспільного середовища з кожним роком все збільшується, що тягне за собою розвиток інших сфер соціальної взаємодії. Завдання логістичної комунікації все частіше і частіше постають перед суспільством і вимагають нових рішень. Поліпшення засобів пересування і зв'язку, складання нових способів і алгоритмів взаємини між громадськими структурами -все це є шляхами вирішення виникаючих задач. Турізм як одна з соціальних сфер тісно пов'язаний з маршрутизацією, і процес розробки і оптимізації маршрутів, а також їх індивідуалізація в даний час набувають найбільшу актуальність. Зазвичай туристичний сектор відстає від інших секторів, пов'язаних з логістикою, а також бізнесом і сферою освіти, в тому числі, через недостатню увагу, приділеною цій сфері. За останні роки туристичний сектор значно активувався. Звичні списки пам'яток щороку поповнюються, завдяки активності на місцевому та регіональному рівнях. Наукове співтовариство в цій сфері сьогодні націлене на кількісний розвиток середовища та повернення забутих туристичних об'єктів назад в поле культури і бізнеса. Актуальною відмінною рисою наукового дискурсу в сфері туризму є націленість на відродження забутих досягнень минулого і включення нових об'єктів культури. Зараз туристична сфера потребує якісного стрибку в одному з основоположних чинників сфери, а саме маршрутизації. Згідно даним останніх соціальних опитувань, люди все частіше займаються самоорганізацією і просвітою, і одним з факторів стримування розвитку туристичного сектора є незручність організації особистого дозвілля. Все рідше люди використовують туристичні бюро за їх прямим призначенням через розбіжність особистих інтересів з пропонованими програмами. Рішення даних проблем може бути істотно підтримана шляхом розробки нового методичного, математичного, програмного забезпечення для формування групових та індивідуальних екскурсійних програм, що дозволить підвищити привабливість об'єктів туризму і звернути на них увагу як бізнесу, так і сфери

освіти. Виникають якісно нові методи, засновані на парадигмі формування динамічних туристичних маршрутів, який дозволить формувати більш підходящих під сучасні реалії туристичного бізнесу екскурсійні програми. При цьому можна виділити кілька питань, які підлягають розгляду в ході реалізації проекту:

-відсутність стандартизації опису туристичних об'єктів;

-визначення універсальних критеріїв оптимізації маршруту;

-складність організації регіональних туристичних маршрутів.

Для визначення вимог до програмного забезпечення потрібне уточнення поточної ситуації та проведення огляду аналогічних рішень. Були обрані наступні критерії порівняння програмних засобів:

- можливість часткової роботи оффлайн;
- можливість додавати та переглядати відгуки;
- відстеження тегів інтересів на потенційних точках маршруту;
- інтеграція з соціальними мережами.
- можливість бронювати готелі, ресторани, купувати квитки;
- інтеграція з місцем розташування на карті;
- можливість додавати дані про місця;
- інтелектуальна формування маршруту;
- можливість змінювати обраний маршрут.

На основі проведеного аналізу було виявлено, що в існуючих програмних рішеннях відсутні технології формування індивідуальних маршрутів на основі особистих інтересів, а також не вирішується завдання складання маршрутів в оптимізаційній постановці. Крім цього, було визначено тенденція на ринку програмного забезпечення вбудовувати в розроблену продукцію можливість автономної роботи без прив'язки до інтернету.

1.2 Задача декількох комівояжерів при формуванні туристичних маршрутів

Задача комівояжера - одна з добре відомих мультидисциплінарних задач в галузі досліджень операцій та інформатики, яка полягає в пошуку найменшого по довжині або вартості гамільтонівського циклу (ланцюга) у мережі міст.

Задачу можна визначити так:

Дано набір міст (вузлів) та відстань між ними. Починаючи і закінчуючи в одному місті, комівояжер повинен відвідати усі інші міста так, щоб загальна пройдена ним відстань (вартість) була мінімальна.

Задача комівояжера широко вивчався в науковому товаристві і для її розв'язання було запропоновано декілька підходів. У випадку, коли є необхідність керувати діями одночасно двох або більше комівояжерів, розглядається узагальнення такої задачі, що має назву задача декількох комівояжерів. У ній всі комівояжери починають і закінчують свою подорож в одному місті. При цьому кожне місто, крім початкового міста, повинен відвідувати лише один комівояжер, а загальна відстань (вартість) пройдена усіма комівояжерами повинна бути мінімальною [1].

Задачу декількох комівояжерів можна формально визначити так:

Нехай у мережі з n міст присутні m комівояжерів, які розташовані в одному початковому місті, d_{ij}, (i, j = 1,2, ..., n) - відстань (вартість) між містами i та j. Місто з індексом i=1 – початкове, а решта міст, i = 2, 3, ..., n - проміжні міста. Кожен з комівояжерів повинен вийти з початкового міста і після обходу його набору міст повинен повернутися знову до початкового міста. У маршрутах різних комівояжерів не повинно бути спільних міст (крім початкового). Метою є отримання оптимального плану маршрурів, тобто перелік міст для кожного комівояжера, такого, щоб загальна відстань (вартість) маршрутів була мінімальною. Очевидно, що якщо m = 1, проблема стає звичайною задачею комівояжера.

В матриці відстаней може подаватися не лише довжина маршруту між двома містами, але і його вартість, час тощо. Залежно від характеру матриці відстані, задачі комівояжера поділяються на два типи - асиметричні та симетричні. Якщо $d_{ij} = d_{ji}$, для будь-якої пари i, j, то це симетрична задача, в іншому випадку асиметрична. Для звичайних асиметричних задач комівояжера для n міст існує $(n\text{-}1)!$ можлива кількість маршрутів. Отже, для випадку з 5 містами існує 24 ймовірні маршрути, а для задачі з 6 містами – 120 маршрутів. Проте для задачі з 10 містами існує 362 880 можливих маршрутів, перевірка яких буде займати досить значний час. Таким чином, обчислювальні витрати прямо пропорційна розмірності задачі. Дуже важко, а іноді навіть не можливо за прийнятний час вирішити задачі великої розмірності. Крім того, в задачі декількох комівояжерів спочатку потрібно визначити міста, що виділяються кожному з них, а потім знайти їх оптимальну послідовність, отже, така задача буде складнішою, ніж звичайна задача комівояжера. Оскільки звичайна задача комівояжера є NP-складною, то і задача декількох комівояжерів також є NP-складною [2].

Задача декількох комівояжерів є однією найскладніших задач оптимізації в дослідженні операцій та інформатиці, що взаємопов’язана з різними практичними проблемами планування та маршрутизації. При цьому задача декількох комівояжерів вважається більш підходящим, ніж звичайна задача комівояжера, для практичних застосувань і може бути використана для моделювання багатьох реальних процесів. Наприклад, задача може бути застосована до планування роботи декількох паралельних виробничих ліній [3]. Крім того, проблема формування маршруту транспортного засобу може також бути змодельована як задача декількох комівояжерів. Така задача може бути застосований до іншого виду варіанту звичайної задачі комівояжера, коли необхідно відвідати n міст протягом періоду, що охоплює m тижнів і повернутися до початкового міста у вихідні дні [4]. Проблема планування маршруту шкільного автобуса полягає у застосуванні задача декількох комівояжерів, що визначає схему завантаження автобуса так, щоб загальна

кількість шляхів була зведена до мінімуму, загальна відстань, яку проїжджають усі автобуси, була мінімальною, жоден автобус не повинен бути перевантажений і час, необхідний для проходження будь-якого маршруту не повинен перевищує максимально дозволене значення [5]. Ще одне застосування задачі декількох комівояжерів [6] – планування роботи декількох груп фотографів в декількох школах. Також є статті, в яких до такої задачі зведено планування друкованої преси [4], планування співбесід [7], планування місій [8] та проектування глобальних мереж навігаційних супутникових зйомок [9]. Задача декількох комівояжерів може мати і інші формулювання [1]. Міста може бути таким, що потребують одноразового або багаторазового відвідування. Маршрути можуть бути закритими або відкритими. Закритий маршрут починається і закінчується в одним місті, тоді як відкритий – не вимагає повернення до початкового місто.

