

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Інформаційна система формування туристичних
маршрутів в Ферганській долині.
Модуль формування багатоденного маршруту»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Шелехов І.В.

Студента групи ІН-73

Рахманов Т.Е.

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.

“ _____ ” _____ 2021 р.

ЗАВДАННЯ
до випускної роботи

Студента п'ятого курсу, групи ІН-73 спеціальності “Комп'ютерні науки”
денної форми навчання Рахманова Темура Еркіновича.

**Тема: «Інформаційна система формування туристичних маршрутів в
Ферганській долині. Модуль формування багатоденного маршруту»**

Затверджена наказом по СумДУ

№ _____ від _____ 2021 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 “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Шелехов І.В.

Завдання прийняв до виконання _____ Рахманов Т.Е.

РЕФЕРАТ

Записка: 33 стор., 6 рис., 2 табл., 1 додаток, 12 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 — У роботі було спроектовано і реалізовано систему планування туристичних маршрутів в Ферганській долині. Вибір оптимального маршруту проводився на базі модифікованого генетичного алгоритму шляхом вирішення задачі комівояжера. Програмна реалізація планувальника виконана в середовищі для наукових і інженерних розрахунків MATLAB.

ГЕНЕТИЧНИЙ АЛГОРИТМ, ХРОМОСОМИ, ГЕН,
КРОСИНГОВЕР, МУТАЦІЯ, ФІТНЕС-ФУНКЦІЯ,
ЗАВДАННЯ КОМІВОЯЖЕРА

ЗМІСТ

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
1.1 ЗАДАЧА ПЛАНУВАННЯ ТУРИСТИЧНИХ МАРШРУТІВ.....	6
1.2 АНАЛІЗ ТЕНДЕНЦІЙ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ В ТУРИСТИЧНІЙ ГАЛУЗІ.....	8
1.3 ПОСТАНОВКА ЗАДАЧІ.....	11
2 ВИБІР МЕТОДУ РОЗВ’ЯЗАННЯ ЗАДАЧІ.....	13
2.1 ОСНОВНІ ВИЗНАЧЕННЯ ТА ПРИНЦИПИ ТЕХНОЛОГІЇ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ	13
2.2 КЛАСИЧНИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ	14
2.3 ОПИС ГЕНЕТИЧНОГО АЛГОРИТМУ РОЗВ’ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА	19
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	22
3.1 ФОРМУВАННЯ ВХІДНОГО МАТЕМАТИЧНОГО ОПИСУ	22
3.2 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	25
3.3 АНАЛІЗ РЕЗУЛЬТАТІВ	26
ВИСНОВКИ.....	29
СПИСОК ЛІТЕРАТУРИ.....	30
ДОДАТОК.....	32

ВСТУП

Останнім часом туризм в Узбекистані активно розвивається через виходу країни на міжнародний туристичний ринок і значного економічного ефекту галузі. При цьому поряд з великими містами, туризм розвивається і в регіонах. Багато локальні культурні об'єкти були заново відкриті завдяки популяризації соціально-економічного потенціалу окремих регіонів, а також завдяки зусиллям регіонального наукового співтовариства, яке створює різноманітні стратегії для туристичної децентралізації та залучення туристичного потоку в місцеві культурні центри. Одним з ключових напрямків розвитку туризму є максимальна індивідуалізація проектування туристичних маршрутів, що реалізується в формалізації слабо структурованих показників в «профілі» туриста, а також сприяє конкурентоспроможності таких маршрутів. Серед туристів стають все більш популярними маршрути, задовольнити які сучасні типові виїзні туристичні програми не в змозі. Великі можливості для підтримки організації туризму відкривають засоби автоматизації, за допомогою яких можна будувати індивідуальні туристичні маршрути з урахуванням побажань клієнта. Таким чином, особливу актуальності набуває розробка гнучкої програми формування туристичних програм. Створення сайтів, груп в соціальних мережах, відео тощо збільшення згадки об'єктів підвищеного інтересу хоч і збільшує рейтинг згадки місць, але не завжди дає бажаний результат. Проблемою широкого застосування засобів автоматизації стає та обставина, що в більшості випадків йде або невірне, або недостатньо комплексне дослідження предметної області та її формальний опис в моделі даних інформаційної системи. Метою даної роботи є створення автоматизованої системи планування багатоденних туристичних маршрутів за допомогою технології генетичних алгоритмів.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Задача планування туристичних маршрутів

Задача планування може бути вирішена введенням типізованого інформаційного опису пам'яток, а також зв'язків між ними, що дозволить будувати і оптимізувати унікальні туристичні маршрути для різних завдань виховної роботи. Для організації багатогранного туризму, відповідного для різних категорій користувачів з максимально індивідуалізованими уподобаннями, недостатньо надавати тільки список пам'яток, а й важливо також знати опис культурно-історичної або іншої цінності пам'ятки, варіанти логістики досягнення з супутніми показниками ціни, часу відвідування, визначення місць схожою тематичної спрямованості. Подібна стратегія надає можливість туристичного охоплення пам'яток цілими тематичними кластерами. Спрощуючи систему і переробляючи вже наявну основу, можна розробити більш яка підходить під сучасні реалії базу культурно-історичних об'єктів, формують потенціал туристичної привабливості регіону.

Серед великого різноманіття туристичних напрямків особливої уваги заслуговує освітній туризм, часто реалізується в рамках виховної роботи освітнього закладу. В цьому випадку туризм сприймається додатково як засіб виховання гармонійно розвиненої і толерантної особистості, психолого-педагогічної корекції поведінкової моделі учня, формування у нього мотивації до саморозвитку і додаткової освіти. Зважаючи на відсутність стандартизації опису регіональних культурних об'єктів, придатних для відвідування індивідуальними туристами, туристичними групами, які навчаються освітніх установ тощо виникають додаткові проблеми для організації виїздів в рамках програми виховної роботи, базується на концептуальній методичній основі, яка передбачає розвиток особистості за тематичними напрямками (патріотичне, загальнокультурний, спортивне, художнє тощо). Таким чином, потрібно інформаційне забезпечення цього процесу, що дозволяє враховувати індивідуальні переваги різних учнів і

вибудовувати тематичні маршрути з мінімальними витратами тимчасових і економічних ресурсів. Крім класифікації по об'єктах відвідування, існують види туристичної діяльності з точки зору впливу на культурний розвиток особистості, такі як :

1. Подієвий туризм - поїздки на конкретні фестивалі, спортивні змагання, музичні конкурси тощо.

2. Сільський туризм - поїздки з тривалим зануренням шляхом безпосереднього проживання у самотньої культурної середовищі народу.

3. Культурно-пізнавальний туризм - відвідування різних історичних, культурних пам'яток та цілісних маршрутів; участь у вивченні фольклору та прикладного мистецтва.

4. Екологічний туризм - подорожі в природні резервації з метою вивчення регіональної культури.

5. Релігійний туризм конфесійно-пізнавальної спрямованості.

Аналіз предметної області також показує, що залучення туризму як засоби формування толерантності стає більш ефективним в разі рішення таких проблем, як:

1. Використання педагогічних практик в рамках туристичної діяльності з метою впливу на того, хто навчається.

2. Необхідність залучення фахівців різних сфер, зокрема, культурологів і працівників соціально-культурної сфери в якості експертів за визначенням важливості туристичного об'єкта.

3. Популяризація туризму з урахуванням попиту населення.

