6-30-2022

Personalized Group Itinerary Recommendation using Cultural Algorithm

Farzaneh Jouyandeh
University of Windsor

Personalized Group Itinerary Recommendation using Cultural Algorithm

By

Farzaneh Jouyandeh

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2022

Personalized Group Itinerary Recommendation using Cultural Algorithm

by

Farzaneh Jouyandeh

APPROVED BY:

R. Rashidzadeh
Faculty of Engineering

Z. Kobti
School of Computer Science

P. Moradian Zadeh, Advisor
School of Computer Science

May 24, 2022

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The tourism industry plays a vital role in today's world. Many people travel around the world to visit and explore other places. However, planning an itinerary is one of the most challenging and time-consuming tasks for many travelers. It could be even more complicated when they travel as a group with different constraints and various choices of points of interest (POIs). The problem of group itinerary recommendation is an extension of the orienteering problem and is NP-hard, which can be defined as an optimization problem. This research will address the problem by proposing a personalized group itinerary recommendation algorithm using cultural algorithms. Cultural algorithms are evolutionary algorithms that use knowledge to guide the search direction during the evolution process. The main objective of our proposed model is to maximize the group's satisfaction by optimizing the number of visiting POIs, while considering the interests of all users, travel time, visit duration, and budget. The performance of our proposed model is evaluated on real-world datasets and compared with the existing methods.

The results revealed that our proposed algorithm outperforms alternative baselines on both datasets in most of the experiments, which means our final solution had better quality compared with other algorithms. Furthermore, non-parametric tests demonstrated that this approach generates consistent results in various situations and is notably different from existing algorithms.

DEDICATION

This thesis work is dedicated to my husband,

Mahdi,

who has been a constant source of support and motivation during the challenges of graduate school and life. I am genuinely grateful for your presence in my life.

This work is also dedicated to my wonderful parents,
Robab and Hadi,
who have always loved me unconditionally and whose lives have taught me to work hard for what I aspire to achieve.

ACKNOWLEDGEMENTS

Foremost, I would like to sincerely express my sincere gratitude to my supervisor Dr. Pooya Moradian Zadeh, for his constant support, patience, enthusiasm, and immense knowledge. I have learned a lot from him, and this dissertation would not have been completed without his supervision and constructive feedback.

Besides my advisor, I would like to offer my special thanks to my thesis committee members, Dr. Kobti and Dr. Rashidzadeh, for their beneficial advice, insightful comments, and suggestions for my thesis.

I owe my deepest gratitude to my beloved husband, parents, and sister for all the tremendous understanding, unconditional support, and continuous encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

POI	Point Of Interest
ML	Machine Learning
EA	Evolutionary Algorithm
CA	Cultural Algorithm
BS	Belief Space
GA	Genetic Algorithm
MCTS	Monte Carlo Tree Search
NFL	No Free Lunch

CHAPTER 1

Introduction

This section offers a brief overview of the problem of personalized group itinerary recommendation. It provides some background information and different aspects of recommending an itinerary. Moreover, it includes our motivation for tackling the problem, a description of the problem, research objectives and contributions, and a brief outline of thesis documentation.

1.1 Itinerary

An itinerary is a comprehensive schedule or timetable for a trip, generally including destinations to be visited at specific times, the sequence of visiting each place, and the modes of transportation to get between them [1]. In this research, an itinerary is defined as a set of unvisited Points of Interest (POIs) with the sequence to visit them. Every tourist needs a plan for visiting a destination to take the most advantage of their trip. Some tourists refer to the travel agencies which offer itinerary planning services. For suggesting an itinerary to a tourist or a group of them, a travel agent needs to understand all the aspects of his/her target market since a travel itinerary can serve various functions for different types of travelers. It must appeal to users' interests or adhere to their travel constraints.

Consequently, the itinerary should be personalized based on the tourists' preferences and limitations [2]. Although the internet and travel guides can provide information related to tourism, these resources only recommend popular POIs or generic itineraries and do not address individual tourists' particular interests or comply with their restrictions, such as their diverse temporal and spatial limits [3]. Therefore, a suitable personalized itinerary can be defined as a tour that includes as many POIs as feasible, maximizes the user's visit duration within a time budget, maximizes overall popularity, and keeps the user's cost on POI entrance to a minimum.

1.2 Itinerary recommendation

The itinerary recommendation is the task of suggesting a plan for a tourist or a group of them to visit some destinations, based on their interests and considering some constraints, including time limits, budget, and the number of places to visit [4]. It usually corresponds to orienteering problems, where the underlying mathematical models consider a wide range of constraints and satisfaction metrics [5]. Itinerary planning is a time-consuming and complicated procedure [6]. Tourists must gather information from various data sources, manually process data and choose which POIs to visit based on their particular preferences [7]. Due to the enormous amount of information available, they find it challenging to narrow down a possible set of POIs to visit in an unfamiliar city [4].

Aside from point selection, the touring problem appears to be computationally intractable and complicated [7]. Even once the visitor has selected an appropriate selection of POIs to visit, planning the proper order in which to visit the POIs will take time and effort [4]. In order to enhance overall satisfaction in traveling, the planning process comprises researching destinations, comparing rates, booking lodgings, creating routes, estimating time, and organizing schedule [8].

Due to the diverse interests and trip constraints, such as having limited time to complete the tour and start and endpoints of each unique tourist, tour recommendation and itinerary planning are complex tasks. In particular, it could be more challenging when a group of tourists travels together since the itinerary needs to cover all the group members' interests to provide a minimum satisfaction for all of them while taking certain constraints into account.

In itinerary recommendations, there are various possible solutions to be suggested to the target market, but the problem is to find the most appropriate solution that matches the tourists' preferences and constraints. This problem is classified as NP-hard [9] and cannot be solved in a polynomial-time algorithm with a bounded approximation ratio, or it might not be possible in a proper time frame. Therefore, this problem can be defined as an optimization problem since finding the ideal solution is impossible in a limited time. Therefore, to solve that, instead of searching for the best solution, we need to find a near-optimal solution according to the variables of the problem. We elaborate on optimization and its different types in the next section.

1.3 Optimization

Optimization is a strategy for finding the optimal collection of solutions for a given objective function under certain restrictions or limitations when there are many possible solutions [10]. The following is a mathematical description of an optimization problem: Let $\delta = (\delta_1, \delta_2, ..., \delta_n)$ be a vector with n entries that can take values from the $X^n : \delta_i \in X$ domain. The domain X can be discrete, such as $X = \{0, 1\}$ or $X = Z$ the set of all integers (in which case it is an integer optimization problem), or continuous, such as $X = R$ the real numbers. Furthermore, let H be a real-valued function, such as the cost function or objective. The challenge of minimization, therefore, becomes: finding $\delta \in X^n$ that minimizes H. Similarly, a maximizing problem is defined in an analogous way. Since maximizing a function H is equivalent to minimizing $-H$, it is sufficient to address minimization problems solely [11].

Thus, optimization techniques identify a plausible solution in a short period of time or with limited resources. The best solution is based on one objective function in single-objective optimization problems [12]. Multi-objective problems are more challenging than single-objective problems since they need to cope with more than one objective function simultaneously.

In this research, we formulate the personalized itinerary recommendation as a multi-objective optimization problem. Therefore, it needs to be optimized into a solution that considers all users' needs and limitations. In this regard, Evolutionary

Algorithms (EAs) have been proved to be efficient for solving optimization problems. Thus, some information and details about EAs are provided in the next section.