У цій роботі розглядається задача декількох комівояжерів для створення туристичних маршрутів, що починаються і закінчуються в одному початковому місті. Така задача є NP-складною, тобто для її вирішення не існує алгоритмів з поліноміальним обчислювальним часом. Отже, знайти її оптимальне рішення дуже важко, а то й неможливо. Таким чином, дослідники шукають кращих евристичних рішень за прийнятний обчислювальний час замість точних оптимальних рішень задача декількох комівояжерів, а також інших складних задач оптимізації. Отже, слід використовувати евристичні методи вирішення цієї задачі. Штучна нейронна мережа [10], модельований отжиг [11], генетичний алгоритм [12], оптимізація рою частинок [13], оптимізація колонії мурашок [14] тощо – це основні підходи, що дозволяють знайти евристичне рішення задачі декількох комівояжерів.

.

1.3Постановка задачі

Результати проведеного аналітичного огляду підтверджує актуальність використання інтелектуальних інформаційних технологій в туристичній

галузі. Одна з таких задач пов'язана з плануванням туристичного маршруту. Якісне планування дозволяє розраховувати найкращий маршрут між початковим і кінцевим пунктом туристичної подорожі.

Для вирішення завдання розробки інформаційної системи планування туристичного маршруту необхідно виконати наступні завдання:

1) сформувати вхідний математичний опис системи;
2) розробити та програмно реалізувати алгоритм пошуку найкращого маршруту;
3) сформувати вхідні дані для тестування працездатності системи у вигляді набору потенційних точок туристичного маршруту;
4) перевірити працездатність системи на прикладі планування маршруту в Ферганській долині.

2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

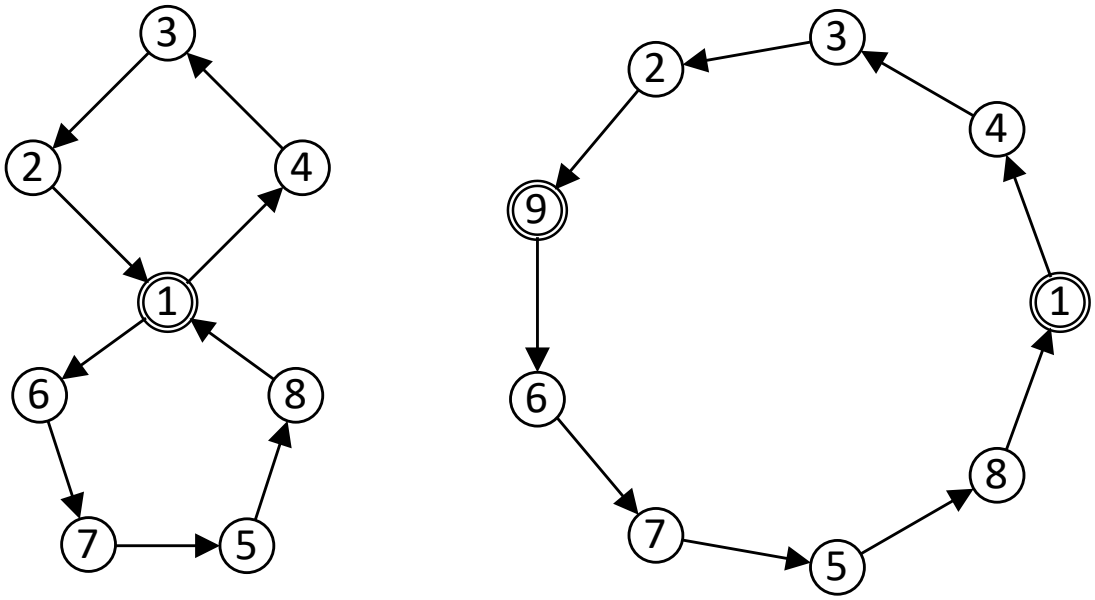
2.1 Генетичний алгоритм

В останні роки було успішно розроблено кілька генетичних алгоритмів для різних складних задач оптимізації, наприклад, для квадратичної задачі про призначення [15], задачі побудови мінімального остовного дерева [16] та задача комівояжера [17]. Генетичний алгоритм вперше розроблений Джоном Холандом в 1970-х роках на основі теорії виживання в природі найбільш пристосованих істот, характеристики яких задаються довільними варіаціями в структурі хромосом. Генетичний алгоритм виявився дуже ефективним через його просту, гнучкість і легкість в реалізації. Генетичний алгоритм завжди починається з початкової популяції хромосом, яка проходить в основному три основні операції, а саме селекцію, кросинговер та мутацію, що достатньо для формування нового покоління більш пристосованих істот популяцій. В ході селекції хромосоми обираються з певною імовірністю, що залежить від значення функції пристосованості, в наступне (ітераційне) покоління. Кросовер випадковим чином відбирає дві батьківські хромосоми та поєднує їх, утворюючи нові хромосоми (нащадки). Мутація змінює значення (ген) у положенні хромосоми. Кросовер разом із селекцією є найвпливовішим процесом у генетичному алгоритмі. Мутація розширює пошуковий простір і захищає втрати генетичного матеріалу, які можуть виникнути внаслідок селекції та операторів кросоверу. Отже, ймовірність реалізації мутаційного оператора фіксується дуже низькою, тоді як ймовірність реалізації кросоверу фіксується дуже високою [18]. З трьох генетичних операторів кросовер є найважливішим оператором, тому для задачі декількох комівояжерів було розроблено декілька варіантів кросоверу. Експериментальне дослідження [19] показало, що послідовний конструктивний кросовер є найкращим оператором. Проте нещодавно було розроблено кілька модифікованих версій такого кросоверу, а саме адаптивний кросовер [17], жадібний кросовер [20], обернений жадібний кросовер [21] та всеосяжний кросовер [21].

У цьому роботі буде спочатку зведено задачу декількох комівояжерів до задачі одного комівояжера шляхом введення штучних міст, а потім розроблено генетичний алгоритм на базі послідовного конструктивного кросоверу.

2.2 Класичний генетичний алгоритм для задачі декількох комівояжерів

Задача декількох комівояжерів може бути зведена до звичайної задачі одного комівояжера. При цьому n міст для m комівояжерів перетворюються на $n + m - 1$ місто для одного комівояжера. При цьому необхідно додати $m - 1$ штучне початкове місто (а саме $n + 1$, ..., $n + m - 1$) [22]. Приклад рішення задачі декількох комівояжерів з $n = 8$, $m = 2$ показано на рис. 2.1 (а), тоді як його перетворення в задачу одного комівояжера зображено на рис. 2.1 (б).



а) б)

Рисунок 2.1 – Графічне відображення маршрутів а) задачі декількох комівояжерів та б) відповідної задачі комівояжера

Матриця відстаней для задачі декількох комівояжерів та матриця відстаней для відповідної задачі одного комівояжера з одним штучним початковим містом № 9 наведена в таблицях 2.1 та 2.2.

Загалом, генетичний алгоритм є дуже ефективним евристичним методом при пошуку рішень для звичайної задачі комівояжера та її різних варіацій. І хоча він і не гарантує знаходження оптимального розв’язку задачі, але зазвичай надає максимально наближений до нього варіант.