Таким чином, можна зробити висновок про те, що при використанні коштів автоматизації для розробки індивідуальних туристичних маршрутів необхідно мати формальну модель опису пам'яток; визначити критерії оптимізації маршрутів і їх співвідношення (час, вартість, пріоритет відвідування тощо); мати можливість використовувати експертні оцінки різних фахівців, здатних оцінити пріоритет туристичної привабливості тієї чи іншої пам'ятки. Розробка методу формування туристичних маршрутів,

з'єднують всі вищевикладені вимоги, і алгоритму, заснованого на цьому методі, є прогресивними кроками для розвитку всієї сфери регіонального туризму.

1.2 Аналіз тенденцій застосування технологій штучного інтелекту в туристичній галузі

Туризм є динамічною сервісно-орієнтованою галуззю. У роботі [1] відзначається, що 45% дорослого населення США користується спеціальними ресурсами з метою пошуку місць для подорожей, відпочинку або проведення дозвілля, бронювання готелів, машин, а також квитків на літаки та поїзди. Схожа тенденція спостерігається також в деяких країнах Європи. Широке використання потенційними туристами спеціальних ресурсів породило появу електронного туризму (e-tourist); розвиток інтелектуальних веб-сервісів для вироблення рекомендацій, які допомагають клієнту визначитися з маршрутом подорожей та інформаційних систем прийняття рішень в індустрії туризму. Труднощі, пов'язані з розробкою ефективних інформаційних систем прийняття рішень в індустрії туризму, обумовлені необхідністю реалізації методів для обробки великих обсягів різномірної інформації і впровадження діалогу з користувачем системи, який часто не здатний чітко сформулювати критерії для пошуку місця відпочинку. Таким чином, головним завданням при розробці відповідних сервісів і інформаційних систем прийняття рішень в індустрії туризму є розробка ефективних механізмів спрямованого пошуку потрібної інформації.

В даний час в усьому світі спостерігається тенденція «заміщення» традиційної індустрії туризму індустрією електронного туризму (e-tourism). Електронний туризм традиційно включає електронні сервіси наступних категорій: інформаційні сервіси (наприклад, надання інформації про пункти призначень, туристичних маршрутах або готелях); комунікаційні сервіси (наприклад, обговорення з клієнтом або відправка електронної пошти); сервіси угод або транзакцій (наприклад, інформація про бронювання,

резервування). У зв'язку зі стрімко зростаючим числом веб-ресурсів різних агентств, які надають туристичні послуги, по всьому світу потенційному туристу-користувачів Інтернет складно орієнтуватися у величезному обсязі інформації, тому для електронної комерції в цілому і для індустрії електронного туризму зокрема актуальним завданням є розробка ефективних інтелектуальних сервісів аналізу веб-контенту та інтелектуальних інформаційних систем, що здійснюють «спілкування» з користувачем на природній мові; пошук інформації, що задовольняє запитам користувача, і прийняття рішень щодо маршруту подорожі.

Одним з основних підходів до реалізації механізму прийняття рішень в інформаційних системах в індустрії туризму є метод міркувань на основі прецедентів (case-based reasoning ілі CBR), що реалізовується, як правило, в рекомендаційних системах, заснованих на знаннях. Процес функціонування прецедентної системи підтримки і прийняття рішень можна представити у вигляді CBR цикла, що складається з наступних основних фаз [3]:

- 1) на підставі відносини подібності здійснюється вибір найбільш підходящого прецеденту або безлічі прецедентів з наявного сховища прецедентів (раніше зустрічалися випадків);

- 2) використання обраних прецедентів для прийняття рішення поставленого завдання;

- 3) перегляд і корекція завдання в разі потреби приймалися раніше в обраних прецедентах рішень;

- 4) збереження в базі даних прийнятого рішення і ситуації, що склалася в якості нового прецеденту або відповідну зміну обраного прецеденту для подальшого використання.

Система, яка реалізує метод міркувань на основі прецедентів при надходженні опису переваг з боку клієнта (туриста) здійснює пошук схожих описів і відповідних їм рішень в сховище прецедентів. Сховище прецедентів, по суті, являє собою репозиторій, який зберігає інформацію про минулі подорожах. Прикладом системи автоматизованої вироблення рекомендацій у

сфері туризму є проект DIETORECS [1,4], заснований на реалізації підходу колаборативної фільтрації і методу міркувань на основі прецедентів.

Пріоритетними напрямками при розробці інтелектуальних систем в індустрії туризму є системи вироблення рекомендацій на основі методу колаборативної фільтрації; використання технології семантичної «павутини» та інтелектуальних агентів.

Семантична «павутина» або технологія семантичного веб (semanticweb) дозволяє подати інформацію системи у вигляді семантичної мережі за допомогою онтологій, на основі яких можна будувати логічні висновки. Іншими словами, семантичну павутину можна розглядати як сукупність моделей та інструментів, що дозволяють інтелектуально створювати і обробляти туристичний контент програмними засобами. Онтології є моделі даних на основі класів, підкласів і властивостей і описуються на основі веб-мови онтологій-OWL (WebOntologyLanguage). Мова OWL була запропонований WorldWideWebConsortium (W3C) для опублікування і розповсюдження даних, а також автоматизації обробки даних комп'ютерами з використанням онтологій на основі веб. При описі предметної області електронного туризму онтологія дозволяє відповісти на 4 типи питань, які визначаються предикатами: «що» (що може робити турист під час відпочинку (проведення дозвілля)), «де» (де розташовані цікавлять туриста пам'ятки), «коли» (коли турист може відвідати його цікавить місце або пам'ятку) і «як» (як турист може дістатися до цікавить його пункту призначення). Двома найважливішими технологіями для розвитку семантичного веб є XML (ExtensibleMarkupLanguage) і RDF (ResourceDescriptionFramework). Мова XML розглядається як стандарт обміну даними на основі веб-технології. RDF використовує XML і реалізує модель для опису і побудови взаємозв'язків між ресурсами. Міжнародною організацією туризму (WorldTourismOrganization) в якості міжнародного стандарту був запропонований тезаурус по туризму та проведення дозвілля, що містить онтології, що описують різні області електронного туризму.

Прикладами проектів, що розвивають технологію семантичного веб та пропонують підходи до опису інфраструктури електронного туризму на основі онтологій, є HARMO-TEN (Tourism Harmonisation Trans-European Network TEN), SATINE (semantic-based Interoperability Infrastructure for Integrating Web Service Platforms to Peer-to-Peer Networks) [5].

Ще одним напрямком організації інтелектуального пошуку на туристичних веб-ресурсах є використання інтелектуальних агентів, тобто спеціальних програм («роботів»), які на основі аналізу контенту веб-ресурсу, з урахуванням побажань потенційного туриста, здійснює пошук найбільш підходящих місць відпочинку або проведення дозвілля. В роботі [2] при розробці інтелектуального агента в галузі електронної комерції використовується метод навчання з підкріпленням (reinforcement learning), заснований на навчанні агента в процесі взаємодії з середовищем. В індустрії електронного туризму в якості середовища може розглядатися список бажаних користувачем подорожей, в якості агента - спеціальна програма, яка записує дані про проведені кліки на сторінці веб-ресурсу, що описує туристичні послуги, і веб-перегляду веб-ресурсу. Інтелектуальний агент аналізує записану інформацію про переваги користувача, використовуючи систему заохочень (заохочення отримують ресурси, які користувач вибирав, клацнувши по посиланню) і штрафів (штраф отримують ресурси, не вибрані користувачем), і на основі аналізу видає список найбільш ймовірних переваг користувача зі списку подорожей .

