

Олександра Стан¹, Тетяна Марусенкова², Ірина Юрчак³

¹ Кафедра програмного забезпечення, Національний університет “Львівська політехніка”, вул. Степана Бандери 12, Львів, Україна, E-mail: oleksandra.stan.mpzip.2022@lpnu.ua, ORCID 0009-0000-7010-8701,

² Кафедра програмного забезпечення, Національний університет “Львівська політехніка”, вул. Степана Бандери 12, Львів, Україна, E-mail: tetiana.a.marusenkova@lpnu.ua, ORCID 0000-0003-4508-5725,

³ Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. Степана Бандери 12, Львів, Україна, E-mail: iryna.y.yurchak@lpnu.ua, ORCID 0009-0005-9100-8511

КОМБІНОВАНИЙ ПІДХІД ДЛЯ ПОБУДОВИ ОПТИМАЛЬНОГО ІНДИВІДУАЛЬНОГО ТУРИСТИЧНОГО МАРШРУТУ У МОБІЛЬНОМУ ЗАСТОСУНКУ

Отримано: лютий 17, 2024 / Переглянуто: червень 12, 2024 / Прийнято: липень 02, 2024

© Стан О., Марусенкова Т., Юрчак І., 2024

<https://doi.org/>

Анотація. Стаття присвячена вирішенню задачі побудови оптимальних маршрутів при плануванні індивідуальних подорожей в умовах впливу багатьох факторів і можливих змін вхідних параметрів (погодних умов, заторів на дорогах тощо). Проаналізовано чотири класи алгоритмів для розв'язання задачі комівояжера та оцінено їхню доцільність для використання у мобільному туристичному застосунку. Форма мобільного застосунку продиктована тим, що туристи переважно не беруть у мандрі техніку, важчу за смартфон. Показано недоліки евристичних і метаевристичних методів, зокрема, високу чутливість до початкових налаштувань, негарантовані оптимальні розв'язки та ризик зациклення в локальних оптимумах. Відкинуті точні методи як непридатні в контексті поставленої задачі внаслідок високої обчислювальної складності. На підставі проведених досліджень запропоновано комбінований підхід, при якому в якості глобальної стратегії використовується генетичний алгоритм, а для покращення знайдених розв'язків – алгоритм локального пошуку в його чотирьох варіаціях (Relocation, 2-opt, 3-permute і Link swap). Описано архітектуру та технологічний стек розробленого мобільного застосунку. Окреслено перспективу подальших досліджень, зокрема, пошук рішення задачі групового комівояжера з можливістю спільного редагування плану подорожі усіма учасниками групи мандрівників та задачі багатоагентної маршрутизації.

Ключові слова: планування подорожей, оптимальний маршрут, генетичний алгоритм, локальний пошук, евристичний алгоритм, метаевристичний алгоритм, задача комівояжера, мобільний застосунок.

Вступ

Від якості надання туристичних послуг залежить рівень задоволення туристичних потреб та ефективність використання матеріальних, часових і людських ресурсів. Ступінь застосування ІТ-технологій у галузі туризму сьогодні є доволі високим. Зокрема, існує низка глобальних систем бронювання (авто-, авіа- та залізничних квитків, житла, автомобілів тощо); інформаційних порталів з актуальною довідковою інформацією про місце призначення (тарифи, ціни, курси валют, прогнози погоди тощо); мультимедійних електронних путівників, які дозволяють віртуально подорожувати запропонованими маршрутами (зокрема, віртуальних музейних турів); картографічних сервісів (з можливістю орієнтуватися на місцевості, прокладати піші та транспортні маршрути, оцінювати відстані, знаходити розташовані поблизу визначні місця, заклади харчування, магазини, зупинки

транспорту тощо навіть у чужому місті) та ін. [1].

Спостерігається активне впровадження рекомендаційних систем, які використовують методи інтелектуального аналізу даних і пошуку інформації для передбачення, наскільки продукт відповідає потребам і бажанням конкретного користувача. Такі системи застосовуються у туристичних мобільних і веб-застосунках для рекомендації туристичних атракцій, авіарейсів і місць відпочинку та навіть комбінування рекомендацій в один пакет, що може бути зручно для планування відпусток (наприклад, система рекомендує і житло в певній місцевості, і події, заплановані в ній на термін відвідування) [2]. Переважно застосовується колаборативна фільтрація – техніка фільтрування елементів, що потенційно можуть сподобатися користувачеві, на основі реакції користувачів зі схожими персональними характеристиками. Зокрема, у [3] представлено приклад застосування рекомендаційної системи, яка враховує особисті обмеження туристів, включаючи ті, що обумовлені станом здоров'я, релігійні або етичні вимоги та особисті вподобання та інтереси. До недоліків запропонованої системи відносяться мала кількість залучених учасників і недосконалий процес збору обмежень від туристів.