Таблиця 2.1 – Матриця відстаней для задачі декількох комівояжерів

Місто	1	2	3	4	5	6	7	8
1	0	75	99	9	35	63	8	11
2	51	0	86	46	88	29	20	15
3	100	5	0	16	28	35	28	2
4	20	45	11	0	59	53	49	8
5	86	63	33	65	0	76	72	5
6	36	53	89	31	21	0	52	6
7	58	31	43	67	52	60	0	9
8	15	95	66	14	54	8	87	0

Таблиця 2.2 – Матриця відстаней для задачі комівояжера

Місто	1	2	3	4	5	6	7	8	9
1	0	75	99	9	35	63	8	11	0
2	51	0	86	46	88	29	20	15	51
3	100	5	0	16	28	35	28	2	100
4	20	45	11	0	59	53	49	8	20
5	86	63	33	65	0	76	72	5	86
6	36	53	89	31	21	0	52	6	36
7	58	31	43	67	52	60	0	9	58
8	15	95	66	14	54	8	87	0	15
9	0	75	99	9	35	63	8	11	0

Щоб використовувати GA для будь-якої задачі оптимізації, слід знайти метод, що представляє рішення за допомогою хромосом. Існує три методи подання, що використовуються для хромосом для задачі декількох комівояжерів. Це метод з однією хромосомою [22], метод з двома хромосомами [23] та метод із двох частин хромосоми [24-25]. В [26] запропоновано інший метод представлення хромосом, який розвиває метод представленням хромосом з [24]. Це подання складається з двох розділів - головного розділу та розділу групи. Основна частина хромосоми складається з n справжніх генів, а групова частина складається з m цілочисельних генів. Цілочисельна частина справжнього гена і в головному розділі вказує

комівояжера, призначеного місту і, тоді як дробова частина визначає порядок відвідування міста *i*. Розділ групи просто показує, що різні групи існують у рішенні так, як вони з'являються в основному розділі. Це відхилення від подання, описаного в [24], де групи можуть відображатися в будь-якому порядку в розділі груп. В [23] запропоновано інший метод представлення хромосом, який представляє хромосому як набір з *m* маршрутів, тобто серед маршрутів немає послідовності.

В роботі використовується метод з однією хромосомою зі штучними містами. Приклад нашої хромосоми для $n = 8$ з $m = 2$ наведено на табл. 2.3.

Таблиця 2.3 – Приклад хромосоми для маршруту (рис. 2.1 б)

1	4	3	2	9	6	7	5	8

У цій хромосомі (1, 4, 3, 2, 9, 6, 7, 5, 8) є $n + m - 1 = 8 + 2 - 1 = 9$ генів, включаючи штучне початкове місто №9. Ця хромосома представляє маршрут $\{1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 9 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 8 \rightarrow 1\}$. Це означає, що 1-й комівояжер відвідував міста 1, 4, 3 і 2 послідовно, а 2-й – міста 6, 7, 5 та 8 послідовно. Цільова функція визначається як загальна відстань, пройдена комівояжерами, яка становить (9 + 11 + 5 + 51 + 63 + 52 + 52 + 5 + 15 =) 263 для цього маршруту.

Спочатку генерується набір випадкових хромосом (початкова популяція) для запуску генетичного процесу пошуку, які потім оцінюються та проходять процедуру селекції для створення множини кросоверу.

Оператор послідовного конструктивного кросоверу було розроблено для звичайної задачі комівояжеру, а лише потім модифіковано для задачі декількох комівояжерів. Він створює лише одно нащадка, що використовує кращі гени батьків. Більше того, він використовує деякі кращі гени, які відсутні у батьків. Основною характеристикою послідовного конструктивного кросоверу є послідовний аналіз хромосом батьків з метою визначення першого міста, що ще не було відвідано після поточного міста. Якщо в батьків немає жодного не відвіданого міста, хромосоми батьків перевіряються з

початку. Потім порівнюється відстані від поточного міста, щоб вибрати наступне місто в хромосомі нащадка. Цей оператор кросовера можна застосовати для задачі декількох комівояжерів, але без модифікацій він може призвести до хромосом, що відповідають нездійсненним маршрутам. Модифікація полягає в застосуванні алгоритму обміну містами на нездійсненних хромосомах в такий спосіб, щоб вони стали здійсненними. Алгоритм обміну випадковим чином вибирає два міста та обмінюється ними. Тепер ми проілюструємо послідовний конструктивний кросовер, використовуючи такі батьківські хромосоми з 9 містами та 2 комівояжерами, P_1: (1, 5, 7, 2, 9, 4, 3, 6, 8) та P_2: (1, 4, 6, 3 , 7, 9, 2, 5, 8) із загальними відстанями маршрутів 265 та 420 відповідно з використанням матриці відстані (табл.2.2).

Місто №1 є початковим містом для P_1 та P_2.

Наступними містами, що ще не відвідувалися і не увійшли до хромосоми нащадку є №5 і №4 в P_1 і P_2, відповідно з $d_{15} = 35$ і $d_{14} = 9$. Оскільки $d_{14} < d_{15}$, то другим геном в хромосомі нащадку приймається місто №4. Отже, неповна хромосома нащадку тепер має два гени (1, 4).

Наступними містами після №4 є №3 в P_1 і №6 в P_2, відповідно з $d_{43} = 11$ і $d_{46} = 53$. Оскільки $d_{43} < d_{46}$, то третім геном нащадку буде місто №3. Отже, неповна хромосома нащадку тепер має три гени (1, 4, 3).

Наступними містами після №3 є №6 в P_1 і №7 в P_2, відповідно з $d_{36} = 35$ і $d_{37} = 28$. Оскільки $d_{37} < d_{36}$, то четвертим геном нащадку буде місто №7. Отже, неповна хромосома нащадку тепер має чотири гени (1, 4, 3, 7).

Наступними містами після №7 є №2 в P_1 і №9 в P_2, відповідно з $d_{72} = 31$ і $d_{79} = 58$. Оскільки $d_{72} < d_{79}$, то п’ятим геном нащадку буде місто №2. Отже, неповна хромосома нащадку тепер має п’ять ген (1, 4, 3, 7, 2).

Наступними містами після №2 є №9 в P_1 і №5 в P_2, відповідно з $d_{29} = 51$ і $d_{25} = 88$. Оскільки $d_{29} < d_{25}$, то шостим геном нащадку буде місто №9. Отже, неповна хромосома нащадку тепер має шість генів (1, 4, 3, 7, 2, 9).

Наступними містами після №9 є №6 в P_1 і №5 в P_2, відповідно з $d_{96} = 63$ і $d_{95} = 35$. Оскільки $d_{95} < d_{96}$, то сьомим геном нащадку буде місто №5. Отже, неповна хромосома нащадку тепер має сім генів (1, 4, 3, 7, 2, 9, 5).

Наступними містами після №5 є №7 в P_1 і №8 в P_2, відповідно з $d_{57} = 72$ і $d_{58} = 5$. Оскільки $d_{58} < d_{57}$, то восьмим геном нащадку буде місто №8. Отже, неповна хромосома нащадку тепер має вісім генів (1, 4, 3, 7, 2, 9, 5, 8).

Наступними містами після №8 є №6 в P_1 і №6 в P_2, відповідно з $d_{86} = 8$ Таким чином, дев’ятим геном нащадку буде місто №6. Отже, повна хромосома нащадку з дев’яти генів має вигляд (1, 4, 3, 7, 2, 9, 5, 8, 6).

Сумарна довжина цього маршруту 214, що краще, ніж у обох батьків.