1.3 Постановка 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) сформулювати вхідний математичний опис системи;
- 2) розробити та програмно реалізувати алгоритм пошуку найкращого маршруту;
- 3) сформулювати 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 Основні визначення та принципи технології еволюційних алгоритмів

Еволюційні алгоритми (ЕА), до числа яких відносяться генетичні алгоритми [26], еволюційні стратегії [28], алгоритми генетичного програмування [27] тощо, беруть початок в перших роботах по еволюційному моделюванню Л. Фогеля з співавторами [19] і Дж. Голланда [26], де було запропоновано моделювати процес біологічної еволюції для синтезу ефективних структур і створення систем штучного інтелекту. Майже одночасно з виникненням методу еволюційного моделювання, А.Г. Івахненком і Л.А. Растрігіним були запропоновані методи групового обліку аргументів [7] і випадкового пошуку [15], де також використовувалися ідеї еволюції і випадкової мутації. Основний принцип роботи ЕА заснований на комп'ютерному моделюванні процесу еволюції з урахуванням факторів мінливості, успадкування та відбору найбільш пристосованих особин. В ЕА еволюційне моделювання, як правило, використовується не для дослідження біологічних популяцій, а для вирішення завдань оптимізації та прийняття рішень. Базова ідея застосування ЕА в оптимізації полягає в побудові деякої множини (популяції) допустимих рішень оптимізаційної задачі і застосуванні «малих» випадкових перетворень з метою отримання рішень вищої якості (більш пристосованих). При цьому чим більше пристосованість особини, тим більше вона має шансів бути обраною в якості батьківської для побудови нових рішень. ЕА мають численні застосування при рішенні різноманітних завдань в управлінні, плануванні, проектуванні, розпізнаванні образів і в інших областях [3, 12, 13, 16]. Одним з типових представників ЕА є класичний генетичний алгоритм (КГА), запропонований Дж. Голландом [26]. У цьому алгоритмі пристосованість особин до умов «навколишнього середовища» подається у вигляді деякої монотонної функцією від значення цільової функції задачі. Чим вище якість допустимого рішення, що

розуміється як значення цільової функції, тим вище його пристосованість. Популяція розвивається за рахунок добору батьківських особин за допомогою оператора пропорційної селекції і застосування до них випадкових операторів, що імітують мутацію генів і рекомбінацію батьківських генотипів (кросинговер). В різноманітних варіантах генетичних алгоритмів (ГА) використовуються різні оператори відбору, при цьому особини з великим значенням пристосованості, в середньому, отримують більше число нащадків на наступному поколінні. При створенні ГА знаходить застосування біонічний підхід, що полягає в запозиченні принципів організації систем з живої природи. В такому випадку має місце запозичення принципу поступових адаптивних перетворень в межах популяції або виду в ході, так званої, мікроеволюції

2.2 Класичний генетичний алгоритм

Класичний генетичний алгоритм (КГА) був запропонований Дж. Голландом [26], як алгоритм, що імітує адаптацію популяції до заданої функції пристосованості. При цьому, як і в популяційній генетиці [1], під пристосованістю розуміється середнє число нащадків від особини даного генотипу в певних умовах навколишнього середовища. Згодом КГА став активно використовуватися і як метод оптимізації [16, 24]. нехай задача оптимізації «на максимум» має загальний вигляд

$$\max\{f(x): x \in X\} \quad (2.1)$$

де X - простір рішень, наприклад, евклідовий простір R^n або простір $\{0,1\}^n$ булевих рядків довжини n . Процес роботи КГА являє собою послідовну зміну популяцій (поколінь), що складаються з фіксованого числа особин. Тут і далі під особиною розуміється елемент простору рішень $x \in X$ або деяке його подання до алгоритмі. Чим більше значення цільової функції особини, тим більше шансів вона має залишити нащадків в наступному

покоління і тим більше пристосованість відповідного генотипу (цей принцип легко адаптується і застосовується до завдань мінімізації). При формуванні наступного покоління частина нащадків повністю ідентична батьківським особинам, а частина змінюється деяким випадковим чином під дією операторів мутації і кросинговеру (схрещування). При використанні КГА рішення представляються двійковими рядками фіксованої довжини l . Кожному рядку з l символів алфавіту $\{0,1\}$ відповідає елемент простору X , тобто визначається функція $x: B \rightarrow X$, де $B = \{0,1\}^l$. Ця функція називається схемою подання розв'язків. Рядки $\xi \in B$ прийнято називати генотипами або хромосомами, а їх образи $x(\xi) \in X$ - фенотипами. Тут і далі для позначення елемента простору рішень використовується той же символ x , що і для схеми подання $x()$, проте, з контексту зміст символу x завжди буде ясний. Вживання термінів «фенотип» і «генотип» пов'язане з тим, що при аналізі і застосуванні генетичних алгоритмів важливе значення має спосіб внутрішнього подання розв'язків в алгоритмі. Елементи рядка $\xi \in B$ прийнято називати генами за аналогією з ділянками молекули ДНК. якщо простір рішень X має нескінченну потужність, то в КГА його неявно обмежують множиною лише тих рішень (фенотипів), які подано генотипами.

Популяцією $\Pi = (\xi_1, \xi_2, \dots, \xi_N)$ чисельності N називається вектор простору B^N , елементами якого є генотипи особин даної популяції. Спосіб нумерації особин в популяції КГА не має значення. Популяція покоління t буде позначатися через $\Pi_t = (\xi_{1t}, \xi_{2t}, \dots, \xi_{Nt})$. Ітерацією КГА є перехід від поточної популяції Π_t до наступної популяції Π_{t+1} . Чисельність популяції N фіксована від початку роботи алгоритму до кінця і для зручності вважаємо, що N парне. При оцінці «якості» особини в КГА замість цільової функції $f()$ використовується функція пристосованості генотипу $\Phi(\xi) = \varphi(f(x(\xi)))$, де $\xi \in B$. Тут $\varphi: R \rightarrow +$ - деяка функція, що монотонно зростає, обрана при адаптації КГА до конкретного завдання оптимізації з урахуванням її специфіки. Найпростішим прикладом функції пристосованості є сама цільова функція $f(x(\xi))$, якщо вона невід'ємна. В біологічній інтерпретації максимуми функції

$\Phi(\xi)$ відповідає найбільш пристосованим генотипам для даного «навколишнього середовища». Нехай кращий зі знайдених генотипів до покоління t позначається ξ_t^* , тобто

$$\xi_t^* = \operatorname{argmax} \{ \Phi(\xi_{i,\tau}), i = 1, \dots, N, \tau = 0, \dots, t \}.$$

Наведемо загальну схему КГА. Використовувані тут імовірнісні оператори

$$Sel : B^n \rightarrow \{1, \dots, N\}$$

$$Cross : B \times B \rightarrow B \times B$$

$$Mut : B \rightarrow B$$

що будуть описані нижче.

Класичний генетичний алгоритм

0. Задати $t := 0$.

1. Поки не заповнена початкова популяція, виконувати кроки 1.1-1.2:

1.1. Побудувати випадковим чином генотип ξ .

1.2. Додати ξ в популяцію Π_0 .