1.4 Evolutionary algorithms (EAs)

EAs are algorithms that can evolve while performing optimization or learning tasks [13]. They have three primary features:

- **Population-based:** EAs keep track of a population of solutions in order to optimize or learn the problem in parallel. The population is a fundamental principle of evolution [13].

- **Fitness-oriented:** An individual is a name given to each solution in a population. Every individual has a performance evaluation, referred to as their fitness value. Fitter individuals are preferred by EAs, which is the basis for algorithm optimization, and convergence [13].

- **Variation-driven:** Individuals will be subjected to a series of various operations in order to simulate genetic evolution, which is necessary for searching in the solution space [13].

EAs usually begin the optimization process by generating a set of randomly generated trial solutions for a specific problem. This random set is then evaluated iteratively by the problem's objective function(s) and evolved to minimize or maximize the objective(s) [14]. Despite the simplicity of this framework, optimizing real-world problems necessitates taking into account and addressing several challenges, the most significant of which include local optima, expensive computing costs of function evaluations, limitations, multiple objectives, and uncertainties [14].

A frequently discussed question is that, despite several optimization techniques, why there are several works around this topic. In the realm of optimization, there is a theory known as No Free Lunch (NFL) that states that there is no optimization solution that can solve all optimization problems [15]. This means that the best algorithm for test functions or other real-world problems may not be the solution to a specific problem. An algorithm may need to be modified, improved, or adapted to address an issue. Consequently, this keeps the field active, evidenced by the high number of new algorithms developed each year.

1.5 Research Motivation

The tourism industry plays an essential role in today's world. It is a crucial source of income, development, and employment for many countries. It also plants a sense of cultural exchange between foreigners and citizens [16].

In 2016, the business volume of global tourism exceeded that of oil exports, food products, or automobiles [17]. Furthermore, in 2018, Canada had 21.2 million international tourists, and it grew compared to the past years [18]. Annually, tourism spending rose 4.4% in 2021 after a 49.0% decline in 2020 because of the impacts of COVID-19 worldwide. As presented in Figure 1, despite the ongoing threat of COVID-19 and the development of additional variations in 2021, rising vaccination rates and the resulting reduction of restrictions led to massive tourism spending in Canada.

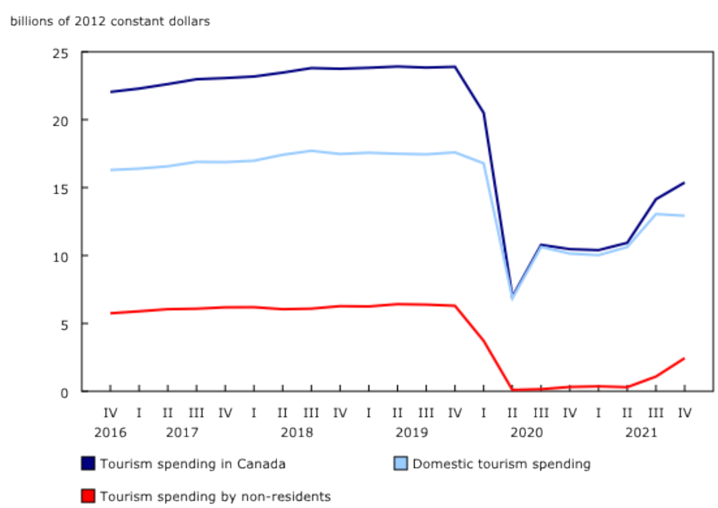

Figure 1: Tourism spending increasing in Canada

While there is a high volume of tourists traveling all around the world every year, planning an itinerary is one of the most demanding tasks for travelers [3]. Tourists want to visit various POIs based on their budget, time, and interest in each destination. The task of recommending an itinerary becomes even more complicated when tourists travel as a group with different constraints and various choices of POIs. In this situation, all the tourists' interests and limitations must be considered to satisfy all of them.

As mentioned before, the problem of group itinerary recommendation can be defined as an optimization problem that extends the orienteering problem (routing problem). This problem is considered NP-hard, which means no polynomial-time algorithm with a bounded approximation ratio exists for this [9]. According to this issue, AI can be used to recommend an appropriate itinerary for a group of tourists, which has been proved to be an efficient method for similar problems [19]. Various research works focus on group itinerary recommendations, but since this problem is NP-hard, finding an algorithm that can solve it efficiently is still an open problem [9]. Accordingly, evolution approaches have received increasing attention in this area recently [20, 21, 22].

1.6 Problem Statement

Assume a region is represented by a complete weighted graph $G = (V, E, w)$, where $V = \{v_1, v_2, ..., v_m\}$ is the set of m POIs, and each edge $e_{ij} \in E$ represents the route from v_i to v_j. The weight of each e_{ij} presented by w_{ij} denotes the distance between v_i and v_j. Each POI is assigned to a category, c_k ,where $c_k \in C = \{c_1, c_2, ..., c_z\}, 2 \leq z$.

A set of d users forms a group denoted by $U = \{u_1, u_2, ..., u_d\}$ wants to visit this region. Each user u_i has a travel history represented by $H_{u_i} = \{(c_1, val_1), (c_2, val_2), ..., (c_k, val_k)\}$, where val_t is a value between 0 and 1 that shows users' interest level to category c_t. Each group has specific time to complete the trip denoted by $MaxT$ and a maximum number of visiting POIs presented by $MaxPOI$ and a specific budget denoted by $MaxB$.

The problem is identifying the most suitable itinerary for the group considering

Figure 2: An example of recommending an itinerary to a group of tourists

their interests based on their travel history and the group constraints. It is an optimization problem since various possible itineraries exist to be recommended to a group of tourists. The main goal is to find the best optimal solution that prepares the tourists' highest satisfaction level while not exceeding their limitations.

As an example, as shown in Figure 2, there is a group of users consisting of three tourists with some limitations, such as budget ($1000), time limit (8 hours), and the maximum number of visiting POIs (4 POIs). As presented in this figure, this group aims to travel to a region including seven pre-defined POIs. The recommended itinerary for this group formed as a set of some selected POIs from that region, such as {1,3,7,5}. It depicts the points to visit and their sequence.

1.7 Research Objectives

In this thesis, we propose a novel evolutionary optimization algorithm based on cultural algorithms to address the problem of group itinerary recommendation. Our main objective is to recommend a personalized itinerary for a given group with four criteria, as shown in the following table:

Criteria	**Description**
POIs	Visit as many POIs as possible
Time	Maximize users' visit duration within a time budget
Popularity	Maximize the overall popularity of the POIs
Budget	Keep users' cost on POI entrance as less as possible within the group budget

Table 1: Problem criteria

We aim to have as many as possible POIs in the recommended itinerary while not exceeding the limitations. The other objective is to maximize users' visit duration within the time budget to take the most advantage of their time. Considering that the most popular POIs are more likely to be satisfactory for most tourists, we aim to maximize the popularity of the recommended points. Finally, we try to minimize tourists' cost during the tour, especially on POIs' entrance cost and keep it less than the group budget.

1.8 Research Contributions

Our main contribution is defined as proposing a novel cultural algorithm to solve the problem of recommending a personalized itinerary to a group of tourists as a multi-objective optimization problem with some characteristics. The first one is that the size of the solution is variable, which means we can have a different number of POIs in the recommended itinerary. The second characteristic is that the order and sequence of selected items in the solution play an important role. Therefore, we need to find the best order of POIs in an itinerary to obtain the highest possible user satisfaction.