Іншим підходом є реалізація системи відповідей на запитання. Такі системи здатні транслювати текст запитань, заданих природною людською мовою, у текст запитів, що можуть бути розпізнані комп'ютером. Результат обробки запиту повертається у вигляді лаконічної і точної відповіді людською мовою. Добре спроектована система на основі знань у сфері туризму дозволяє швидко отримати інформацію, пов'язану з подорожами, і надати користувачеві технічну підтримку для укладання індивідуальних планів подорожей за текстом запиту [4]. У [5] показано проектування інтелектуальної системи відповідей на запитання для прокладення туристичного маршруту. Система заснована на графі знань про туризм, ядро якого включає модель розпізнавання іменованих сутностей на основі Bert-BiLSTM-CRF (для екстракції сутностей із питань) та шаблонну модель класифікації на алгоритмі NB (для зіставлення запитання з існуючим шаблоном для отримання зв'язку сутностей). Система застосовна для багатьох сценаріїв подорожей. Однак, шаблони для класифікації фахівці здебільшого будують вручну, що тягне за собою затрати часу та зусиль і суттєво збільшує вартість супроводу системи.

Не повністю розв'язаними залишаються задачі ефективного планування маршрутів для індивідуальних туристичних подорожей в умовах часових, просторових, бюджетних, транспортних, сезонних, карантинних і інших обмежень. Групові поїздки, що пропонуються туроператорами, поступаються індивідуальним у зручності і не враховують персональних уподобань туриста. Натомість побудова оптимального індивідуального туристичного маршруту потребує аналізу великого обсягу даних, що при здійсненні вручну вимагає значних затрат часу і зусиль. Водночас розвиток технологій, зокрема штучного інтелекту, уможливує автоматизацію цієї задачі.

Складність проблеми зростає зі збільшенням числа факторів, що беруться до уваги при плануванні подорожей. Процес оптимізації ускладнює також і число комбінацій різноманітних факторів (доступних транспортних засобів, наявного часу, числа самих туристичних об'єктів для відвідування тощо). Також слід враховувати, що під час подорожі можуть виникати несподівані обставини (наприклад, раптове погіршення погодних умов або утворення заторів на дорогах). Отже, повинна залишатися можливість оперативного скоригувати плани подорожі в залежності від зміни обставин.

Огляд сучасних джерел інформації за тематикою публікації

Світові тенденції вирішення проблеми побудови оптимальних туристичних маршрутів передбачають використання машинного навчання, штучного інтелекту [6] та оптимізаційних алгоритмів для укладання індивідуальних туристичних маршрутів.

Існує низка типових задач оптимізації туристичних маршрутів, які можуть бути вирішені за допомогою оптимізаційних методів. До таких задач належать:

1. Задача комівояжера (Traveling Salesman Problem): полягає у знаходженні маршруту, що дозволяє обійти всі пункти призначення та повернутися в початкову точку, витративши найменше

часу або пройшовши найкоротшу відстань.

2. Задача групового комівояжера (Generalized Traveling Salesman Problem): відрізняється від TSP тим, що потрібно відвідати по одному вузлу в кластері, при чому кластери слід обійти послідовно.

3. Задача з обмеженими ресурсами (Resource-Constrained Routing Problem): полягає у пошуку маршруту, що максимізує ефективність планування і при цьому задовольняє всі обмеження, які накладаються на ресурси (фінансові, транспортні, часові тощо).

4. Задача маршруту з відвідуванням (Route with Visitation Problem): полягає у пошуку оптимального маршруту, за яким можна обійти всі обов'язкові пункти призначення (у цій модифікації задачі одні пункти важливіші за інші і є пріоритетними для відвідування).

5. Задача маршруту з орієнтованими часовими вікнами (Time-Dependent Routing Problem): полягає у пошуку найкоротшого за часом маршруту з урахуванням часу доби та ситуації на дорогах.