2. Для k від 1 до $N/2$ виконувати кроки 2.1-2.3:

2.1. Селекція: вибрати генотипи $\xi := \xi_{t, Sel(\Pi_t)}$, $\eta := \xi_{t, Sel(\Pi_t)}$.

2.2. Схрещування: побудувати $(\xi', \eta') := Cross(\xi, \eta)$.

2.3. Мутація: $\xi_{2k-1, t+1} := Mut(\xi')$, $\xi_{2k, t+1} := Mut(\eta')$.

3. Перейти до наступної популяції $t := t + 1$.

4. Якщо $t \leq t_{max}$ то перейти на крок 2, інакше - на крок 5.

5. Результатом роботи КГА є найкраще з знайдених рішень $x(\xi_{t_{max}}^*)$.

Пояснимо наведену схему. Для роботи алгоритму необхідна початкова популяція Π_0 , елементи якої генеруються у відповідності з рівномірним розподілом на множині генотипів B , тобто

$$P\{\xi_{i,0,k} = 0\} = P\{\xi_{i,0,k} = 1\} = 1/2, i = 1, \dots, N, k = 1, \dots, l.$$

Імовірнісний оператор селекції особин на просторі популяцій $Sel(\Pi)$ має те ж значення, що і природний відбір. Дія цього оператора полягає у виборі номера батьківської особини для побудувати чергового нащадка. Генотип $\xi_{i,t}$ з номером $i, i = 1, \dots, N$ з популяції Π_t є батьківською особиною при формуванні чергового генотипу $\xi_{k,t+1}$ популяції Π_{t+1} з ймовірністю

$$P_s(i, \Pi_t) = \frac{\Phi(\xi_{i,t})}{\sum_{j=1}^N \Phi(\xi_{j,t})} \quad (2.2)$$

Іншими словами, будь-яка особина $\xi_{i,t}$ популяції Π_t бере участь у створенні наступного покоління відповідно до схеми Бернуллі з N випробувань, де ймовірність успіху дорівнює $P_s(i, \Pi_t)$. При цьому не виключається вибір $\xi_{i,t}$ одночасно в якості ξ і η на кроці 2.1. Описаний оператор Sel іноді також називають селекцією методом рулетки [16, 24]. Припустимо, що колесо рулетки розбите на N секторів, причому сектор i відповідає особі i і має радіанну міру $2\pi P_s(i, \Pi_t)$. Тоді селекцію особини $\xi_{i,t}$ можна подати, як вибір i -го сектора на колесі рулетки.

Селекція називається пропорційною, якщо середнє число нащадків особини пропорційно відношенню її пристосованості до суми пристосованість всіх особин популяції.

Описаний вище оператор селекції здійснює пропорційну селекцію, тому що середнє число нащадків особини $\xi_{i,t}$ дорівнює $\frac{\Phi(\xi_{i,t})}{\sum_{j=1}^N \Phi(\xi_{j,t})}$. Якщо задати функцію $\Phi(\xi_{j,t})$ так, щоб виконувалась умова

$$\sum_{j=1}^N \Phi(\xi_{j,t}) = N \quad (2.3)$$

то пристосованість особини буде дорівнює середньому числу її нащадків. Таким чином, запозичення терміну «пристосованість» з біології цілком

виправдано, в тому окремому випадку, коли виконується рівність (2.3). У загальному випадку значення функції $\Phi(\cdot)$ легко нормувати з допомогою такого масштабу функції $\phi(\cdot)$, яке забезпечувало б співвідношення (2.3) на кожній ітерації КГА, однак, на практиці це не має сенсу.

Далі, для опису КГА необхідно описати двомісний оператор кросинговеру (схрещування) $Cross(\xi, \eta)$ і одномісний оператор мутації $Mut(\xi)$, дія яких носить випадковий характер. Результат кросинговеру (ξ', η') = $Cross(\xi, \eta)$ з ймовірністю P_c формується у вигляді

$$\begin{aligned}\xi' &= (\xi_1, \xi_2, \dots, \xi_\chi, \eta_{\chi+1}, \eta_{\chi+2}, \dots, \eta_l) \\ \eta' &= (\eta_1, \eta_2, \dots, \eta_\chi, \xi_{\chi+1}, \xi_{\chi+2}, \dots, \xi_l)\end{aligned}$$

де випадкова координата схрещування χ обрана з рівномірним розподілом від 1 до $l - 1$. З ймовірністю $1 - P_c$ обидва генотипу зберігаються без змін, тобто $\xi' = \xi$, $\eta' = \eta$. Вплив оператора кросинговеру регулюється параметром P_c . Даний оператор прийнято називати одноточковим кросинговером.

Оператор мутації в кожній позиції генотипу із заданою вірогідністю P_m змінює її вміст. В іншому випадку ген залишається без змін. Таким чином, мутація елементів генотипу відбувається за схемою Бернуллі з ймовірністю успіху P_m . Зміна ймовірностей мутації і кросинговеру дозволяє регулювати роботу КГА і налаштовувати його на конкретні завдання. Збільшення ймовірності мутації до 0.5 перетворює КГА в простий випадковий перебір. Зменшення ж P_m до нуля призводить до малого різноманітності генотипів в популяції і може викликати зациклення КГА на невеликій підмножині рішень. Величини P_c і N також можуть істотно впливати на швидкість збіжності популяції до рішень прийнятної якості. Параметри КГА вибирають, як правило, в таких діапазонах: $0 \leq P_c \leq 1$, $10^{-3} \leq P_m \leq 0.3$, $30 \leq N \leq 1000$.

2.3 Опис генетичного алгоритму розв'язання задачі комівояжера

Розглянемо спосіб алгоритмічного рішення задачі комівояжера [8-9] для побудови туристичного маршруту з використанням генетичного алгоритму. Маршрут буде складатися в системі опорних точок заданого маршруту, в ході якого визначається, наскільки збільшується сумарна відстань сформованого туристичного маршруту, в порівнянні з іншими. Отже, в якості вхідних даних є масив опорних точок, через які повинен пройти туристичний маршрут, а також вказана точка, з якої він буде запусканий. До вхідних даних також відносяться параметри алгоритму, такі, як розмір популяції і відсоток нових особин. Під особоною розуміється будь-який маршрут комівояжера. Він являє собою масив послідовності опорних точок, причому перша і остання точки співпадають - це означає, що початок і завершення маршруту співпадають. Таким чином, особина має розмір, що дорівнює кількості опорних точок + 1. Під популяцією розуміється сукупність всіх особин на кожній ітерації алгоритму. Таким чином, кількість особин завжди дорівнюватиме розміру популяції. Популяція представлена у вигляді масиву. Алгоритм починається з того, що створюється квадратна матриця відстаней, представлена двовимірним масивом розміру n на n , де n - кількість опорних точок в маршруті. В перетину i -го рядка і j -го стовпчика даного масиву записуються відстані між i -й і j -й точками. Далі формується перша популяція, що складається з випадково згенерували особин. Далі обчислюється довжина маршруту для кожної особини. Для цього підсумовуються відстані між опорними точками, що містяться в особини. Відстані беруться з матриці відстаней. Тепер, коли для кожної згенерованої особини відома довжина маршруту, масив популяції сортується від особини з найменшою довжиною до особини з найбільшою. Таким чином, найперша особина в масиві популяції матиме найкоротший маршрут. Однак це буде не найоптимальніший маршрут, а всього лише кращий з перших, що випадково згенерували. Для пошуку найкращого маршруту запускається цикл, який буде працювати до тих пір, поки користувач його не зупинить. У цьому циклі