In this study, our goal is to propose an algorithm which is able to recommend near optimal sequence of POIs for a given group, and identify the proper length of the itinerary. We aim to design a new structure of belief space to store the extracted knowledge of sequence and selection of points and utilize it in the recommendation process. Using the noted knowledge, we target to improve the accuracy of the recommendation process and study the time complexity of the proposed algorithm. We hypothesize that using the knowledge gained through the evolution process will lead to improving the performance and accelerating the convergence. We expect our solution to obtain more user satisfaction within a reduced time frame compared with the existing approaches, especially when facing big data.

This study considers a group of tourists with diverse priorities traveling together. Thus, the preferences and interests of all of them should be covered to meet the expectations of all group members and get the most overall satisfaction. This could be a challenging task, especially when some tourists have some interests in contrast with others in a group. There is a trade-off between different selections to receive the most pleasing outcome in this situation.

Our project's outcome has the potential to enhance the traveler's experience and pleasure. It is also able to positively contribute to the economic growth process of tourism and assist the tourism industry in introducing the new POIs. Moreover, both individuals and travel agencies can use this recommendation algorithm. In addition, the proposed algorithm can be utilized as a solution for other similar problems in computer science, such as educational planning, which has the same structure as the problem mentioned above. In educational planning, we need to select a set of courses or learning materials according to the users' needs and put them in an appropriate sequence to get the highest efficiency.

There are some similar problems to the itinerary recommendation problem in the field of computer science from which scheduling [23], orienteering [21], and team formation [24] problems can be mentioned. In all of these problems, point selection is an issue. In other words, finding the best selection from a set of options is a part of all of these topics. At the same time, some factors differentiate this problem from others.

The first one is the unspecified size of the solution. In scheduling, some specific tasks need to be covered. So, for finding the best solution, we know how many tasks/jobs should be assigned; in orienteering, we need to generate a path through a set of specific nodes, and in team formation, there are a required set of skills to be covered by some individuals. Therefore, in all of them, the size of the solution is defined, and fixed [23].

The second point is identifying the sequence or the order of the selected set in the next step. In itinerary planning, after selecting the appropriate set of points, their sequence in the recommended itinerary plays an important role in users' satisfaction. While in team formation or scheduling, the problem is to allocate the best individual to a task, and the order is not taken into account.

In itinerary recommendation, we know the maximum number of POIs that a group is able to visit, but the length of the itinerary can variate from two points to MaxPOI. Sometimes, a shorter itinerary can satisfy the tourists more when it contains POIs matching users' interests and are more popular, rather than a longer itinerary with POIs with less popularity or more entrance cost. Thus, there is a trade-off between the number of points, time, popularity, and cost that needs to be balanced. In addition, we need to find the best sequence of the selected points in the recommended itinerary, considering the distance between POIs.

1.9 Thesis Outline

The following chapters of the thesis are organized as follows: Chapter 2 contains a review of some of the existing methodologies and models in the area of itinerary recommendation. In Chapter 3, the proposed methodology for achieving the research's goals is presented. Chapter 4 covers the experimental setup, data, analysis, and evaluation results, followed by the conclusion and future work in Chapter 5.

CHAPTER 2

Related Works

Various approaches focus on addressing the itinerary planning problem by recommending interesting POIs for tourists, generating realistic itineraries, and taking travelers' constraints into account. This section briefly reviews the various techniques and methods for recommending an itinerary and presents it in three categories. The first one contains the papers which used optimization approaches. The second group of literature contains deep learning based recommendations. Other related approaches proposed for this problem are reviewed in the third part. The last section presents a summary of limitations in the existing literature.

2.1 Optimization approaches

Itinerary recommendation seeks to recommend several POIs as part of a connecting itinerary to enhance tourist satisfaction while adhering to the given constraints. There are various research that employed optimization approaches to address this problem.

To tackle the itinerary planning problem efficiently, meta-heuristic algorithms performed population-based optimization, such as some evolutionary algorithms. The Genetic Algorithm (GA) was one of a kind that solves optimization problems using natural selection and genetic principles. This algorithm was used in different versions to address this problem. One of them was [25] in which the authors focused on travel itinerary with restaurant selection. Their objective was to maximize the total collected utility in each visited location while maintaining the total travel time under a specific constraint. They developed a GA to solve this as an optimization problem. Concerning the existing limitations, it developed a satisfactory trip itinerary consisting of a collection of high-ranked tourist attractions, and restaurants [25].

In [26], the authors developed the PersTour system that could recommend POIs that were interesting to the tourist and plan these POIs in the form of a tour itinerary. This system was able to recommend tours based on either POI popularity or tourist interest preference. For generating this system, they adapted the Ant Colony Optimization algorithm [26]. This algorithm was based on the idea that agents are more likely to pick a better path and has been traveled recently. As a result of this preference, picking a single option was reinforced over time, resulting in choosing the path as the best answer.

In [27], the proposed algorithm recommended an itinerary comprising a series of POIs in a city and including as many mandatory POIs as possible within the travel time budget. This was the problem of planning a sequence of visits of a given number of POIs which must be visited within a limited time. It was a multi-objective optimization problem, and it was hard to design a fixed fitness score for each tour. As a result, they focused on optimizing the objective function directly. By assigning various levels of priority to the metrics used to evaluate individuals, the objective function was defined based on mandatory POIs, visit duration, and total profit of POIs [27].

An adaptive genetic algorithm (AGAM) was proposed in [28] to formulate the personalized itinerary recommendation task as the Multi-objective optimization problem. In AGAM, the crossover and mutation probabilities were dynamic, which helped the algorithm locate the optimum solution and avoid the program falling into the local best solution. These probabilities were calculated based on the average fitness of the population. Moreover, after the crossover and mutation process, some unvisited POIs weer inserted into the individuals of the next generation while it did not exceed the program limitations such as time and budget. The authors also assigned different weights to each factor in the fitness function to generate tailored itinerary planning that better meets a variety of tourist preferences [28].

In [29], using a GA, an algorithm called PWP was proposed that recommends multiple itineraries depending on visitor interest, the popularity of itineraries, and the cost of itineraries. The multi-objective optimization approach used in this research is NSGA-II [30]. However, it did not work well when a tourist wishes to visit unfamiliar places. This algorithm was developed with the intention that POI attractions could fall into any category. However, it ignored the case where a POI attraction falls under more than one category. The Flickr dataset [31] was being used to compare the PWP method to alternative baseline approaches in different cities. Even GA was an efficient stochastic optimizer with a focus on planning problems. However, the crossover and mutation probability still had flaws.

Another trip itinerary recommendation was proposed in [32] for a group of tourists with individual preferences on various POIs. The path to walk through POIs was also suggested by utilizing the Prize-Collecting Vehicle Routing Problem solution. The proposed algorithm was a combination of game theory and meta-heuristic approach. The n-person Battle of Sexes game was used to pre-configure the set of POIs to be visited, yielding three separate sets based on whether their presence was restricted, mandatory, or optional. The routes between POIs were then determined using the meta-heuristic firefly method, which was strengthened by the coordinates-related encoding/decoding process.

2.2 Deep learning based recommendations

Authors in [33] suggested a solution to create multi-day itineraries that are personalized according to the user's travel style using some factors, including the user's level of interest. They also added personalization options, such as the traveler's pace and the diversity of POI categories expected from the itinerary. To create clusters, they applied agglomerative clustering with a time limit to the POIs from each category separately. They claimed that personalizing the visiting duration of POIs provided better itineraries in terms of matching the pace of the traveler and in providing more satisfying itineraries.