6. Задача географічного обмеження (Geographical Constraint Problem): полягає у пошуку оптимального маршруту, що задовольняє географічні обмеження (політичні межі, клімат, погодні умови тощо).

7. Задача багатоагентної маршрутизації (Multi-Agent Routing Problem): полягає в знаходженні оптимального розподілу маршрутів і ресурсів з-поміж групи осіб (агентів), вподобання та інтереси яких не збігаються.

У [7] проблема створення оптимального плану туру передбачає використання алгоритму рюкзака 0-1. У [8] розглянуто алгоритми, що найчастіше використовуються в туристичних інтелектуальних системах. Виявлено, що найчастіше застосовується генетичний алгоритм і експертні системи, що базуються на правилах. Недоліком таких систем є необхідність модифікувати правила щоразу при внесенні змін в область знань, а обмеженням генетичного алгоритму – чутливість до початкових параметрів. У [9] генетичний алгоритм застосований для планування маршруту, що охоплює фіксоване число (47) визначних місць в Чунцині, причому враховується фактор вартості транспортування. У [10] розглянуто метод планування подорожі з використанням алгоритму мурашиних колоній. Перевагами цього методу є вища відповідність потребам туриста та вищий ступінь врахування схожості з минулими туристичними поїздками, у порівнянні з базовими методами. Однак, він чутливий до параметрів та демонструє повільну збіжність у задачах з величезними графами. Представлений у [11] покращений алгоритм оптимізації мурашиних колоній для планування подорожей передбачає збір траєкторії подорожей туристів із сигнальних даних їхніх телефонів. Він володіє кращою адаптованістю, масштабованістю для складних задач і вищою точністю за рахунок збільшення обчислювальної складності і підвищеної чутливості до параметрів.

Постановка задачі

Об'єктом дослідження є процес автоматизації планування індивідуальних туристичних маршрутів. Предметом дослідження є алгоритми оптимізації для побудови індивідуальних туристичних маршрутів. Метою статті є аналіз доцільних підходів і алгоритмів побудови персоналізованих туристичних маршрутів за обраними критеріями, що можуть бути реалізовані у мобільному застосунку. Вимога щодо мобільного застосунку зумовлена тим, що лише невелика частка туристів подорожують з ноутбуками чи комп'ютерами, але практично всі – зі смартфонами.

Виклад основного матеріалу

Усі алгоритми для вирішення задачі комівояжера можна поділити на 4 класи: евристичні, метаввристичні, точні та комбіновані.

Точні методи (динамічне програмування, лінеаризація та ін.) намагаються знайти точний оптимальний розв'язок задачі, якщо він існує. Базуючись на математичних моделях і алгоритмах, методи цього класу гарантують знаходження найкращого розв'язку, але зазвичай вимагають

значних обчислювальних ресурсів і часу, і тому є ефективними лише для задач на графах з невеликою кількістю вершин. Також вони можуть слугувати еталоном для оцінки евристичних і метаевристичних методів.

Натомість, евристичні алгоритми не використовують точних математичних методів, а базуються радше на досвіді, інтуїції та загальних правилах. Вони є простими у реалізації та здатні надавати швидкі та доволі наближені розв'язки, але не гарантують знаходження оптимального розв'язку. Для розв'язання задачі комівояжера найчастіше застосовують алгоритм вставки, метод найближчого сусіда та алгоритм локального пошуку (2-opt).

Метаевристичні алгоритми для розв'язання складних оптимізаційних задач емулюють природні процеси (наприклад, еволюцію чи поведінку зграї тварин, комах або птахів), застосовуючи стохастичні підходи. Оптимальність знайденого розв'язку при цьому не гарантується. Тим не менш, на практиці вони здатні знаходити вельми наближені розв'язки у складних задачах. Представники цього класу, що застосовуються для вирішення задачі комівояжера, – генетичні алгоритми, алгоритми зграї частинок, мурашині алгоритми.

Комбіновані методи передбачають поєднання переваг методів різних класів. Наприклад, евристичний алгоритм є доцільним для швидкої локалізації оптимального розв'язку, а детальне уточнення цього розв'язку краще досягати точними методами.

Доцільність вибору підходу до розв'язання задачі комівояжера залежить від розмірності задачі, заданої точності розв'язку, наявних обчислювальних ресурсів, а також часових обмежень.