буде генеруватися нова популяція на основі попередньої їй, потім знову буде проводитися сортування, і так далі. Для генерації нової популяції в циклі проводиться схрещування двох особин, взятих з популяції, сформованої на попередній ітерації, потім наступних двох, і так до тих пір, поки кількість нових особин залишається менше заданого у вхідних параметрах відсотка нових особин. При схрещуванні двох особин виходять також дві особини. Всі нові особини записуються в кінець нового масиву популяції, який є копією поточного. Таким чином, особи з найбільшою довжиною маршруту замінюються новими особинами. Алгоритм, за яким схрещуються особини, називається «частково відображається схрещування». Суть цього алгоритму полягає в наступному:

Нехай дано дві батьківські хромосоми $ch_{1,1} = \{1, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$ і $ch_{1,2} = \{1, 5, 10, 3, 7, 8, 9, 2, 6, 4, 1\}$. Випадковим чином знаходимо дві точки розриву: $p_1 = 3$ і $p_2 = 7$ ($0 < p_1 < p_2 < n$). Вважаємо, що розрив відбувся таким чином: $ch_{1,1} = \{1, 10, 9 \mid 8, 7, 6, 5 \mid 4, 3, 2, 1\}$ і $ch_{1,2} = \{1, 5, 10 \mid 3, 7, 8, 9 \mid 2, 6, 4, 1\}$. Нагадаємо, що позиції 1 і 11 в операції участі не беруть, як ніби реальна хромосома розташована між цими позиціями. При формуванні нащадків $ch_{2,1} = \{1, x, x, \mid x, x, x, x \mid x, x, x, 1\}$ $ch_{2,2} = \{1, x, x, \mid x, x, x, x \mid x, x, x, 1\}$ спочатку проводимо обмін частин, що знаходяться між точками розриву: $ch_{2,1} = \{1, x, x, \mid 3, 7, 8, 9 \mid x, x, x, 1\}$ $ch_{2,2} = \{1, x, x, \mid 8, 7, 6, 5 \mid x, x, x, 1\}$. Далі розставляємо залишилися позиції від відповідних хромосом зліва направо до виникнення конфлікту (номера вершин повторюються в уже сформованій частині хромосоми). Якщо стався конфлікт, то записуємо не конфліктує номер, а номер із сусідньої хромосоми в тій позиції, в якій був знайдений конфлікт в поточній хромосомі. Так триває до повного вирішення конфліктів. Отримуємо: $ch_{2,1} = \{1, 10, 5, \mid 3, 7, 8, 9 \mid 4, 6, 2, 1\}$ $ch_{2,2} = \{1, 9, 10, \mid 8, 7, 6, 5 \mid 2, 3, 4, 1\}$. Однак якщо проводити схрещування двох ідентичних особин, то виявиться, що їх нащадки виявляються ідентичними своїм батькам. У підсумку в популяції буде вже чотири ідентичні особини, а так як наступна популяція формується на підставі попередньої, то рано чи пізно вся нова

популяція буде складатися з ідентичних особин. Щоб цього уникнути, після створення двох нових особин проводиться перевірка на їх ідентичність один одному, і якщо виявляється, що вони ідентичні, то виконується мутація кожної особини. Мутація являє собою перестановку двох випадково обраних позицій в кожній особини.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Формування вхідного математичного опису

В роботі розглядаються туристичні маршрути по Ферганській долині – унікальній території тектонічного походження в Середній Азії площею 22 тис. км², що оточена горами з усіх боків, і має лише один невеликий прохід у вигляді Каракумського водосховища, який відкриває доступ в Ферганську долину з Голодного степу. Тут присутні унікальні природні пам'ятки – гірські хребти Гіссаро-Алая і Тянь-Шань заввишки 6 тис. км², могутні ріки Нарин і Сирдар'я, напівпустельні простори, що змінюються мальовничими альпійськими луками з м'яким кліматом. Крім того, на цій території процвітали великі стародавні міста Середньої Азії з багатонаціональним населенням, тому в Ферганській долині безліч історичних пам'яток, античних руїн і середньовічних пам'ятників, вік яких налічує не одне століття. Наприклад, в Коканде можна відвідати розкішний палацовий комплекс Худояр-хана. Відчуті неймовірну східну атмосферу в повній мірі допоможе відвідування Фергани, Андижани, Маргилана, Шахимардана, Намангана і інших міст Ферганської долини.



a)



б)

Рисунок 3.1 – Ферганська долина а) історичні та б) природні пам'ятки

В роботі було обрано 9 найбільш привабливих з точки зору туризму міст Ферганської долини – міста Маргілан, Коканд, Андижан, Наманган, Кувасай, Киргили, Фергана, Чимион, Вуадиль

Таблиця 3.1 – Квадратна матриця відстаней між точками туристичного маршруту

№	Назва	Маргілан	Коканд	Андижан	Наманган	Кувасай	Киргили	Фергана	Чимион	Вуадиль
1	Маргілан	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

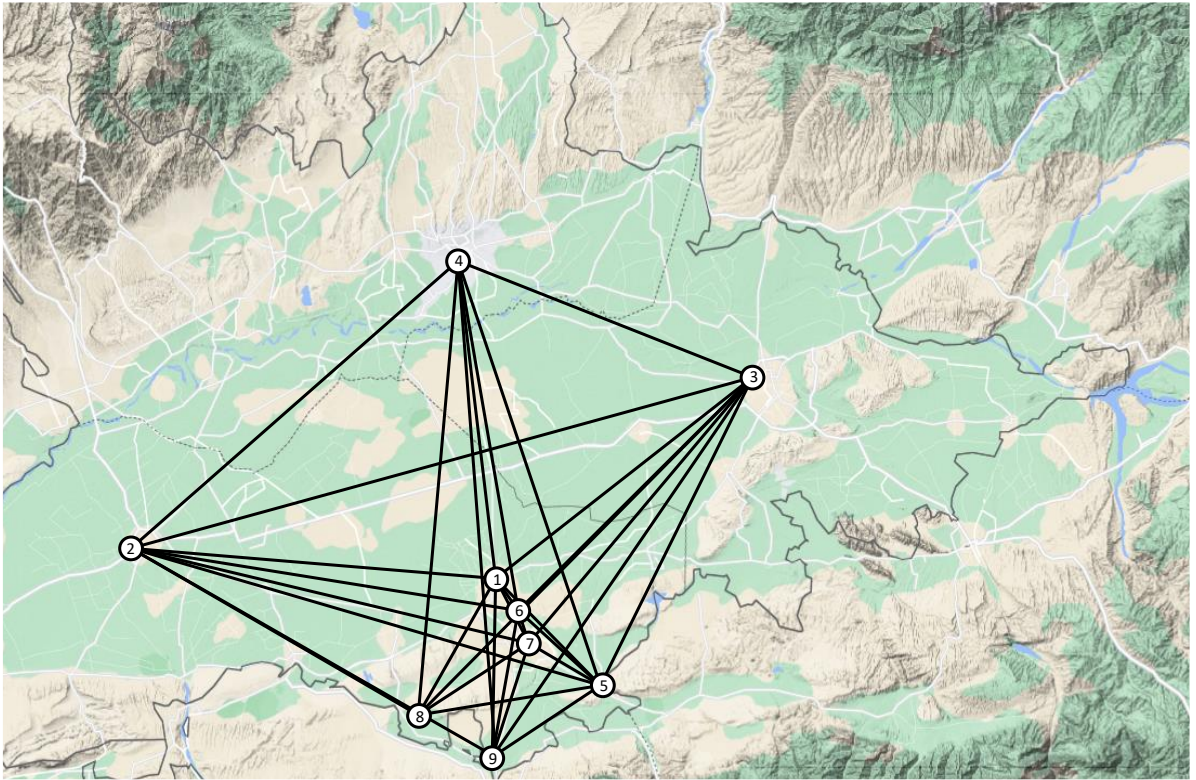


Рисунок 3.2 – Повний граф туристичних маршрутів Ферганською долиною

Таким чином, для подання варіантів багатоденного туристичного маршруту Ферганською долиною у вигляді хромосоми генетичного алгоритму необхідно використовувати одномірний масив з десяти елементів – генів.