Most deep learning techniques were incapable of concurrently handling many conflicting close and long-distance preferences and recent and prior visit influences. Recent visits and nearest preferences were the subjects of some deep learning approaches, such as Long Short-Term Memory (LSTM) or Recurrent Neural Network (RNN) based techniques, depending on spatiotemporal relationships. As a result, learning spatiotemporal dependencies could be difficult and complicated [34]. The POI queuing time and its prediction for the following POI recommendation were considered for the first time in [34]. A Transformer based Learning Recommendation was proposed as a multi-task, multi-head attention transformer model. It recommended the next POIs to the target users and predicted queuing time to access the POIs simultaneously using two parallel joint learning processes. However, it only studied the queuing time aware top-k POI recommendation problem and could not construct a full itinerary considering the budget time.

An unsupervised deep learning model was used in [35] to embed the POI textual contents. It proposed a Deep model for Contextual Collaborative learning (DCC), which seamlessly integrated POI textual contents, the historical user-POI visits, and the POI categories to predict the user interests and visit duration. Following that, an Iterated Local Search based algorithm was provided to calculate the visit sequence with optimum satisfaction, which comprised numerous POIs and personalized POI visit duration. As for the limitations of this approach, it still had the user cold-start problem, which means it cannot recommend itineraries to people whom there is no visiting history for them.

Although most research focused on users' visit history to tourist attractions, it did not supply enough information on their own, and their reviews of these locations are highly significant. Consequently, another group of recommendation systems used comment analysis to determine user preferences. Users' reviews on social media sites were mined in [36] as a rich data source that incorporated their preferences directly. It employed sentiment analysis and semantic clustering to derive users' preferences. Furthermore, the authors extended their work in [37] by identifying attractions' features from ratings and reviews and delivering a customized recommendation system. They proposed a context-aware recommendation system where user preferences were derived from their comments and reviews in the first step. Similarly, the characteristics of tourist attractions were collected from tourist reviews in the second step. Finally, personalized recommendations were offered based on contextual data and similarities between user preferences and tourist attraction attributes. It semantically compared the user's preferences with the features of attractions to suggest the most matching points of interest to the user. However, it did not consider tourists' limitations and the route or the sequence that they will visit the points.

The path that leads to each POI is considered in [38] as an extension of the orienteering problem, assuming that if a visitor arrives at a POI via an appealing route, the user experience will be enhanced. A personalized trip recommendation with attractive routes was proposed in this study that finds the most attractive routes based on the popularity and Gini coefficient of each POI in the users' travel history. Then the preference of each attractive route was obtained by unsupervised learning. Finally, the proposed algorithm used k-means clustering to cluster user preferences in z-dimensional space, which was chosen for its computing efficiency. The performance of the algorithm was tested by applying that to a synthetic dataset and Foursquare dataset [39].

Existing solutions may not represent tourists' preferences because they preferred to promote POIs with short prior visitation periods when constructing itineraries. These suggestions may contradict real-life scenarios, as visitors spend less time at POIs they are not interested in, resulting in the inclusion of unsuitable POIs. Furthermore, creating itineraries based on selected POIs is a time-consuming and challenging task. Most existing methods entailed filtering through many non-optimal, redundant itineraries, which takes time to examine and compile. Authors in [40] proposed an adaptive Monte Carlo Tree Search (MCTS)-based reinforcement learning algorithm that employed an effective POI selection strategy that prioritized POIs with long visiting times and short queue times, as well as high POI popularity and visitor interest. They applied the MCTS pruning methodology to reduce search space by filtering out non-optimal and duplicate itineraries early in the process, which improved the time efficiency of the proposed method.

2.3 Other related approaches

In this section, other approaches that were utilized for itinerary recommendation are covered. In [4], the algorithm named PERSTOUR was proposed for recommending personalized tours with POIs and visit duration based on POI popularity, users' interest preferences and trip constraints. The problem was modelled using an Orienteering problem formulation that took into account user trip limitations like time limits and the need to start and stop at certain POIs. The concept of time-based user interest was also introduced in this paper, where a user's level of interest in a POI category was based on his/her time spent at such POIs, relative to the average of users visit duration.

In [41], the authors developed a framework for automatically detecting real-life travel sequences and determining POI popularity and user interest using geo-tagged photos, which was utilized to train their algorithm. They extended their work in this research, where in their previous work, they recommended tour itineraries that employ the same non-personalized POI visit duration for all users, or they did not take POI visit duration into account at all [4]. In [41], the tour itineraries were being provided with unique POI visit duration that are tailored to individual users' time-based preferences. They also improved the initial time-based user interest by prioritizing recent POI visits and ignoring POI visits from the distant past, and finally evaluated their proposed algorithm using a Flickr dataset across ten cities.

A variation of the Orienteering issue was used in [42] to model their recommendation problem. Their main goal in this tour recommendation was to suggest a tour itinerary that maximized the total profit from visiting the list of POIs while also ensured that the tour itinerary could be completed within a specific time budget. This was an Integer program where, unlike previous research that considered time-based or frequency-based user interest, they introduced photo frequency-based user interest generated from the number of photos taken by the user at a POI of a specific category. The reason behind it was that a user is more likely to click on more photos of the POI that arouse his/her interest. Consequently, this algorithm worked well when recommending an itinerary to users with no prior knowledge about them.

In [43], a tour recommendation framework was proposed that takes each tourist's list of must-see and preferred points of interest and creates multi-day tours that included all of the must-see points. Fairness among group members is a feature that was considered in this study to ensure that all members of the group are motivated to take part in the group tour. A greedy approach was followed in the proposed model, where first tried to cover all the must-visit points and then the remaining budget and time was allocated to the preferred points equally between group members according to the fairness parameter. However the proposed solution was ineffective if tourists desired to explore a new city without knowing the list of POIs.

An algorithm, called gTour, was proposed in [44] to give a set of itineraries for a tourist group in order to boost the tour's interest and popularity among all tourists in the group while lowering transportation costs. It used the Subgame-Perfect Nash equilibrium (SPNE) of game theory, where each member of the group takes on the role of a game player, observing the other players' behavior to fulfill the wishes of all group members. The proposed method used geo-tagged photos to provide visitors with realistic trip sequences. Its positive aspect was that it worked well when a visitor decides to visit an unknown location where they have no prior travel experience.

Another recommendation approach was proposed in [45] based on Monte Carlo Tree Search (MCTS). This algorithm was a personalised itinerary recommendation based on the implementation of MCTS. It had four steps: Selection, Expansion, Simulation, and Back-propagation. The core notion was that game play began with iterations of random node selection to explore moves, with the results of those moves being recorded. Following that, MCTS moved away from random moves and gradually built on earlier successes by converging to moves that resulted in win states during following game plays [45]

The suggested algorithm in [1] was based on two different sorts of users: local and global. A local user was someone who lives in a specific location and uses to frequent the POIs there. Using this concept, this algorithm was capable of recommending new locations or a new type of POIs, regardless of the users' travel history. This algorithm also recommended multiple itineraries using the MCTS approach, whereas previous works suggested multiple POIs in a single itinerary. The MCTS algorithm [46] was based on the tree search problem. Each board position was represented as a node in the tree, and the game's end state (win/loss) was represented as the leaf node. The algorithm's outcome was a list of multiple itineraries suggested to tourists based on user interest, the popularity of tours, and travel costs.

2.4 Overall limitations

After conducting a comprehensive literature review, we summarize the most critical limitations of the reviewed literature in this section. The most usual problems and deficiencies are as follows:

1. **Cold start problem:** It refers to the algorithms that cannot perform well on unfamiliar POIs or users for whom there is no prior information about them [47].