Генетичні алгоритми, як підказує їхня назва, моделюють процес природного відбору. Здатні адаптуватися до змін у навколишньому світі виживають, розмножуються і продовжують життя в наступному поколінні. Перевагами генетичного алгоритму є: застосовність до широкого спектру оптимізаційних задач різної складності, можливість генерації різноманітних розв'язків завдяки схрещуванню та мутації, а також відносна простота реалізації. Однак, пошук оптимального маршруту генетичним алгоритмом може потребувати значних затрат часу, особливо за наявності багатьох індивідів і поколінь у популяції. Ефективність роботи алгоритму істотно залежить від початкового налаштування параметрів. Крім того, може трапитися застрягання у локальних оптимумах, особливо при великій кількості вершин у графі. Таким чином, незважаючи на поширеність і універсальність генетичних алгоритмів, вони є сумнівним вибором, принаймні у чистому вигляді, для вирішення задачі комівояжера, оскільки їхня ефективність залежить від специфіки задачі та вдалого налаштування параметрів.

Алгоритми оптимізації мурашиної колонії для вирішення оптимізаційних задач відтворюють поведінку мурах. Основними перевагами цих алгоритмів є здатність працювати з недиференційованими функціями, простота реалізації і можливість розпаралелення. Недоліками є складність налаштування параметрів (незначна зміна параметрів призводить до зовсім інших результатів), потреба у значних обчислювальних ресурсах, особливо при роботі з великою кількістю міст, для досягнення швидкої збіжності, перевантаження феромонними слідами та можливість заиклення в локальних оптимумах. У кожній конкретній задачі ефективність алгоритму мурашиної колонії залежатиме від її характеристик, параметрів алгоритму та контексту використання.

Алгоритми локального пошуку, що належать до класу евристичних алгоритмів, використовуються, коли потрібен сам розв'язок, а не шлях до нього. Локальний пошук використовується в більшості моделей штучного інтелекту для пошуку оптимальних розв'язків згідно з функцією вартості моделі.

Пошук кращого рішення стартує з деякого початкового розв'язку, який послідовно уточнюють за рахунок внесення локальних змін. Алгоритм локального пошуку продовжує пошуки у межах обмеженого околу поточного розв'язку. Якщо вдасться знайти розв'язок, кращий за поточний, то щойно знайдений розв'язок стане новим поточним. Процес пошуку зупиняється при досягненні заданої умови. Такою умовою може бути досягнення локального оптимуму або ж повне використання усіх виділених ресурсів. Задача знаходження рішення цілком покладається на

оператори локального пошуку. Пошуковий підхід оператора локального пошуку може бути одним із декількох типів [12]: переміщення, 2-opt, 3-permute та link swap.

Переміщення є процесом, при якому вибраний вузол, що називається ціллю, переміщується з поточної локації в нову позицію (пункт призначення). Тобто, розміщення вибраного вузла (цілі) змінюється. Зміна порядку вузлів потенційно призводить до покращених розв'язків.

Алгоритм 2-opt (two-optimization) намагається поліпшити наявний маршрут за рахунок обміну двох ребер, які утворюють цикл. Початковий маршрут вибирається випадковим чином або ж будується якимось стороннім алгоритмом. На кожній ітерації алгоритм 2-opt вибирає пару ребер для обміну. Якщо модифікований маршрут є кращим за поточний, він стає новим поточним. Процес зупиняється, коли стає істинною задана умова. Типова умова зупинки – нові ітерації не дають поліпшення маршруту.

Алгоритм 3-permute використовує перестановки трьох вузлів. Усього таких перестановок може бути шість. Таким чином, маючи вузли ABC, можемо згенерувати інші п'ять перестановок, ACB, BAC, BCA, CAB і CBA, і «приміряти» їх в надії знайти кращий розв'язок.

Алгоритм локального пошуку на основі обміну посиланнями (link swap) міняє місцями пару послідовних ребер і перевіряє, чи модифіковане рішення є кращим за поточне.

Алгоритм локального пошуку в усіх його варіаціях швидкий і простий у розумінні та реалізації. Недоліки алгоритму включають залежність від початкових умов, ймовірність застрягання в локальному оптимумі та локальність рішень.