Оператор мутації, як правило, випадково відбирає ген в хромосомі і модифікує його значення. При цьому зазначається, що мутація може допомогти іншим операторам подолати локальний оптимум і отримати краще рішення. В цій роботі оператор мутації випадковим чином обирає два гени, крім початкового і штучних початкових місць і міняє їх місцями.

В роботі використовується класичний генетичний алгоритм без використання іншого евристичного методу. Починаючи з випадково створеної первинної популяції, за допомогою послідовності з ймовірнісної селекції та послідовного конструктивного кросоверу та мутація, що міняє місцями гени формуються нові популяції. При цьому критерієм останову є обмеження по кількості кроків такого алгоритму.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Формування вхідного математичного опису

В рамках розвитку різноманітних транспортних та туристичних проектів у Євразії Ферганська долина вважається досить складним випадком. В давні часи Ферганська долина знаходилася на «перетині давньогрецької, китайської, бактрийської і парфянської цивілізацій», була важливим пунктом на Шовковому шляху, оточена горами та розділена між трьома державами Центральної Азії: Киргизстаном, Таджикистаном та Узбекистаном.

Перед тим, як розглянути безпосередньо до тему дослідження, необхідно виділити ряд характеристик Ферганської долини:

- Ферганська долина – це еліпс, довжиною 300 і шириною 170 кілометрів. Долина оточена горами і має один прямий вихід без переходів по перевалам - через таджицький місто Худжанд.
- Територія Ферганської долини розділена між трьома державами: Киргизькою Республікою, Республікою Таджикистан та Республікою Узбекистан.
- Ферганская долина - найкращий за щільністю населення район Центральної Азії. Загальне заселення областей трьох країн, між якими поділена долина, складає близько 15 мільйонів чоловік або близько 30% всього населення трьох країн. У киргизькій частині проживає 3,5 мільйона людей, в узбецькій частині долини проживає 9,5 мільйонів людей, а в таджицькій частині - 2,5 мільйона людей.
- Ферганская долина – це своєрідна провінція для таких столиць, як Ташкент, Бішкек і Душанбе.

  Основними містами Ферганської долини є:

Коканд - столиця могутнього Кокандского ханства в XVI-XX ст До наших днів зберігся оригінальний палац Худоярхана (1871 г.) - останнього хана Коканда, в якому розташований місцевий історичний музей. В

облицюванні використана майоліка оригінальної гами фарб і візерунків, що не зустрічається в інших областях Узбекистану. Також цікаві медресе Норбута-Бия (1799) і мавзолей Модаріхон (1825 г.).

Маргилан - невелике старовинне містечко поруч з Ферганой. У X ст. це місто було центром шовкового виробництва, де зараз знаходиться єдина в Середній Азії фабрика ручного виробництва шовку (хан-атласу). Туристи можуть простежити всю технологію створення шовкової продукції від вирощування шовкопряда до забарвлення готових тканин і купити оригінальні сувеніри на пам'ять. Також цікавий комплекс Кафтарлік (1742 г.) з мавзолеєм святого Бенкету-Сиддіка.

Андижан - батьківщина Захірітдіна Бобура, засновника династії Великих Моголів, що правили в Індії більше 400 років. Зараз тут знаходиться меморіальний парк Бобура і музей історії і найбільша в Ферганській долині Джума-мечеть (XIX ст.).

Наманган веде свою історію з XVII століття, у XVII столітті сюди, після руйнівного землетрусу переселилися жителі Аксікента (Ахсикента) - стародавнього міста, що був колись центром Ферганської долини. В середині XVIII століття місто стало адміністративним центром вілоята. У 1875 році Наманган увійшов до складу Росії. Тоді ж було закладено за регулярним планом нове місто. Його відділяла від старого міста фортеця, від якої, за тим самим сформованому порядку, віялом розходилися радіальні вулиці. На початку XX століття Наманган був другим за населенням містом і центром по обробці бавовни в Ферганській долині, а потім і третім за величиною в республіці, після Ташкента і Самарканда. В даний час Наманган розвивається як центр легкої і харчової промисловості. Місто потопає в зелені фруктових садів і парків. Недалеко від Наманган знаходяться руїни стародавнього міста Аксікента. Археологічні дослідження показали, що місто мало цитадель і потужні кріпосні стіни. Тут була розвинена торгівля і ремісниче виробництво. До XIII століття Аксікент був столицею Ферганської долини. Але потім місто було зруйноване монголами, а в XVII столітті сильним землетрусом.

Кувасай - великий промисловий центр, де діють 5 великих промислових підприємств. У місті Кувасай є Палац культури, 3 професійних коледжу, 2 лікарні. У 2000 році в центрі міста Хокіміат Кувас виділила земельну ділянку, де був закладений православний храм святого Іоанна Кронштадського, освячення якого відбулося в 2012 році. Визначною пам'яткою міста є площа Леніна, пам'ятник «Дружба народів». До 1990-х років у місті було пам'ятник, зведений в пам'ять про загиблих в роки Великої Вітчизняної війни, який згодом був знищений. Викликає інтерес також реконструйована міська мечеть, яка зовні нагадує Тадж-Махал.

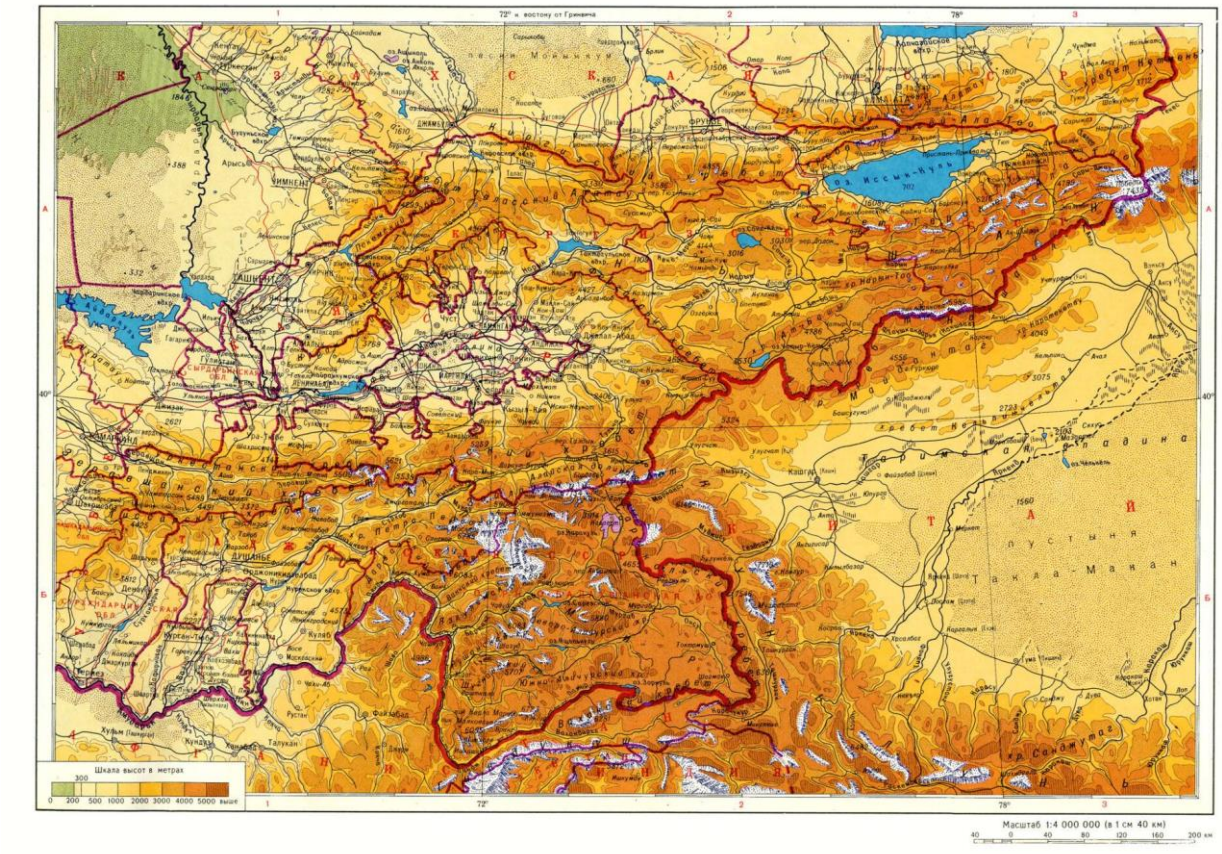


Рисунок 3.1 – Карта Ферганської долини

В цілому в роботі розглядаються 9 міст Ферганської долини : Вуадиль Маргилан, Киргили, Коканд, Андижан, Кувасай, Фергана, Наманган, Чимион.