2. **Grey-sheep problem:** In recommendation systems, it refers to users who have unique interests and tastes, making it challenging to create accurate profiles [48]. In itinerary recommendation, it happens for users with unique preferences or POIs with unique categories.

3. **Route selection:** It happens when the algorithm selects only the best POIs matching users' interests but do not consider the route between them or their sequence.

4. **Knowledge extraction:** The existing models do not employ the knowledge gained during the recommendation process to obtain a better result.

A review of some of the existing models and techniques on itinerary recommendation was given in this section. We witnessed that although some algorithms were proposed as a solution to this problem, each of them had inefficiency or lacks performance in some aspects, and some limitations exist, as mentioned in this section.

To the best of our knowledge, no research work has been done that applies the Cultural Algorithm to this kind of recommendation problem. Accordingly, to overcome some of the shortcomings of the existing methods, we propose a personalized itinerary recommendation algorithm for a group and present our proposed algorithm in the following section. To cover the Route selection and Knowledge extraction problem, in this research, we try to extract and utilize the knowledge during the evolution process and consider the sequence of POIs to visit.

CHAPTER 3

Proposed Model

This section discusses our knowledge-based evolutionary approach for recommending the best itinerary to a given group, which is based on a cultural algorithm (CA).

3.1 Cultural algorithm

We use a cultural algorithm (CA) to address the mentioned problem. Cultural algorithm is an Evolutionary Algorithm that is based on the conceptual models of the human cultural evolution process [49]. Human behavior results from two distinct but interconnected evolutionary processes: genetic evolution and cultural evolution. Variations in genes can lead to cultural changes, which can influence the genetic selection, and vice versa. The concept of CAs is based on the idea of using diverse sources of knowledge during the search process [49].

The population space of CAs is a genetic component containing individual solutions. CAs have been highly used to solve complex optimization problems like team formation. It also uses the knowledge sources gained during the evolution to influence the search process. In this algorithm, as shown in Figure 3 [49], there is a knowledge component that is called "Belief Space" (BS) in addition to the population component. BS represents the cultural information gained during the evolution process, and collects the information about the behavior of individuals in the search space. Therefore, the learning process is performed in population and belief space simultaneously [49]. The concept behind this algorithm is that we will be able to develop a better population in later iterations by gaining the knowledge that underpins their good performance.

In CA, various types of knowledge such as Normative, Situational, Historical will be extracted from the selected group of individuals [50]. To further elaborate, situational knowledge saves the individual's best experience from each generation of the population space. In addition, individuals' standards of behavior in each dimension of the problem are indicated by normative knowledge [50].

The belief space of CAs plays the role of a knowledge repository, where the knowledge acquired by individuals through generations is stored [51]. Like other meta-heuristic algorithms, the algorithm starts with a randomly generated population space. Then, the fitness function evaluates the generated individuals in the population. The best individuals regarding their fitness scores will be transferred to the BS using Accept function, which determines which individuals in the population can affect the BS. Thus, in the following iteration, the CA uses the knowledge to steer the direction of the search and accelerate evolution, in addition to performing a crossover or mutation.

Therefore, after adjusting BS, a new population will be generated either from BS or by variate functions, including crossover and mutation, from the original population. When generating new individuals from the BS, the Influence function determines the effect of the knowledge sources on the population space. In other cases, various crossover and mutation methods are employed, using some selected individuals from the previous population. These operators help the algorithm to escape from the local maxima. This process will be repeated until the stop criteria have been met or for the specific number of iterations. The individual with the highest fitness value is chosen as the final solution for the particular problem at the end of the procedure [51].

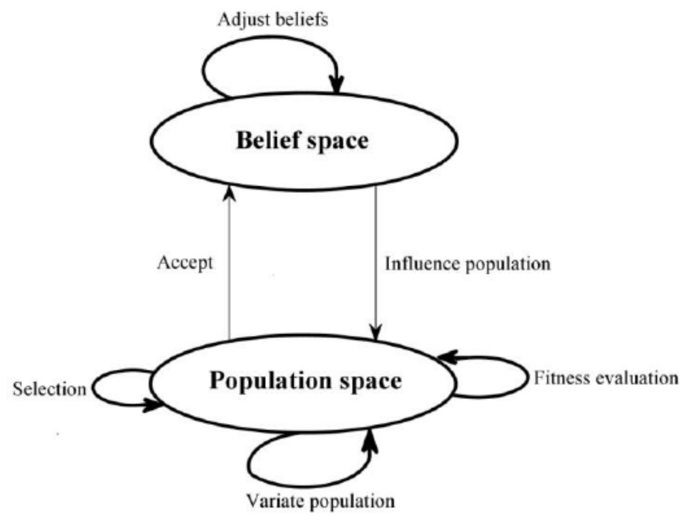

Figure 3: The CA framework

3.2 Constraints

All tourists have some limitations, such as time limits and budget, needed to be considered for recommending an itinerary. Thus, as shown in Table 2, four constraints are taken into account for solving this problem.

Limit	**Description**
Visiting POIs	No POI is visited more than once
Time Limit	The time taken for the itinerary is within the time limit MaxT
Number of POIs	Maximum number of POIs in an itinerary should not exceed MaxPOI
Budget	The entrance cost of the POIs should not exceed the budget MaxB

Table 2: Constraints

Therefore, considering an itinerary, $I = \{v_u, ..., v_k\}$, which is recommended to a group of tourists, we attempt to solve the problem such that:

$$\sum_{i=1}^{m}\sum_{j=1}^{m} e_{i,j} \leq 1 \tag{3.2.1}$$

$$\sum VisitDuration(v_i) \leq MaxT, \quad \forall v_i \in I \tag{3.2.2}$$

$$|I| \leq MaxPOI \tag{3.2.3}$$

$$\sum Cost(v_i) \leq MaxB, \quad \forall v_i \in I \tag{3.2.4}$$

We formulate the mentioned constraints and consider them in the recommendation process. Constraint 3.2.1 ensures that the itinerary should not contain any duplicated POIs. As formulated in constraint 3.2.2, the time that the itinerary takes to complete the itinerary should also be less than the group time limit (MaxT). Moreover, the group will give the maximum number of POIs capable of visiting during the time budget in advance (MaxPOI). Thus, constraint 3.2.3 ensures that the number of POIs in the itinerary should not be more than MaxPOI. Finally, the group has a budget in which the entrance cost of recommended POIs should not exceed this amount (MaxB), which is provided in constraint 3.2.4.

3.3 Proposed algorithm

3.3.1 Population

Assume a group on tourists, denoted by $U = \{u_1, ..., u_d\}$, tend to travel to a region with m number of POIs. Primarily, this algorithm generates a pre-defined number of random itineraries to make the initial population. An itinerary consists of some POIs with a specific sequence. Figure 4 presents a sample individual, which is a set of POIs with the length of MaxPOI such as $I = [v_6, v_4, v_2, v_5, v_1, v_3]$. It presents an itinerary that starts from the first node and continues to the last one.

v_6	v_4	v_2	v_5	v_1	v_3

Figure 4: An example of the individual representation

Assume population $P = \{I_1, I_2, ..., I_n\}$ which is a set of n individuals, where each individual $I_i = \{v_u, ..., v_k\}$, $v_j \in V$ is a sequence of POIs which forms an itinerary with maximum length MaxPOI. In the first step, a population of size n is created, including n individuals, which are randomly generated from the set of POIs. For better understanding, we present the population as a matrix where each row contains an individual. An example of the population is presented as follows, which shows its structure:

$$P = \begin{bmatrix} v_3 & v_6 & 0 & 0 \\ v_2 & v_4 & v_7 & v_1 \\ v_1 & v_6 & v_2 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ v_7 & v_2 & v_5 & 0 \end{bmatrix}$$

As presented in Figure 5, assume a group of three users with some limitations needs an itinerary to travel to this region. A sample generated population is shown consisting of n individuals. Each individual has six genes, as MaxPOI is 6 in this example. Each individual has a different size, and if its size is less than MaxPOI, the empty genes at the end of the individual are set as zero. This is to have individuals of equal lengths.