Для вирішення поставленої задачі оптимізації туристичних маршрутів нами вибрано комбінацію генетичного алгоритму з алгоритмом локального пошуку, що є представниками класів евристичних і метаевристичних алгоритмів. Обґрунтуванням такого вибору є наступні аргументи:

- Генетичний алгоритм веде пошук у глобальному просторі і може використовуватися для пошуку приблизних розв'язків, які згодом підлягають уточненню алгоритмом локального пошуку.
- Перевагами генетичного алгоритму є гнучкість і адаптивність. У комбінації з алгоритмом локального пошуку генетичний алгоритм здатний до адаптації до змін у просторі рішень і характеристик вирішуваної задачі.
- Генетичний алгоритм забезпечує швидкість знаходження приблизного розв'язку, в околі якого алгоритм локального пошуку досягає точності поліпшеного розв'язку.

Вибрані два методи можна комбінувати у кілька способів. Найкраще використовувати генетичний алгоритм для пошуку глобального оптимуму, а алгоритм локального пошуку – для уточнення приблизного розв'язку в його вузькому околі.

Найбільш доцільним є багатократне застосування алгоритму локального пошуку з генетичним алгоритмом на різних етапах виконання останнього (при створенні початкового маршруту, при мутаціях, при еволюції і після неї). Генетичний алгоритм створює початкову популяцію (початковий маршрут). Алгоритм локального пошуку поліпшує отримані індивіди. Таким чином, навіть початковий розв'язок є доволі наближеним до оптимального, що дозволяє компенсувати чутливість до вибору початкового розв'язку. Алгоритм локального пошуку поліпшує індивіди поточної популяції також і під час еволюції, перед вибором нового покоління, щоб лише найкращі могли перейти до наступного етапу. Коли вже одержаний глобальний оптимум (після завершення еволюції), алгоритм локального пошуку застосовується ще раз для остаточного вдосконалення знайденого маршруту. Зрештою, алгоритм локального пошуку опційно використовується на проміжних етапах генетичного алгоритму або поза ними, зокрема, під час мутацій чи при побудові нових індивідів.

Описаний спосіб комбінування генетичного алгоритму з алгоритмом локального пошуку сприяє балансуванню глобального та локального аспектів оптимізації планування туристичних маршрутів.

Результати та обговорення

Основна функція оптимізаційного алгоритму виглядає наступним чином:

```
async function optimizeRouteWithGoogleAPI(initial route, origin, destination, waypoints) {
  try {
    // Отримання потенційних рішень за допомогою генетичного алгоритму
    const potentialSolutions = await runGeneticAlgorithm(initialRoute);
    // Паралельне виконання
    const lsPromises = potentialSolutions.map((solution) => runLocalSearch(solution));
    // Отримуємо проміжні результати
    const optimizedRoutes = await Promise.all(lsPromises);
    // Знаходимо кращий маршрут
    const bestRoute = findBestRoute(optimizedRoutes);
    return bestRoute;
  } else {
    throw new Error(`Error optimizing route: unexpected response format`);
  }
} catch (error) {
  throw new Error(`Error optimizing route: ${error.message}`);
}
```

Ефективність комбінованого підходу залежить від налаштування початкових параметрів для генетичного алгоритму та вдало заданого критерію завершення пошуку для алгоритму локального пошуку. У цьому випадку застосовується число послідовних ітерацій без поліпшення розв'язків:

```
async function findBestRoute(routes) {
  let bestRoute = routes[0];
  let bestDistance = await calculateTotalDistance(bestRoute);
  const maxConsecutiveNoImprovement = END_CRITERIA_ITERATIONS_COUNT;
  let consecutiveNoImprovement = 0;
  for (const route of routes.slice(1)) {
    const distance = await calculateTotalDistance(route);
    if (distance < bestDistance) {
      bestRoute = route;
      bestDistance = distance;
      consecutiveNoImprovement = 0;
    } else {
      consecutiveNoImprovement++;
      if (consecutiveNoImprovement >= maxConsecutiveNoImprovement) {
        break;
      }
    }
  }
  return bestRoute;
}
```

Запропонований підхід реалізований у мобільному застосунку для планування індивідуальних подорожей Львовом. Мобільний застосунок розроблений за клієнт-серверною архітектурою. Для клієнтської частини обрано технологію React Native (завдяки кросплатформності), для стилізації – Tailwind CSS, для серверної частини – серверне середовище Node.js, веб-фреймворк Express.js. Взаємодія між серверною та клієнтською частинами відбувається за допомогою REST API. Для керування базою даних використовується PostgreSQL, а для маніпуляцій даними з бази даних – мова запитів SQL. Доступ до бази даних здійснюється за допомогою бібліотеки Node-postgres. На рис. 1 представлено діаграму сутностей.