Таблиця 3.1 – Квадратна матриця відстаней між точками туристичного маршруту

№	Назва	Маргилан	Коканд	Андижан	Наманган	Кувасай	Киргили	Фергана	Чимион	Вуадиль
1	Маргилан	0	77	80	67	33	9	15	37	42
2	Коканд	77	0	132	119	113	99	84	73	103
3	Андижан	80	132	0	68	80	81	79	109	110
4	Наманган	67	119	68	0	102	76	80	106	111
5	Кувасай	33	113	80	102	0	25	21	53	36
6	Киргили	9	99	81	76	25	0	7	33	32
7	Фергана	15	84	79	80	21	7	0	31	33
8	Чимион	37	73	109	106	53	33	31	0	34
9	Вуадиль	42	103	110	111	36	32	33	34	0

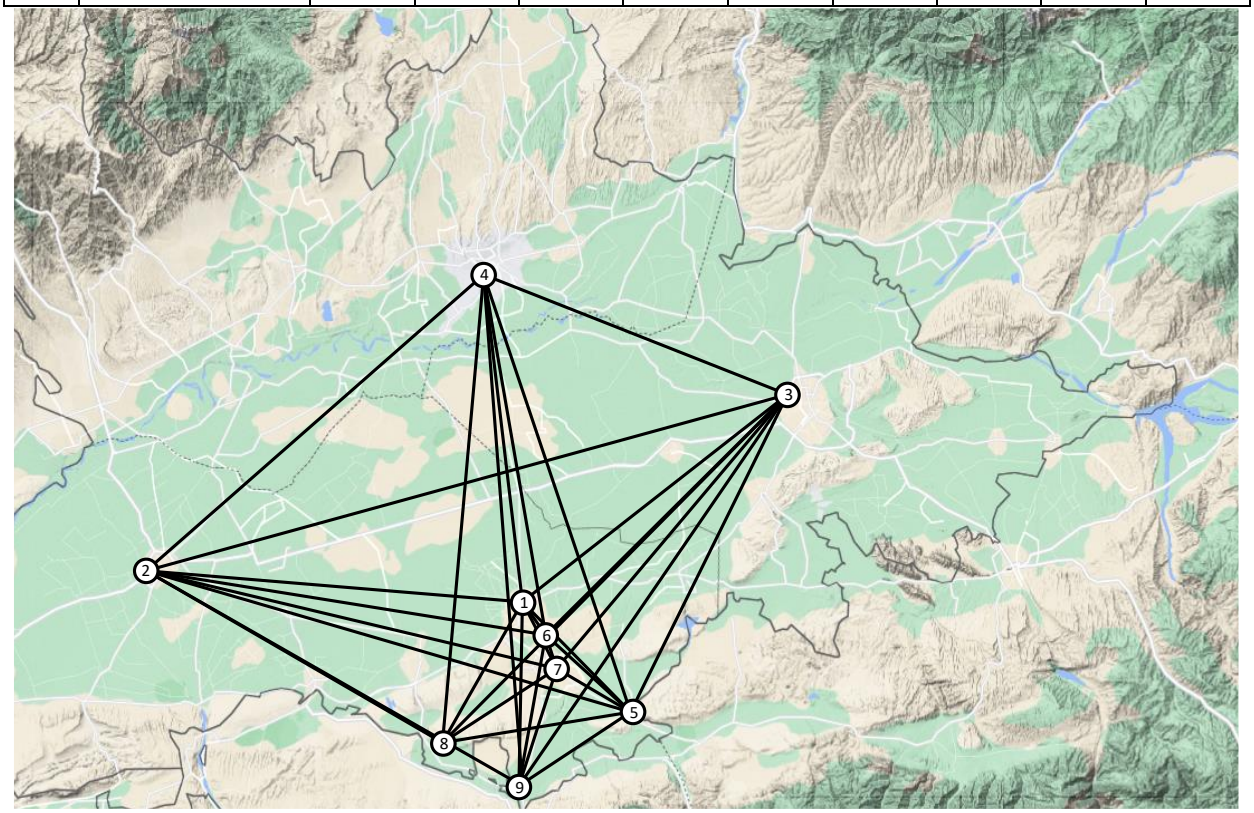
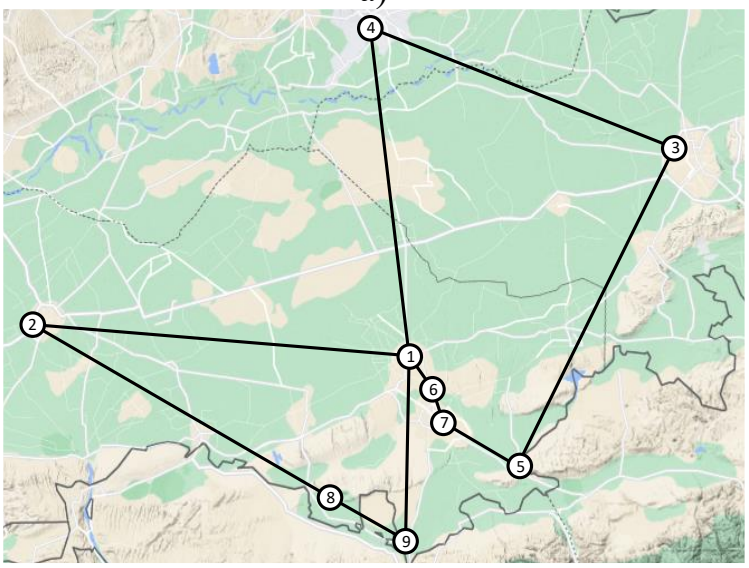


Рисунок 3.2 – Туристичні маршрути Ферганської долини (повний граф)

Для відображення варіантів декількох одноденних маршрутів Ферганською долиною у вигляді хромосом будемо використовувати масив з $n+m-1$ генів, де n – кількість міст, m – кількість днів туристичного маршруту.