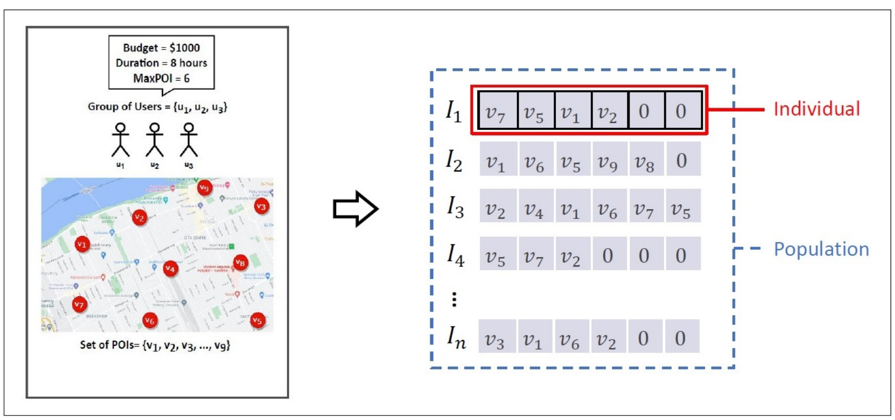

Figure 5: An example of the generated population

3.3.2 Fitness function

After generating the first population, the quality of all itineraries in that is evaluated using a fitness function. Regarding our objectives in this research, the fitness score is calculated using group interests, the total number of visiting POIs, the total popularity, visit duration, and entrance cost. First, the individual, I_i, is processed to check whether it exceeded the group limitations, including time and budget. If it crossed the constraints, the fitness score, $F(I_i)$, is set to -1. It leads the algorithms not to select this individual in the following steps. If the individual passed this evaluation, its fitness score is calculated using the following equation [28]:

$$F(I_i) = w_1 GroupInt_U(I_i) + w_2 Tn(I_i) + w_3 Tp(I_i) + w_4 Tv(I_i) + w_5(1 - Tc(I_i)) \quad (3.3.1)$$

Where w_j is the weight of each factor that can be adjusted to change the impact of each factor on fitness score, in addition, $GroupInt_U(I_i)$ is the total interest of the group U to the POIs in I_i, $Tn(I_i)$ is the total number of POIs included in I_i, $Tp(I_i)$ is the total popularity of the points included in I_i, $Tv(I_i)$ is the total visit duration or the total time that I_i takes to complete, and $Tc(I_i)$ is the total cost for I_i. All of these factors are normalized by dividing them by the maximum value of their kind.

The total interest of group presented by $GroupInt_U(I_i)$ is calculated using the following equation:

$$GroupInt_U(I_i) = \frac{1}{|U|} \sum_{z=1}^{d} \sum_{j=s}^{y} (Int_{uj}(c_z)\delta(Cat_{vz} = c_z), \forall c_z \in C \quad (3.3.2)$$

where

$$\delta(Cat_{vz} = c_z) = \begin{cases} 1 & \text{if } Cat_{vz} = c_z \\ 0 & \text{Otherwise.} \end{cases}$$

We calculate the group interests using Eq.3.3.2, which is the sum of users' interests for all categories in the generated itinerary divided by the number of users. The division is added to the equation to normalize the resulting number and keep it a value between 0 and 1. The symbols used in the following equations are defined in Table 3.

Symbol	**Meaning**
$Cat_(v_z)$	Category of POI v_z
$Nop(I_i)$	The number of POIs included in I_i
|V|	Total number of POIs
$Totp(I_i)$	Total popularity of the POIs in I_i
MaxP	The largest popularity among all POIs
$Totv(I_i)$	Total visit duration and travel time for I_i
MaxV	The longest visit duration among all POIs
$Totc(I_i)$	Total entrance cost for POIs in I_i
MaxC	The largest entrance cost among all POIs

Table 3: Symbols definition

As presented in the equations in Eq.3.3.3, all the other factors of the fitness function, such as the number of POIs, total time, total popularity, and total cost, are also divided by the maximum value of its kind to normalize all the values.

$$\begin{gathered} Tn(I_i) = \frac{Nop(I_i)}{|V|} \quad , \quad Tp(I_i) = \frac{Totp(I_i)}{MaxP} \\ Tv(I_i) = \frac{Totv(I_i)}{MaxV} \quad , \quad Tc(I_i) = \frac{Totc(I_i)}{MaxC} \end{gathered} \tag{3.3.3}$$

To elaborate, the number of POIs in individual I_i is divided by the total number of POIs in the region. $MaxP$ is calculated as the maximum popularity an individual can provide with size $MaxPOI$. $MaxV$ and $MaxC$ are the longest possible duration and the highest cost of an individual with size $MaxPOI$, respectively.

After calculating the fitness score for all the individuals, the population is sorted by fitness scores. Consequently, top x% of itineraries with highest fitness values will be selected for the next step.

3.3.3 Belief space

In the current step, different sources of knowledge are extracted from the selected individuals to form the belief space. In the proposed algorithm, the BS consists of two matrices. The first matrix, BS_1, is created out of historical knowledge. Assuming a selected individual as $I_i = \{v_u, ..., v_k\}$, this matrix is defined as a set of selected individuals $BS_1 = [I_1, I_2, ..., I_x]$. The knowledge about the POIs that have been selected resulting in the best fitness scores is stored in this part of the BS. Thus, each row of this matrix is one of the selected itineraries, with nodes contains one POI. The structure of BS_1 is inspired from [24] and presented as follows:

$$BS_1 = \begin{bmatrix} I_1^1 & I_1^2 & \cdots & I_1^{MaxPOI} \\ I_2^1 & I_2^2 & \cdots & I_2^{MaxPOI} \\ \vdots & \vdots & \ddots & \vdots \\ I_x^1 & I_x^2 & \cdots & I_x^{MaxPOI} \end{bmatrix} \tag{3.3.4}$$

Rather than the POI selection, the other crucial factor is their sequence in itinerary planning. As a result, the second part of the belief space, BS_2, reserves the order of POIs in the selected individuals. This helps the algorithm in the evolution process select the POI with the best result according to its previous node. Thus, the frequency that each POI occurred after each of the other POIs in the selected individuals is saved in BS_2. The historical knowledge is extracted and collected in this matrix. Considering a set of m POIs in a region, BS_2 is formed as follows:

$$BS_2 = \begin{bmatrix} 0 & fr_1^2 & fr_1^3 & \cdots & fr_1^m \\ fr_2^1 & 0 & fr_2^3 & \cdots & fr_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ fr_{m-1}^1 & fr_{m-1}^2 & fr_{m-1}^3 & \cdots & fr_{m-1}^m \\ fr_m^1 & fr_m^2 & fr_m^3 & \cdots & 0 \end{bmatrix} \tag{3.3.5}$$

where fr_j^i is related to the number of times POI v_i is located right after v_j in the best-selected individuals.