Для опрацювання географічних даних (карт, координат, маршрутів) застосовано Google Maps API, а саме компоненти Google Maps JavaScript API, Google Maps Android API, Google Maps iOS SDK, Google Places API і Google Distance Matrix API. Google Places API дозволяє отримувати інформацію про локації (ресторани, готелі тощо), а Google Distance Matrix API – про відстань та час подорожі між локаціями. Інтеграція у застосунок та візуалізація карт Google реалізовані за допомогою елемента `MapView`. Щоб користувачеві було легше вводити адреси та інші локації на карті, в застосунку реалізоване автозаповнення пошукових полів з допомогою функціоналу `GooglePlacesAutoComplete`.

Для користування застосунком необхідна автентифікація. Імена користувачів унікальні. Паролі зберігаються у базі даних і зашифровуються алгоритмом Blowfish. Визначено дві ролі – турист і гід.

Тестування надсилання запитів серверу здійснювалося з допомогою інструмента Katalon Studio v9.

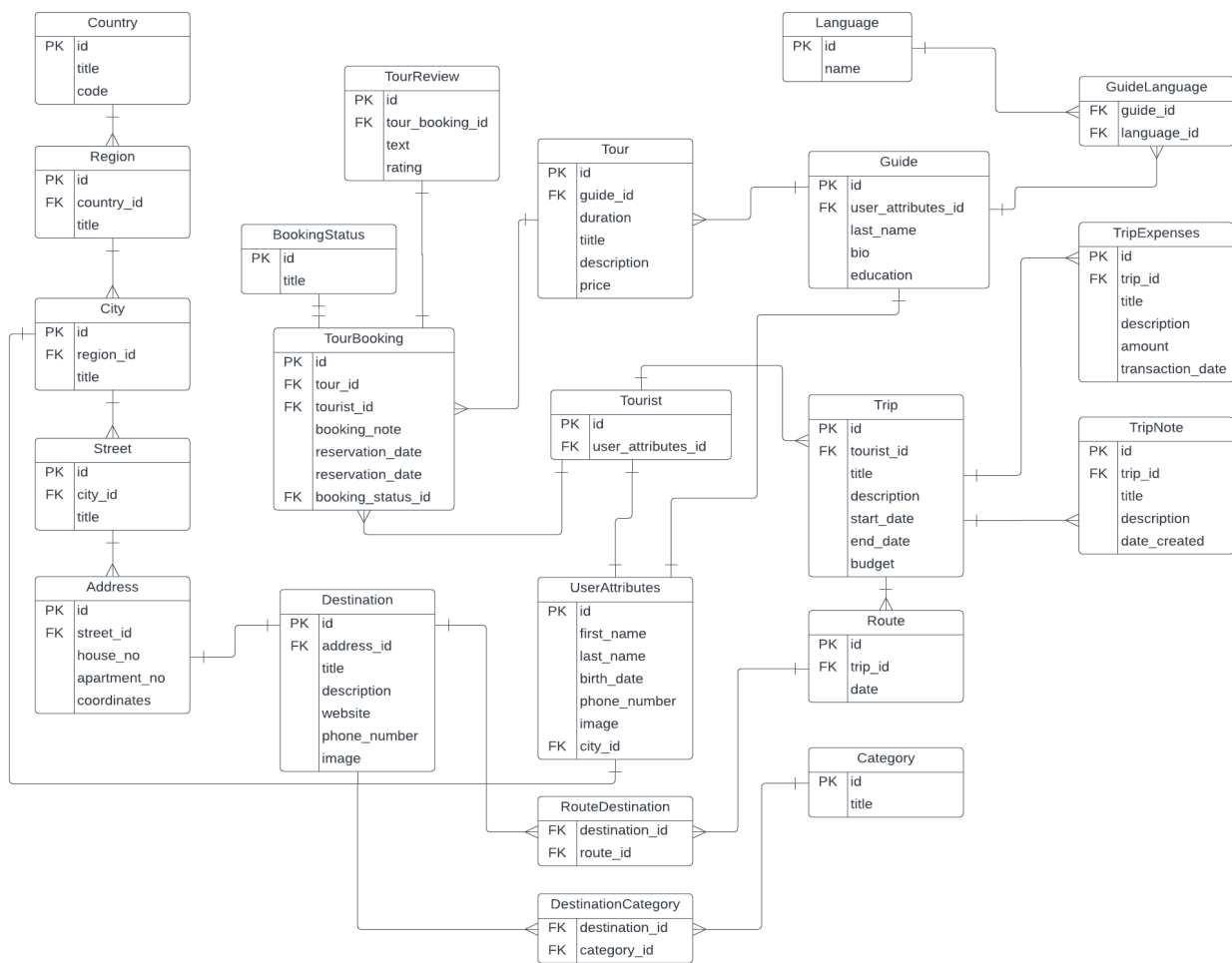


Рис. 1. Діаграма сутностей

Мобільний застосунок успішно пройшов тестування на операційних системах Android v13.0 та iOS v17.1.1. Встановлені значення параметрів генетичного алгоритму для верифікації роботи розробленого підходу: розмір популяції: 100, кількість поколінь: 100, ймовірність схрещення: 0.7, ймовірність мутації: 0.02, елітизм: встановлений. Алгоритм локального пошуку завершує роботу, якщо за останні десять ітерацій не досягнуто поліпшення розв'язків.

Висновки

У даній статті проаналізовано підходи та алгоритми для вирішення задачі комівояжера в контексті планування індивідуальних туристичних маршрутів у мобільному застосунку з використанням карт Google. Розглянуто основні проблеми, що впливають під час вирішення задачі оптимізації туристичних маршрутів: наявність багатьох альтернативних маршрутів, брак даних, обмежені фінансові, часові, транспортні та інші ресурси та зміна обставин.