1	4	3	5	7	6	10	2	8	9	1

а)



б)

Рисунок 3.2 – Один з варіантів маршруту: а) хромосома б) граф

3.2 Програмна реалізація

Matlab або Matrix Laboratory - це мова програмування високого рівня, що складається з інтерактивного середовища, що в основному використовується для чисельних обчислень, програмування та візуалізації. Вона була розроблений компанією MathWorks. Основними функціями Matlab є побудова графіків функцій та даних, створення користувальницьких інтерфейсів, маніпуляції з матрицями. Він також забезпечує підтримку взаємодії з іншими мовами програмування на C, C ++, Fortran та Java. Крім того, він також використовується для аналізу даних, створення моделей та додатків, а також для розробки алгоритмів. Поряд із усім цим, Matlab також надає безліч вбудованих функції для математичних операцій, що включають численні обчислення, виконання числових методів, генерацію графіків та багато інших функцій. Нижче наведено найпоширеніші функції та математичні обчислення, що використовуються в Matlab -

1. Робота з матрицями та масивами
2. 2-D та 3-D графіки та графіки

3. Лінійна алгебра
4. Алгебраїчні рівняння
5. Нелінійні функції
6. Статистика
7. Аналіз даних
8. Обчислення та диференціальні рівняння
9. Чисельні обчислення
10. Інтеграція
11. Перетворює
12. Підгонка кривої
13. Різні інші спеціальні функції

За основу інформаційної системи було взято набір m-функцій, які було розроблено в першій частині комплексної роботи:

main.m - основна функція, яка призначена для виклику інших функцій і відображення результатів обчислень;

popul_fitness.m - розрахунок фітнес-функції;

popul_mutation.m - реалізація мутації;

popul_cross.m - реалізація схрещування;

popul_gen.m - створення початкової популяції;

points_gen.m - введення опорних точок;

points_show.m - графічне відображення вхідних даних.

При цьому було виконано модифікацію

popul_cross.m, що тепер реалізує послідовний конструктивний кросовер замість двоточкового кросинговеру;

popul_gen.m, що тепер реалізує додавання до опорних точок штучних початковихточок.

Повний код наведено в додатку.

3.3 Аналіз результатів

При генеруванні початкової популяції будемо вважати що кількість днів проходження туристичного маршруту дорівнює 2 і після кожного дня туристичній групі необхідно повернутися в початкове місто. Таким чином, n =9, m = 2 Згідно алгоритму декількох комівояжерів необхідно додати m-1 = 1 штучне початкове місто. Популяція згенерована на першому кроці подана в табл. 3.2

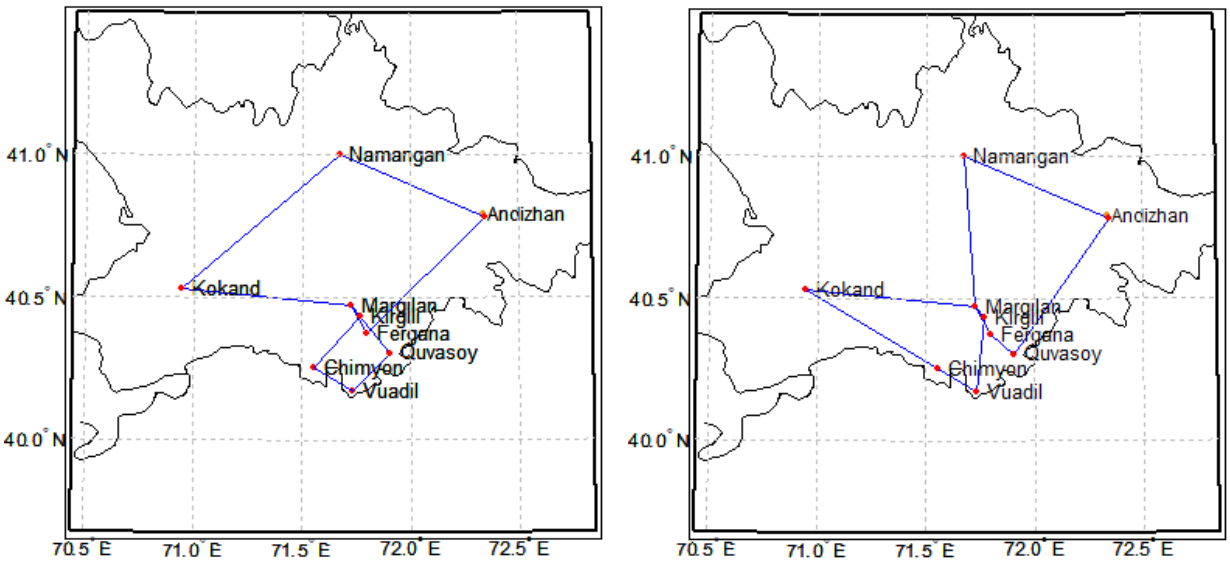
Таблиця 3.2 – Початкова популяція

№	Хромосоми											Фитнес функція
1	1	7	5	10	6	8	9	2	3	4	1	515
2	1	8	4	3	5	6	7	9	2	10	1	536
3	1	9	7	2	6	5	8	3	4	10	1	580
4	1	2	3	8	5	4	9	6	7	10	1	638
5	1	8	3	9	5	7	10	6	4	2	1	609
6	1	8	3	4	6	5	7	2	9	10	1	565
7	1	9	3	10	8	2	5	7	6	4	1	626
8	1	7	5	3	9	2	6	8	4	10	1	634
9	1	2	5	7	3	4	6	8	9	10	1	543
10	1	6	10	7	5	3	8	4	2	9	1	613

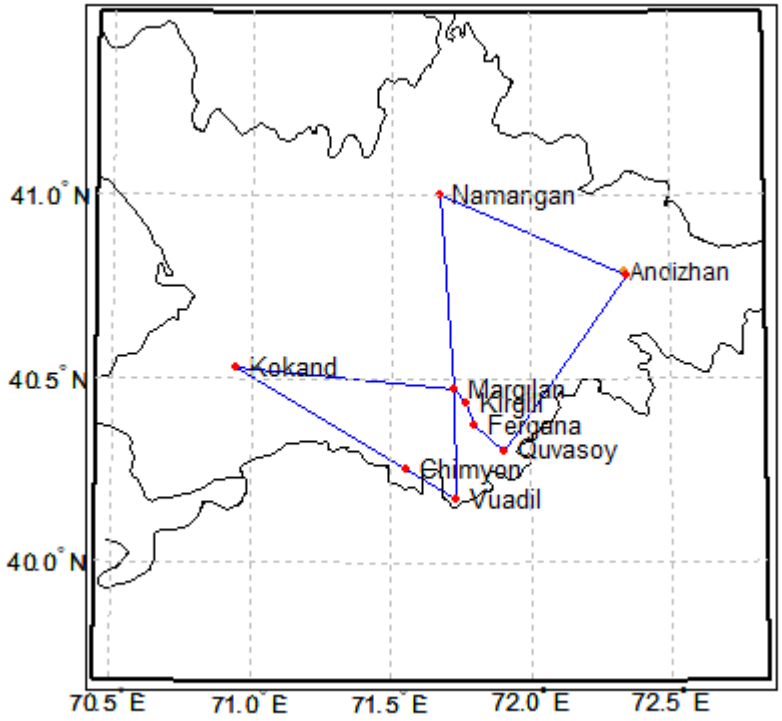
Перша, остання точки маршруту та точка з номером 10 відповідають одному і тому самому початковому місту.

Як фітнес-функція використовувалась сума відстаней між точками маршруту, але на відміну від стандартного генетичного алгоритму необхідно було знайти мінімум такої функції, тобто найкоротший маршрут.

На рис. 3.4 подано графічне відображення найкоротшого маршруту на різних кроках роботи генетичного алгоритму. При цьому як крок розглядають формування нової популяції з 10 хромосом.



а) б)



в)

Рисунок 3.4 – Графічне відображення маршрутів сформованих після а) 10 кроків б) 100 кроків в) 2000 кроків роботи генетичного алгоритму

Графік зміни значення фінтес-функції в ході оптимізації наведено на рис. 3.5. На цьому графіку відображається найменше значення фінтес-функції в популяції сформованої на кожному кроці роботи генетичного алгоритму.