Using this knowledge, we can take the order of POIs into account in addition to the POIs selection. In the subsequent iterations, the new population will be generated not only by crossover and mutation methods but also by using the collected knowledge in the belief space. This will reduce the search space and accelerate the evolution process compared with the genetic algorithm.

Therefore, for generating an individual from BS, the first POI will be randomly selected from the first column of BS_1, and according to that node, BS_2 is employed afterward. Assuming v_i is selected from BS_1 for the first node, for picking the next one, we look at the v_i-th row of BS_2. First, we choose the columns with the highest values, which means the nodes with a high frequency of occurrence after v_i. If there is only one node with the highest value in that row, it is selected as the next node. In case of more than one column with highest value, one of them is randomly selected as the next node, and we repeat this process to the last node of the individual. Thus, if the selected value is located in v_j-th column, then v_j is the next node on the itinerary.

As an example to elaborate the process of generating a new individual from BS, assume a group of three users, $U = \{u_1, u_2, u_3\}$ are traveling to a region with a set of seven POIs, $V = \{v_1, v_2, ..., v_7\}$. Their limitations are defined as MaxPOI = 4, MaxB = \$500, and MaxT is 8 hours. The workflow of producing an individual from BS is depicted in Figure 6. In steps 1-3, the algorithm primarily generates n random individuals with different sizes of a maximum of four as population P. The fitness values for all individuals in P are calculated, and P is sorted according to it. Then, the top x% of individuals in P is selected as the best ones and transferred to the next step for creating the belief space.

Assuming 4 top individuals of the populations are selected, BS_1 and BS_2 are formed as shown in step 4. For generating a new individual from BS, a value from the first column of BS_1 is selected randomly for the first node of the new individual. In this example, v_3 is selected and allocated to the first node. For the remaining nodes, BS_2 is employed. For instance, for the next node we check the row for v_3 in BS_2. Since column v_6 has the highest value, it is chosen as the next node. For the next node, since there are two equal highest values in row v_6, we randomly select one node between v_1 and v_4. This process is repeated to the end of the new individual with the length of MaxPOI.

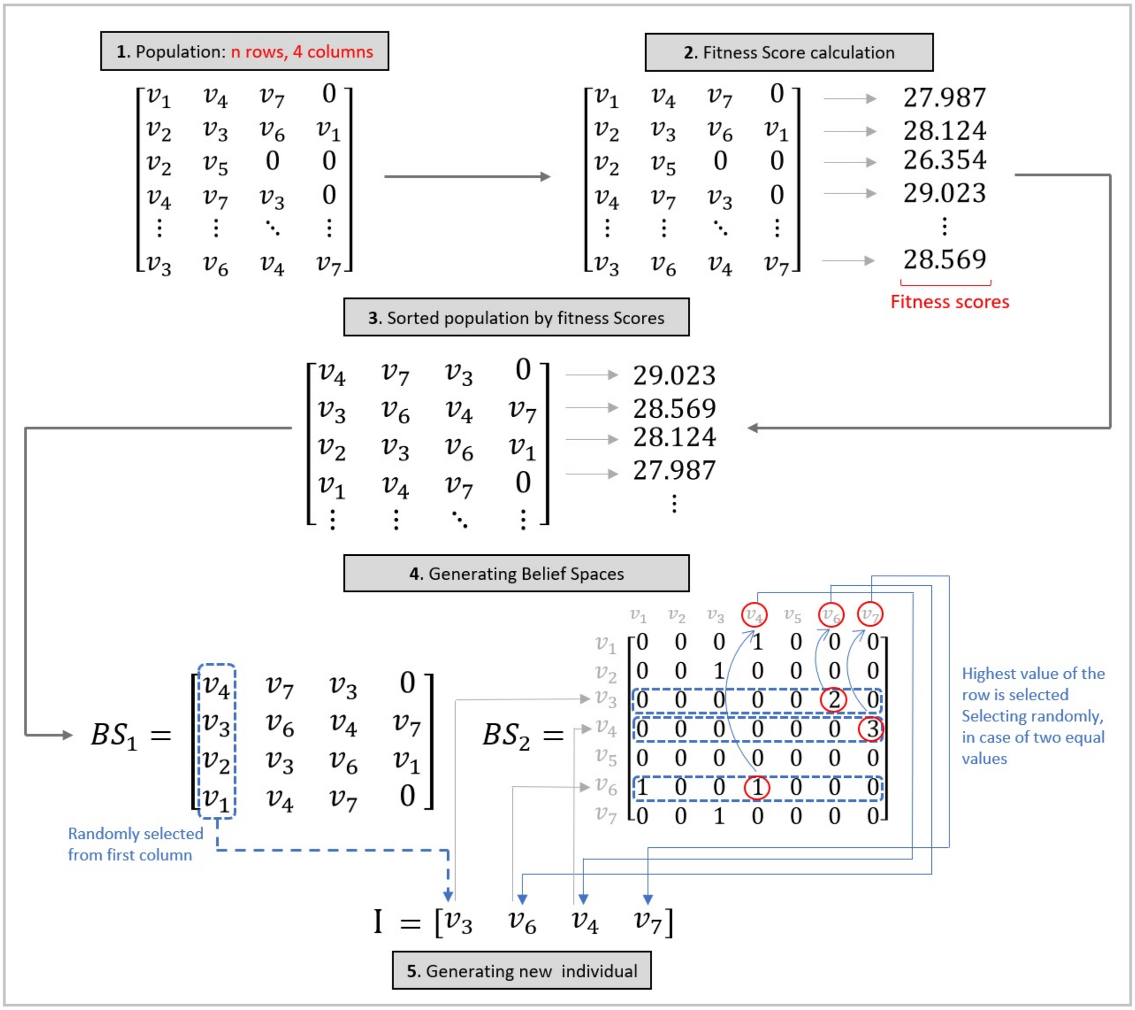

Figure 6: An example of one iteration and generating a new individual from belief space

3.3.4 Crossover and mutation

In the case of using crossover for producing a new individual, two rows of the population are selected randomly. Afterward, the nodes are compared with each other two by two. If both are non-equal to zero, one of them is picked arbitrarily. In other cases, if one of them is zero, the other one is selected, and if both of them are zero, the corresponding node of the individual is set as zero. As depicted in Figure 7, assume I_1 and I_2 are two individuals which are randomly selected from the population.

Then using crossover strategy, a new individual, $newI$, is generated according to the described methodology.

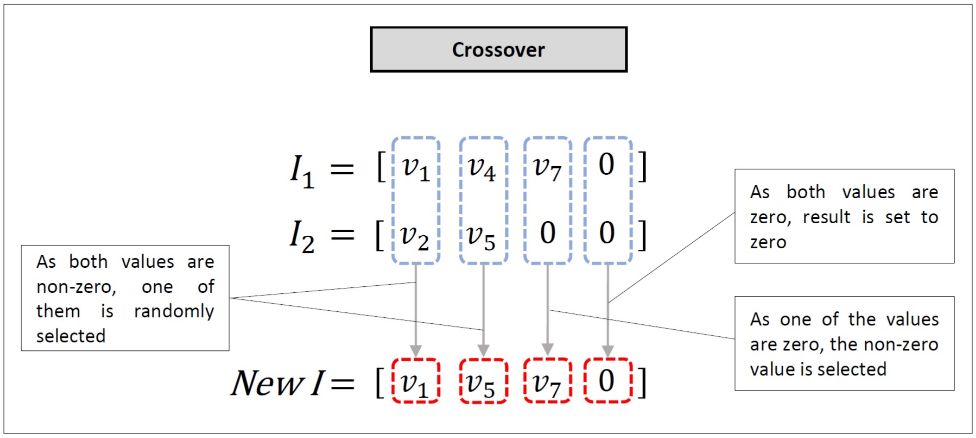

Figure 7: An example of generating a new individual using crossover

In mutation, first, a random size is chosen for the new individual between two and MaxPOI. Next, an utterly random individual with the selected size is generated by picking a set of POIs from V. The remaining nodes at the end of the individual are then set to zero. As a result, the number of non-zero nodes could be between two and MaxPOI.