Досліджено комбінацію евристичних і метаевристичних методів. Запропоновано застосовувати генетичний алгоритм для глобального пошуку приблизних розв'язків, а алгоритм локального пошуку для їх подальшого уточнення, що дозволяє вирішувати задачу комівояжера більш ефективно та збалансовано.

Розроблений мобільний застосунок може слугувати платформою для порівняльного аналізу та тестування масштабованості алгоритмів оптимізації, а також точного налаштування параметрів цих алгоритмів. Також він полегшує збір і аналіз даних, використовуючи інформацію користувача для дослідження ринку, аналізу поведінки та вдосконалення систем рекомендацій.

Користувачів, які бажають обійтися без допомоги гіда, могла б зацікавити можливість придбання туристичного путівника, в якому запропонований готовий маршрут з детальними поясненнями. Таким чином, користувач замість створювати власний маршрут міг би використати вже готовий із заздалегідь визначеними локаціями та інформацією про них.

Оскільки туристи часто мандрують з родиною чи друзями, корисною є опція запрошення друзів, родичів і знайомих до кооперації у плануванні подорожі. Усі учасники мали б отримати доступ до загального плану поїздки і вносити в нього корективи.

Удосконалення потребують також і механізми захисту персональних даних користувачів розробленого мобільного застосунку. Реєстрація та вхід можуть бути одночасно покращені та спрощені за рахунок використання облікових записів соціальних мереж (Google, Facebook, X).