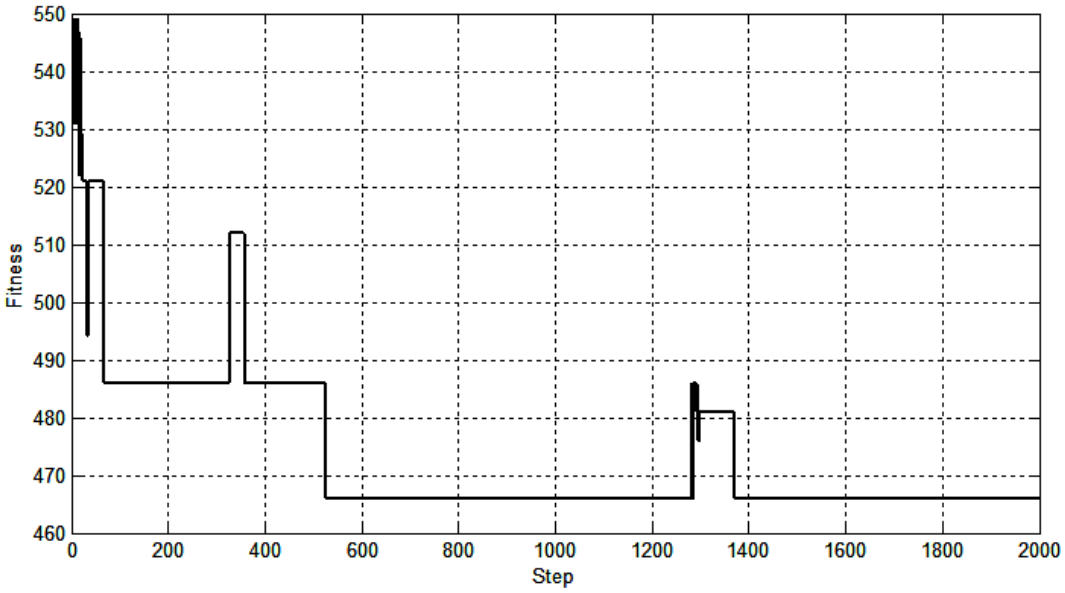


Рисунок 3.5 – Динаміка зміни значень фітнес-функції в ході роботи генетичного алгоритму

Аналіз рис. 3.5 показує, що на 2000 кроці навчання було сформовано популяцію, до якої належить декілька хромосом, що характеризуються фінтес-функцію 466. Графічне відображення відповідних маршрутів наведено на рис.3.6

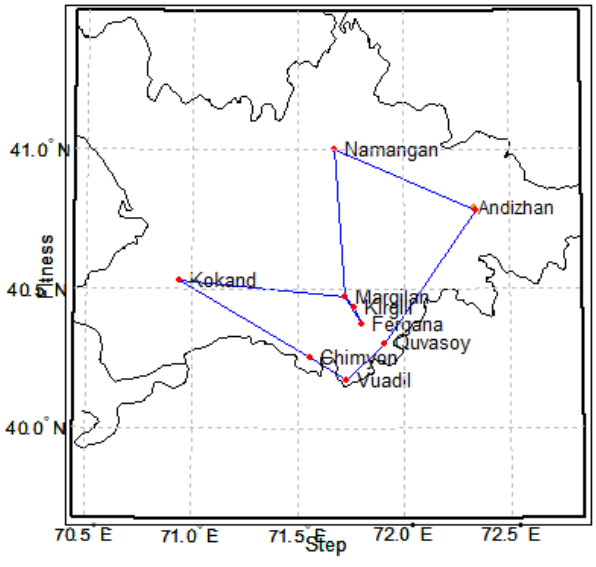


Рисунок 3.6 – Графічне відображення найкоротшого маршруту, що складається з двох одноденних

ВИСНОВКИ

В роботі проведено аналітичний огляд застосування інтелектуальних інформаційних технологій в туристичній галузі. Одна з таких задач пов'язана з плануванням туристичного маршруту. Якісне планування дозволяє розраховувати найкращий маршрут між початковим і кінцевим пунктом туристичної подорожі.

Для вирішення завдання розробки інформаційної системи планування туристичного маршруту виконано наступні завдання:

1) сформувано вхідний математичний опис системи;
2) розроблено та програмно реалізувано алгоритм пошуку найкращого маршруту;
3) сформувано вхідні дані для тестування працездатності системи у вигляді набору потенційних точок туристичного маршруту;
4) перевірити працездатність системи на прикладі планування множини одноденних маршрутів в Ферганській долині.

СПИСОК ЛІТЕРАТУРИ

1. T. Bektas,“The multiple traveling salesman problem: An overview of formulations and solution procedures,”Omega,vol. 34, pp. 209–219, 2006.
2. M.R. Garey, and D.S. Johnson,“A guide to the theory of np-completeness, Computers and intractability,”W. H. Freeman & Co., New York, NY, USA, 1990.
3. A.E. Carter,andC.T. Ragsdale,“Scheduling pre-printed newspaper advertising inserts using genetic algorithms,”Omega, vol. 30, pp. 415–421, 2002.
4. A.E. Carter,and C.T. Ragsdale,“A new approach to solving the multiple traveling salesperson problem using genetic algorithms,”European Journal of Operational Research,vol. 175, pp. 245–257, 2006.
5. R.D. Angel, W.L. Caudle, R. Noonan, and A. Whinston,“Computer assisted school bus scheduling,”Management Science, vol. 18, pp. 279–288,1972.
6. T. Zhang, W.A. Gruver, and M.H. Smith,“Team scheduling by genetic search,”In Proceedings of the second international conference on intelligent processing and manufacturing of materials, 1999,vol.2, pp. 839–844.
7. K.C. Gilbert, and R.B. Hofstra, “A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem,”Decision Sciences, vol. 23, pp. 250–259,1992.
8. B. Brummit, and A. Stentz, “Dynamic mission planning for multiple mobile robots,”In Proceedings of the IEEE international conference on robotics and automation, April 1996.
9. H.A. Saleh, and R. Chelouah, “The design of the global navigation satellite system surveying networks using genetic algorithms,”Engineering Applications of Artificial Intelligence, vol. 17, pp. 111–122, 2004.
10. E. Wacholder, J. Han, and R.C. Mann, “A neural network algorithm for the multiple traveling salesmen problem,”Biology in Cybernetics, vol. 61, pp. 11–19, 1989.

11. C. Song, K. Lee, and W.D. Lee, “Extended simulated annealing for augmented TSP and multi-salesmen TSP,”In Proceedings of the international joint conference on neural networks, 2003, vol.3, pp. 2340–2343.
12. S. Yuan, B. Skinner, S. Huang, and D. Liu, “A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms,”European Journal of Operational Research, vol. 228, pp. 72–82, 2013.
13. X.S. Yan, C. Zhang, and W. Luo, “Solvetraveling salesman problem using particle swarm optimization algorithm,”International Journal of Computer Science, vol. 9(6), pp. 264–271, 2012.
14. G. Singh, and R. Mehta, “Implementation of travelling salesman problem using ant colony optimization,”Journal of Engineering Research and Application, vol. 6(3), pp. 385–389, 2014.
15. Z.H. Ahmed, “A simple genetic algorithm using sequential constructive crossover for the quadratic assignment problem,”Journal of Scientific & Industrial Research, vol. 73(12), pp. 763-766, 2014.
16. K. Singh, and S. Sundar, “A hybrid genetic algorithm for the degree-constrained minimum spanning tree problem,”Soft Computing, vol. 24, pp. 2169–2186, 2020.
17. Z.H. Ahmed, “Adaptive sequential constructive crossover operator in a genetic algorithm for solving the traveling salesman problem,”International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11(2), pp. 593-605, 2020.
18. D.E. Goldberg, “Genetic algorithms in search, optimization and machine learning,”Addison-Wesley, Reading, MA, 1989.
19. M.A. Al-Omeer, and Z.H. Ahmed, “Comparative study of crossover operators for the MTSP,”In proceedings of 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 2019,pp. 1-6.
20. Z.H. Ahmed, “Solving the Traveling Salesman Problem using Greedy Sequential Constructive Crossover in a Genetic Algorithm,”IJCSNS

International Journal of Computer Science and Network Security, vol. 20(2), pp. 99-112, 2020.