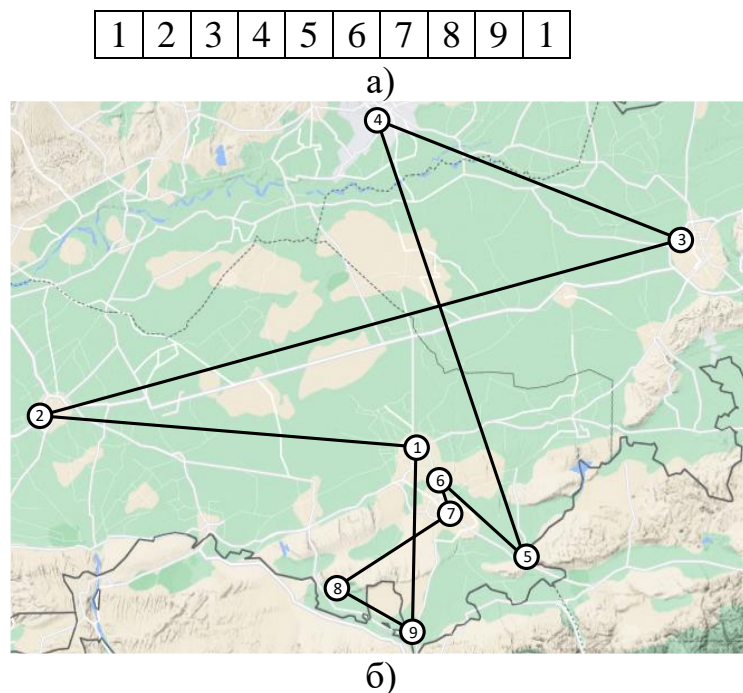


Рисунок 3.3 – Один з варіантів маршруту: а) хромосома б) граф

3.2 Програмна реалізація

Програмна реалізація планувальника виконана в середовищі для наукових і інженерних розрахунків MATLAB [10-13]. MATLAB - потужний засіб автоматизації математичних розрахунків, що відрізняється, перш за все, широким застосуванням матричних операцій. Одна з основних завдань системи MATLAB - надання мови програмування, орієнтованого на технічні та математичні розрахунки, здатного перевершити можливості традиційних мов програмування як по швидкості обчислень, так і по адаптації до вирішення найрізноманітніших завдань. Важливо, що з системою MATLAB можуть інтегруватися такі популярні системи як Mathcad, Maple і Mathematica. Засіб останніх версій Matlab Notebook дозволяє готувати документи в текстовому процесорі Word зі вставками у вигляді результатів обчислень MATLAB, представлених в чисельному, табличному або графічному вигляді

Система являє собою набір m-функцій:

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_conflict.m - вирішення конфліктних ситуацій після схрещування;

popul_gen.m - створення початкової популяції;

points_gen.m - введення опорних точок;

points_show.m - графічне відображення вхідних даних.

Повний код наведено в додатку.

3.3 Аналіз результатів

Популяція згенерована на першому кроці подана в табл. 3.2

Таблиця 3.2 – Початкова популяція

№	Хромосоми										Фітнес функція
1	1	4	3	2	7	6	5	8	9	1	512
2	1	2	4	8	5	9	7	6	3	1	592
3	1	2	9	7	6	4	3	8	5	1	559
4	1	8	4	2	3	6	7	5	9	1	581
5	1	5	8	6	2	3	4	9	7	1	577
6	1	5	2	8	6	7	3	4	9	1	559
7	1	4	3	2	7	6	8	5	9	1	522
8	1	2	8	4	5	7	3	6	9	1	613
9	1	9	6	8	7	4	2	3	5	1	582
10	1	4	2	5	9	8	3	6	7	1	581

Перша і остання точки маршруту співпадають, тобто після проходження туристичного маршруту група повертається в початковий пункт.

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.