Перелік використаних джерел

- [1] Z. Malcienė, L. Skauronė, “Application of Information Systems in Tourism and Leisure Sector”, *Int. Jou. Soc. Hum. Inve*, Iss. 6, No 2, pp. 5341–5346, Feb. 2019. <https://doi.org/10.18535/ijsshi/v6i2.11>
- [2] F. Ricci, “Recommender Systems in Tourism”, in *Handbook of e-Tourism*, Cham, Germany, Springer, 2022, pp. 457–474. https://doi.org/10.1007/978-3-030-48652-5_26
- [3] J. Li, Z. Luo, H. Huang, Z. Ding, “Towards Knowledge-Based Tourism Chinese Question Answering System”, *Mathematics*, Iss. 10, No. 4, p. 664, 2022. <https://doi.org/10.3390/math10040664>
- [4] B. Ojokoh, “A Review of Question Answering Systems”, *J. of Web Eng.*, Iss. 17, No. 8, pp. 717–758, Jan. 2019.
- [5] Y. Sui, “Question Answering System Based on Tourism Knowledge Graph”, in *J. Phys. Conf. Ser.*, Wuhan, China, p. 012064, Mar. 2021. <https://doi.org/10.1088/1742-6596/1883/1/012064>
- [6] J. A. Orama, A. Huertas, J. Borràs, A. Moreno, S. Clavé, “Identification of Mobility Patterns of Clusters of City Visitors: An Application of Artificial Intelligence Techniques to Social Media Data”, *Appl. Sci.*, Iss. 12, No. 12, p. 5834, Jun. 2022. <https://doi.org/10.3390/app12125834>
- [7] B. Rathnayake, D. Kasthurirathna, “Generating an Optimal Tour Plan with Optimization”, *Int. J. of Comp. Appl.*, Iss. 184, No. 38, pp. 31–39, Dec. 2022. <https://doi.org/10.5120/ijca2022922473>
- [8] R. A. Sánchez-Ancajima, M. Jiménez-Carrión, F. Gutierrez, A. O. Hermenegildo-Alfaro, M. A. Saavedra-López, “Applications of Intelligent Systems in Tourism: Relevant Methods”, *J. of Internet Services and Information Security*, Iss. 13, No. 1, pp. 54–63, Mar. 2023. <https://doi.org/10.58346/JISIS.2023.I1.006>
- [9] Y. Chen, X. Zheng, Z. Fang, Y. Yu, “Research on Optimization of Tourism Route Based on Genetic Algorithm”, *J. Phys. Conf. Ser.*, Iss. 1575, No. 1, p. 012027, Jun. 2020. <https://doi.org/10.1088/1742-6596/1575/1/012027>
- [10] E. Saeki, S. Bao, T. Takayama, N. Togawa, “Multi-Objective Trip Planning Based on Ant Colony Optimization Utilizing Trip Records”, *IEEE Access*, Iss. 10, pp. 127825–127844, Dec. 2022. <https://doi.org/10.1109/ACCESS.2022.3227431>
- [11] H. Sun, Y. Chen, J. Ma, Y. Wang, X. Liu, J. Wang, “Multi-Objective Optimal Travel Route Recommendation for Tourists by Improved Ant Colony Optimization Algorithm”, *J. of Advanced Transportation*, Vol. 2022, p. 6386119, Oct. 2022. <https://doi.org/10.1155/2022/6386119>
- [12] L. Sengupta, R. Mariescu-Istodor, P. Frănti, “Which Local Search Operator Works Best for the Open-Loop TSP?”, *Appl. Sci.*, Iss. 9, No. 19, p. 3985, Sept. 2019. <https://doi.org/10.3390/app9193985>

Oleksandra Stan¹, Tetiana Marusenkova², Iryna Yurchak³

¹ Software Department, Lviv Polytechnic National University, S. Bandery street 12, Lviv, Ukraine,
E-mail: oleksandra.stan.mppzip.2022@lpnu.ua, ORCID 0009-0000-7010-8701,

² Software Department, Lviv Polytechnic National University, S. Bandery street 12, Lviv, Ukraine,
E-mail: tetiana.a.marusenkova@lpnu.ua, ORCID 0000-0003-4508-5725,

³ Computer Design Systems Department, Lviv Polytechnic National University, S. Bandery street 12, Lviv, Ukraine, E-mail: iryna.y.yurchak@lpnu.ua, ORCID 0009-0005-9100-8511

COMBINED APPROACH TO BUILDING OPTIMAL ROUTES FOR INDIVIDUAL TRIPS IN A MOBILE APPLICATION

Received: February 17, 2024/ Revised: June 12, 2024 / Accepted: July 02, 2024

© Stan O., Marusenkova T., Yurchak I. 2024

Abstract. The paper deals with building optimal routes for individual trips under the influence of many factors and possible changes in the input parameters (such as weather conditions, traffic congestion, etc). We have analyzed four classes of algorithms for solving the traveling salesperson problem and evaluated their applicability in a tourist mobile application. The software should be a mobile application since only a few travelers take computers or laptops but most of them carry smartphones. The disadvantages of heuristic and metaheuristic algorithms have been considered. These include the dependence on the initial parameters, non-guaranteed optimal solutions, and the risks of being stuck in local optima. The exact methods have been discarded as unaffordable in mobile applications because of their computational complexity. Upon the conducted research, we propose a combined approach that uses the genetic algorithm as a global strategy and the four variations of the local search algorithm (Relocation, 2-opt, 3-permute, and Link swap) for refining the found solutions. The architecture and technology stack for the developed mobile application have been given, too. The future work implies searching for solutions to the group traveling salesman problem with the possibility of a joint trip plan edition by all the tourist group members and the multi-agent routing problem.

Keywords: planning trips, optimal route, genetic algorithm, local search, heuristic algorithm, metaheuristic algorithm, traveling salesperson problem, mobile application.