21. Z.H. Ahmed, “Genetic algorithm with comprehensive sequential constructive crossover for the travelling salesman problem,”International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11(5), pp. 245-254, 2020.
22. L. Tang, J. Liu, A. Rong, and Z. Yang, “A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex,”European Journal of Operational Research, vol. 124, pp. 267–282, 2000.
23. Singh, and A.S. Baghel, “A new grouping genetic algorithm approach to the multiple traveling salesperson problem,”Soft Computing, vol. 13, pp. 95-101, 2009.
24. L. Davis, “Job-shop scheduling with genetic algorithms,”In Proceedings of an International Conference on Genetic Algorithms and their Applications, pp. 136-140, 1985.
25. I.M. Oliver, D.J. Smith, and J.R.C. Holland, “A study of permutation crossover operators on the travelling salesman problem,”In Genetic Algorithms and Their Applications: Proceedings of the 2nd International Conference on Genetic Algorithms, J.J. Grefenstette, Ed. Lawrence Erlbaum Associates, Hilladale, NJ, 1987.
26. D.E. Goldberg, and R. Lingle, “Alleles, loci and the travelling salesman problem,”In Genetic Algorithms and their Applications: Proceedings of the 1st International Conference on Genetic Algorithms, J.J. Grefenstette, Ed. Lawrence Erlbaum Associates, Hilladale, NJ, 1985.

ДОДАТОК

m-сценарій main.m

```
clear;
m=2;n=9;
n_pop=10;
[p,ns,ds]=points_gen;
for k=2:m
    ds(end+1,:)=ds(1,:); ds(end,1)=999;
    ds(:,end+1)=ds(:,1); ds(1,end)=999;
end;
points_show(p,ns,0.05);
pop=popul_gen(n,m,n_pop);
[fit,p_norm]=popul_fitness(pop,ds);
for  step=1:2000
min_fit=min(fit);
opt_n=find(fit==min_fit);
pop_opt=pop(opt_n(1),:);
gr(step)=min_fit;
opt_path=[1 pop(opt_n(1),:) 1];
opt_path(find(opt_path>n))=1;
pop=popul_cross(pop,p_norm,ds);
pop=popul_mutation(pop,0.01);
[fit,p_norm]=popul_fitness(pop,ds);
max_fit=max(fit);
noopt_n=find(fit==max_fit);
pop(noopt_n(1),:)=pop_opt;
end;
for i=1:n+m;
 opt_p(i,:)=p(opt_path(i),:);
end;
hold on
linem(opt_p(:,1), opt_p(:,2), 'b-');
opt_path
gr(end)
hold off;
```

m-функція popul_fitness.m

```
function [fitness,p_norm]=popul_fitness(popul,points_dist);
[k n]=size(popul);
 for i=1:k
   fitness(i)= points_dist(1,popul(i,1));
   for j=1:n-1
       fitness(i)=fitness(i)+points_dist(popul(i,j),popul(i,j+1))
   end;
   fitness(i)= fitness(i)+points_dist(popul(i,n),1);
end;
   fit_max=max(fitness)*1.1;
   t=(fit_max-fitness)./sum(fit_max-fitness);
   p_norm(1)=0;
   for i=1:k-1
       p_norm(i+1)=p_norm(i)+t(i);
   end;
end;
```

m-функція popul_mutation.m

```
function [new_popul]=popul_mutation(popul,p_mutation);
    [k n]=size(popul);
    t=zeros(1,1);
    for i=1:k
        for j=1:n
            if rand(1,1)<p_mutation
                tn=round(rand(1,1)*(n-1))+1;
                t=popul(i,tn);
                popul(i,tn)=popul(i,j);
                popul(i,j)=t;
            end;
        end;
    end;
    new_popul=popul;
end;
```

popul_cross.m

```
function [new_popul]=popul_cross(popul,p_norm,ds);
    [k n]=size(popul);
    new_popul=[];

for j=1:n
    ch=[];
    pt=rand(1,2);
    n_pt1=find(p_norm<=pt(1));
    n_pt2=find(p_norm<=pt(2));
    np1=n_pt1(end);
    np2=n_pt2(end);
    pt1=popul(np1,:);
    pt2=popul(np2,:);

    if ds(1,np1)<ds(1,np2)
        ch(1)=pt1(1);
    else
        ch(1)=pt2(1);
    end;
    n1=find(pt1==ch(1));
    n2=find(pt2==ch(1));
    pt1(n1)=[];
    pt2(n2)=[];
    if n1>size(pt1) n1=1; end;
    if n2>size(pt2) n2=1; end;
for i=2:n-1
        if ds(n1,np1)<ds(n2,np2)
            ch(i)=pt1(n1);
        else
            ch(i)=pt2(n2);
        end;
    n1=find(pt1==ch(i));
    n2=find(pt2==ch(i));
    pt1(n1)=[];
    pt2(n2)=[];
    if n1>size(pt1) n1=1; end;
    if n2>size(pt2) n2=1; end;
end;
ch(n)=pt1(1);
new_popul=[new_popul; ch];
end;

end;
```

popul_gen.m

```
function [popul]=popul_gen(n,m,popul_size);
for k=1:popul_size
    ch=2:(n+m-1);
    for i=1:(n+m-1)
        j=round(rand(1,2)*((n+m-1)-2))+1;
        t=ch(j(1));
        ch(j(1))=ch(j(2));
        ch(j(2))=t;
    end;
popul(k,:)=[ch];
end;
end;
```

points_gen.m - введення опорних точок;

```
function [points,names,distances]=points_gen()
points=[
    40.47240, 71.7246;
    40.52861, 70.9425;
    40.78210, 72.3442;
    40.99830, 71.6726;
    40.30538, 71.9079;
    40.43583, 71.76722;
    40.37338, 71.79783;
    40.2526512, 71.5575586;
    40.1742, 71.7301
];
names=[
    'Margilan';
    'Kokand  ';
    'Andijon ';
    'Namangan';
    'Quvasoy ';
    'Kirgili ';
    'Fergana ';
    'Chimyon ';
    'Vuadil  '
];
distances=[
    0   77  80  67  33  9   15  37  42;
    77  0   132 119 113 99  84  73  103;
    80  132 0   68  80  81  79  109 110;
    67  119 68  0   102 76  80  106 111;
    33  113 80  102 0   25  21  53  36;
    9   99  81  76  25  0   7   33  32;
    15  84  79  80  21  7   0   31  33;
    37  73  109 106 53  33  31  0   34;
    42  103 110 111 36  32  33  34  0];
```

points_show.m - графічне відображення вхідних даних.

```
function points_show(points,names,eps);
lims=minmax(points');
lims(:,1)=lims(:,1)-eps*10;
lims(:,2)=lims(:,2)+eps*10;

worldmap('hi',lims(1,:),lims(2,:));
for j=1:size(points,1)
    textm(points(j,1), points(j,2)+eps, names(j,:));
    linem(points(j,1), points(j,2), 'r.');
end;

end;
```