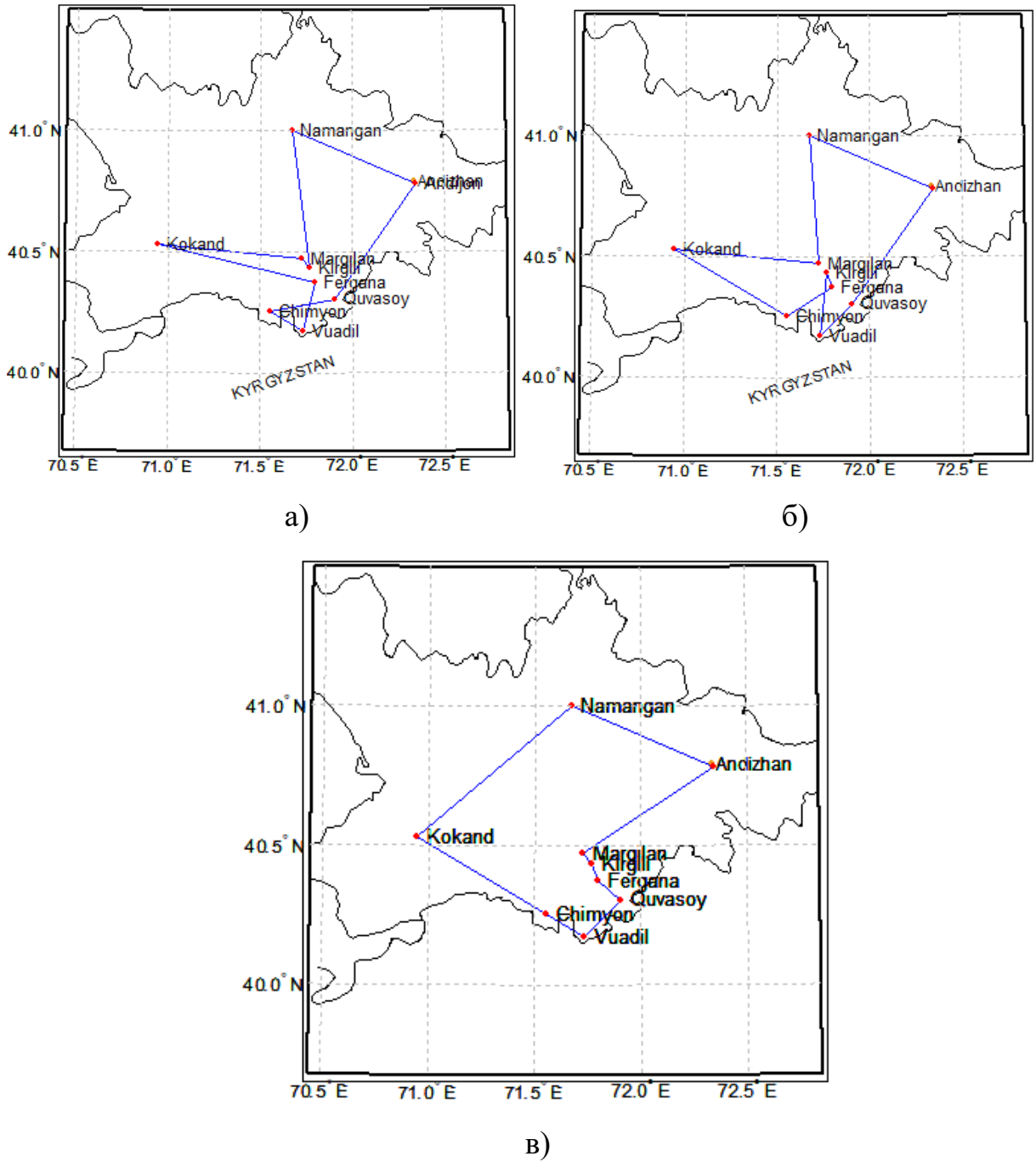


Рисунок 3.4 – Графічне відображення маршрутів сформованих після а) 10 кроків б) 100 кроків в) 1000 кроків роботи генетичного алгоритму

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.

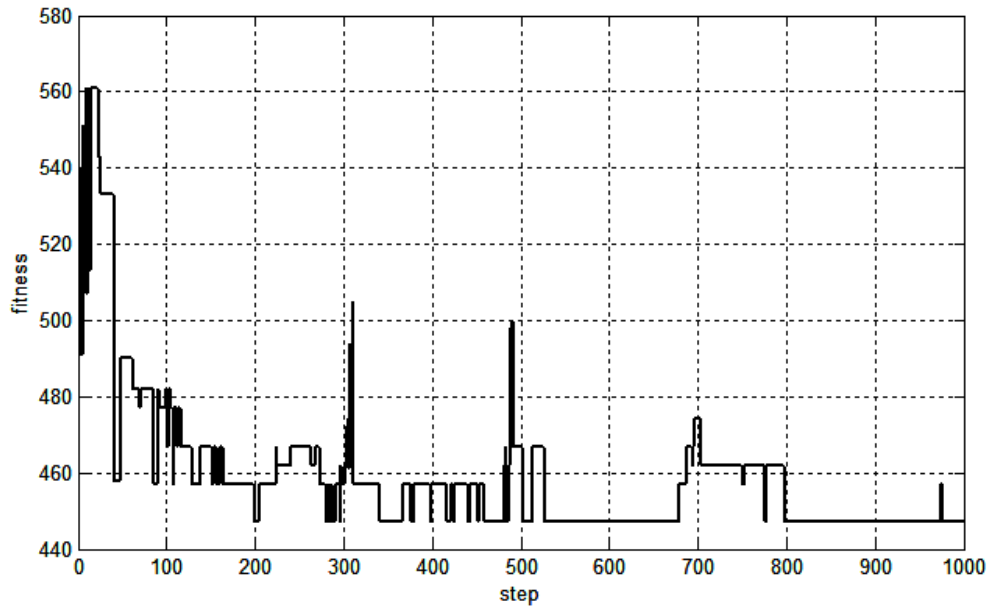


Рисунок 3.5 – Динаміка зміни значень фітнес-функції в ході роботи генетичного алгоритму

Аналіз рис. 3.5 показує, що на 1000 кроці навчання було сформовано популяцію, до якої належить декілька хромосом, що характеризуються фітнес-функцією 447. Графічне відображення відповідних маршрутів наведено на рис.3.6

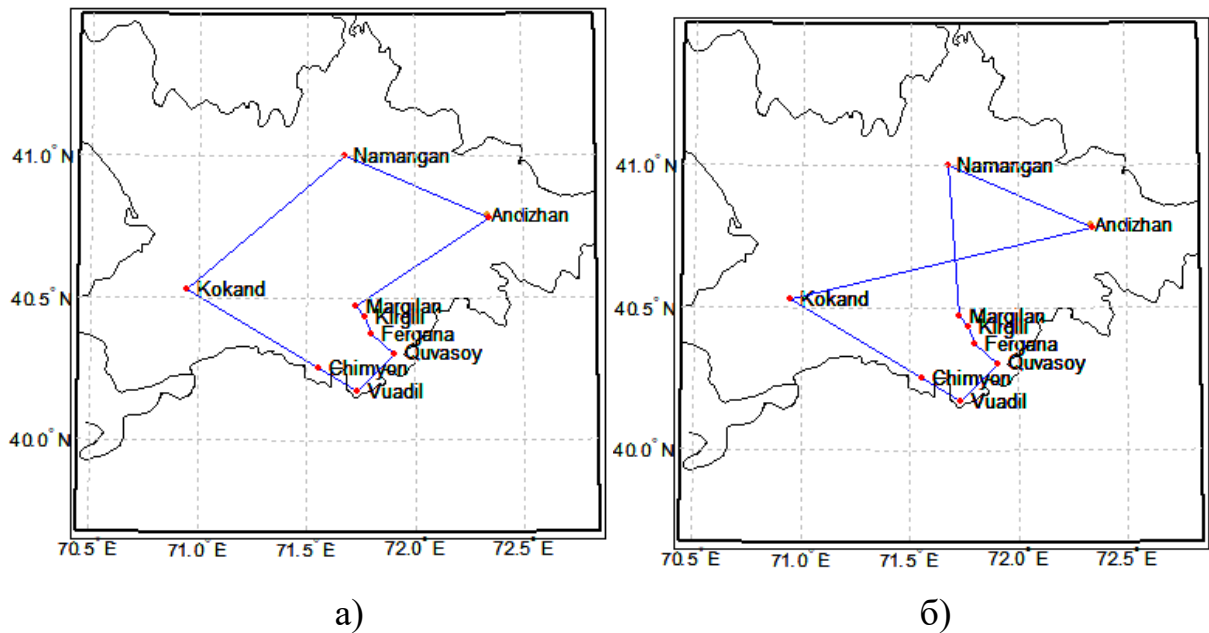


Рисунок 3.6 – Графічне відображення найкоротших маршрутів

ВИСНОВКИ

В випускній роботі було проведено інформаційний синтез інтелектуальної системи, що. система планування туристичних маршрутів. Вибір оптимального маршруту проводився на базі модифікованого генетичного алгоритму шляхом вирішення задачі комівояжера. Програмна реалізація планувальника виконана в середовищі для наукових і інженерних розрахунків MATLAB.

Далі планується провести підбір функціональних параметрів генетичного алгоритму і порівняння його ефективності з іншими алгоритмами вирішення даного завдання.

СПИСОК ЛІТЕРАТУРИ

1. Yalaoui F., Amodeo L., Talbi E.-G. (eds.) Heuristics for Optimization and Learning. – New York: Springer, 2021. – 444 p.
2. Amir H. Gandomi et al. (Eds.) Evolutionary Computation in Scheduling. –Wiley, 2020. – 350 p.
3. Doerr B., Neumann F. (eds.) Theory of Evolutionary Computation. Recent Developments in Discrete Optimization.– Springer Nature Switzerland AG, 2020. – 506 p.
4. Jili Tao, Ridong Zhang, Yong Zhu. A Computing Based Genetic Algorithm: Applications in Industrial Process Modeling and Control. – New York: Springer, 2020. – 280 p.
5. Buontempo F. Genetic Algorithms and Machine Learning for Programmers: Create AI Models and Evolve Solutions. – Pragmatic Bookshelf, 2019. – 234 p.
6. Zhi-Hua Zhou, Yang Yu, Chao Qian Evolutionary Learning. Advances in Theories and Algorithms. – New York: Springer, 2019. – 361 p.
7. Carson J. (Ed.) Genetic Algorithms: Advances in Research and Applications. – New York. USA: Nova Science Publishers, Inc., 2017. – 127 p.
8. Cuevas E., Osuna V., Oliva D. Evolutionary Computation Techniques: A Comparative Perspective. – New York: Springer, 2017. – 236 p.
9. Kramer O. Genetic Algorithm Essentials. – Springer International Publishing AG, 2017. – 94 p.
10. Кравченко І.В., Микитенко В.І. Інформаційні технології. Системи комп'ютерної математики. – Навчальний посібник. — Київ: КПІ ім. Ігоря Сікорського, 2018. – 243 с.
11. Mikhailov E.E. Programming with MATLAB for Scientists: A Beginner's Introduction. – CRC Press, Taylor & Francis Group, LLC, 2017. – 265 p.

12. Jacobson Lee, Kanber Burak. Genetic Algorithms in Java Basics: Solve Classical Problems like. The Travelling Salesman with GA.– Apress Media LLC., 2015. – 162 p.

ДОДАТОК

m-сценарій main.m

```

1- clear;
2- n=9;
3- [p,ns,ds]=points_gen;
4- points_show(p,ns,0.05);
5- popul=popul_gen(n,10);
6- for step=1:2000
7- [fit,p_norm]=popul_fitness(pop,ds);
8- max_fit=max(fit);
9- min_fit=min(fit);
10
11- gr(step)=min_fit;
12- noopt_n=find(fit==max_fit);
13- opt_n=find(fit==min_fit);
14- pop(noopt_n(1))=pop(opt_n(1));
15- opt_path=[1 pop(opt_n(1),:) 1];
16- popul=popul_cross(pop,p_norm);
17- popul=popul_mutation(pop,0.02);
18- end;
19
20- for i=1:n+1;
21- opt_p(i,:)=p(opt_path(i),:);
22- end;
23- hold on
24- linem(opt_p(:,1), opt_p(:,2), 'b-');
25- opt_path
26- gr(end)
27- hold off;

```

m-функція popul_fitness.m

```

1- function [fitness,p_norm]=popul_fitness(popul,points_dist);
2- [k n]=size(popul);
3- for i=1:k
4- fitness(i)= points_dist(1,popul(i,1));
5- for j=1:n-1
6- fitness(i)=fitness(i)+points_dist(popul(i,j),popul(i,j+1));
7- end;
8- fitness(i)= fitness(i)+points_dist(popul(i,n),1);
9- end;
10- fit_max=max(fitness)*1.1;
11- t=(fit_max-fitness)./sum(fit_max-fitness);
12- p_norm(1)=0;
13- for i=1:k-1
14- p_norm(i+1)=p_norm(i)+t(i);
15- end;
16- end;

```

popul_conflict.m

```

1- function [ch]=popul_conflict(ch,n);
2- more_1=[];
3- less_1=[];
4- for i=2:n
5- cht=find(ch==i);
6- col=size(cht,2);
7- if (col>1) more_1(end+1:end+col-1)=cht(2:end); end;
8- if (col<1) less_1(end+1)=i;end;
9- end;
10- ch(more_1)=less_1;
11- end;

```


m-функція popul_mutation.m

```

1 function [new_popul]=popul_mutation(popul,p_mutation);
2     [k n]=size(popul);
3     t=zeros(1,1);
4     for i=1:k
5         for j=1:n
6             if rand(1,1)<p_mutation
7                 tn=round(rand(1,1)*(n-1))+1;
8                 t=popul(i,tn);
9                 popul(i,tn)=popul(i,j);
10                popul(i,j)=t;
11            end;
12        end;
13    end;
14    new_popul=popul;
15 end;

```

popul_cross.m

```

1 function [new_popul]=popul_cross(popul,p_norm);
2     [k n]=size(popul);
3     new_popul=[];
4     for j=1:round((k-1)/2)
5         pt=rand(1,2);
6         pt1=find(p_norm<=pt(1));
7         pt2=find(p_norm<=pt(2));
8         parents=[pt1(end) pt2(end)];
9
10        p=round(rand(1,2)*(n-1))+1;
11        if p(1)>p(2)
12            t=p(1);
13            p(1)=p(2);
14            p(2)=t;
15        end;
16        ch1=popul(parents(1),:);
17        ch1(p(1):p(2))=popul(parents(2),p(1):p(2));
18        ch1=popul_conflict(ch1,n+1);
19
20        ch2=popul(parents(2),:);
21        ch2(p(1):p(2))=popul(parents(1),p(1):p(2));
22        ch2=popul_conflict(ch2,n+1);
23
24        new_popul=[new_popul; ch1; ch2];
25    end;
26 end;

```

popul_gen.m

```

1 function [popul]=popul_gen(n,popul_size);
2 for k=1:popul_size
3     ch=2:n;
4     for i=1:n
5         j=round(rand(1,2)*(n-2))+1;
6         t=ch(j(1));
7         ch(j(1))=ch(j(2));
8         ch(j(2))=t;
9     end;
10    popul(k,:)=ch;
11 end;
12 end;

```

points_gen.m - введення опорних точок;

```

1 function [points,names,distances]=points_gen()
2 points=[
3     40.47240, 71.7246;
4     40.52861, 70.9425;
5     40.78210, 72.3442;
6     40.99830, 71.6726;
7     40.30538, 71.9079;
8     40.43583, 71.76722;
9     40.37338, 71.79783;
10    40.2526512, 71.5575586;
11    40.1742, 71.7301
12 ];
13 names=[
14     'Margilan';
15     'Kokand  ';
16     'Andijon  ';
17     'Namangan';
18     'Quvasoy  ';
19     'Kirgili  ';
20     'Fergana  ';
21     'Chimyon  ';
22     'Vuadil  '
23 ];
24 distances=[
25     0  77  80  67  33  9  15  37  42;
26     77  0  132 119 113 99 84 73 103;
27     80 132 0  68  80 81 79 109 110;
28     67 119 68 0  102 76 80 106 111;
29     33 113 80 102 0  25 21 53 36;
30     9  99 81 76 25 0  7  33 32;
31     15 84 79 80 21 7  0  31 33;
32     37 73 109 106 53 33 31 0  34;
33     42 103 110 111 36 32 33 34 0];

```

points_show.m - графічне відображення вхідних даних.

```

1 function points_show(points,names,eps);
2 lims=minmax(points');
3 lims(:,1)=lims(:,1)-eps*10;
4 lims(:,2)=lims(:,2)+eps*10;
5
6 worldmap('hi',lims(1,:),lims(2,:));
7 for j=1:size(points,1)
8     textm(points(j,1), points(j,2)+eps, names(j,:));
9     linem(points(j,1), points(j,2), 'r. ');
10 end;
11
12 end;

```