Moreover, on each iteration, we store the best solution of the previous population in an elite set and add it to the new generation. That is to make sure that the new generation has at least equal quality as the previous one.

Therefore, n new individuals are generated to form a new population. By some probability, they are generated from belief space; in other cases, crossover or mutation are utilized. This process is repeated, and knowledge in the belief space is updated on each iteration. The whole process continues to meet the termination criteria. Finally, the individual with the best fitness score is returned as the algorithm's output.

3.3.5 Proposed algorithm

Algorithm 1 shows the pseudo-code of our approach for finding the set of POIs and their sequence as the best itinerary to be recommended. To have a more readable algorithm, there are two procedures called inside that, including Generate BS_2 and Generate From BS procedures.

As presented in Algorithm 1, a random population is generated in line 1. The fitness score of the individuals in the population is calculated in lines 3-9 and if any individual exceeds the limitations the fitness score is set as -1. The population is sorted by the fitness scores in line 10 and best ones are selected and then form the belief space in lines 11 to 15. Next, a new population is produced using the belief space or crossover and mutation methods in line 16-25. The whole process is repeated for a defined number of times and the final result is returned in line 30.

Algorithm 1 Knowledge-based Itinerary Recommendation Algorithm

Input: Graph G; Time limit $MaxT$; Budget MaxB; Population size n; Iteration number δ; MaxPOI; The proportion of population to build belief space x; Probability of generating individual from belief space $prob1$; Probability of generating individual using crossover $prob2$

Output: Best solution

$P \leftarrow$ Generate n individuals randomly with size $MaxPOI$ as initial population set, considering time and budget constraint: $Cost(I_j) \leq MaxB$, $TotalTime(I_j) \leq MaxT$ where $I_j \in P$;
for $i = 1$ to δ **do**
  for $j = 1$ to n **do**
    if $Cost(P[j]) \leq MaxB$ and $TotalTime(P[j]) \leq MaxT$ **then**
      $FS \leftarrow$ Calculate fitness score $F(P[j])$
    else
      $FS \leftarrow -1$
    end if
  end for
  Sort individuals in P based on their FS
  $P1$ = An empty array
  $el \leftarrow P[0]$ ▷ Keep the elite itinerary
  $kb \leftarrow (x$ * length of P)/100 ▷ Calculate x% of population length
  $BS_1[1, ..., kb] \leftarrow P[1, ..., kb]$ ▷ Transfer best individuals to belief space
  $BS_2 \leftarrow GenerateBS_2(BS_1)$
  for $k = 1$ to $n - 1$ **do**
    if $rand1() \leq prob1$ **then**
      $new\_I \leftarrow GenerateFromBS()$ ▷ generate a new individual from belief space
    else if $rand2() \leq prob2$ **then**
      $new\_I \leftarrow$ generate a new individual using crossover strategy
    else
      $new\_I \leftarrow$ generate a new individual using mutation strategy
    end if
    $P1[k] \leftarrow new_I$
  end for
  $P[1] \leftarrow el$
  $P[2, ..., n] \leftarrow P1$
end for
Calculate fitness scores of P and sort individuals
return $P[1]$

The whole process of extracting knowledge out of the population and creating both BS_1 and BS_2 is outlined in Algorithm 2 in detail. Moreover, the procedure that uses the stored knowledge and generates a new individual from the belief space is presented in Algorithm 3.

Algorithm 2 Generate BS_2 procedure

Input : Belief space matrix BS_1; Total number of POIs $POINum$; Maximum number of visiting POIs in an itinerary $MaxPOI$
Output: Belief space BS_2

procedure $GenerateBS_2(BS_1,\ POINum,\ MaxPOI)$
    Initialize BS_2 as an 2D array $[POINum \times POINum]$
    $n \leftarrow$ number of rows in BS_1
    for $i = 1$ to n **do**
        for $j = 2$ to $MaxPOI$ **do**
            $x \leftarrow BS_1[i][j-1]$
            $y \leftarrow BS_1[i][j]$
            $BS_2[x][y] \leftarrow BS_2[x][y] + 1$
        end for
    end for
    return BS_2
end procedure

Algorithm 3 Generate From BS procedure

Input : Belief space matrices BS_1 and BS_2; Maximum number of visiting POIs in an itinerary $MaxPOI$
Output: Generated individual I

procedure GenerateFromBS($BS_1,\ BS_2,\ MaxPOI$)
    Initialize I as an empty array with size MaxPOI
    $I[1] \leftarrow$ select a POI randomly from $BS_1[1]$
    for $j = 2$ to $MaxPOI$ **do**
        $x \leftarrow$ select one column randomly from highest values of $BS_2[I[j-1]]$
        $newPOI \leftarrow$ select column number of x
        $I[j] \leftarrow newPOI$
    end for
    return I
end procedure

CHAPTER 4

Evaluation

This section gives a full review of the experiments carried out in this study. It starts with an explanation of the experimental setup, then moves on to the conducted experiments, datasets, baseline algorithms and evaluation metrics, followed by results of our algorithm and comparisons with other algorithms. It emphasizes the experiment's results as well as some of the conclusions made from it.

4.1 Setup

In order to evaluate the performance of our model, we have conducted a series of experiments and analyses. All the experiments were executed on a PC with an Intel Core I7 CPU, Windows 10, and 8GB of RAM. For comparing the efficiency of our algorithm with the existing baselines, we implemented them in Python language.

Some parameters play an important role in CA's efficiency and need to be set up. Some experiments are conducted to find the best settings for these parameters. As for the number of iterations that the CA algorithm repeats, 30 and 50 are examined, and 50 is selected with more satisfying results compared with 30. Moreover, we examined different probabilities of generating new individuals from the belief space and the probability of choosing between crossover and mutation. We tested various probabilities and found out that the best setup for these probabilities is 60% for using belief space to create a new individual, and in other cases, 70% for using the crossover method, and 30% for utilizing the mutation approach. Thus, these probabilities are utilized for the reported results.

4.2 Datasets

We use two real-world datasets extracted from the Yahoo! Flickr Creative Commons 100M to evaluate our proposed model, which comprises 100 million photos and videos, for our experiment and analysis. The details and some statistics of the selected datasets are provided in this section.

Flickr User-POI Visits Dataset [4][52]: In this dataset, users and their visits to various POIs in different cities are included along with some information about each POI. The user-POI visits are calculated using geo-tagged YFCC100M Flickr photos that have been mapped to specific POI locations and POI categories and sorted into separate travel sequences (consecutive user-POI visits that differ by 8 hours). The statistics of this dataset, including cities, the number of POIs, the number of users, POI visits, and travel sequences, are presented in Table 4.

City	# POIs	# Users	# POI visits	# Travel sequences
Glasgow	29	601	11,434	2,227
Budapest	39	935	18,513	2,361
Delhi	26	279	3,993	489
Vienna	29	1,155	34,515	3,193

Table 4: Flickr User-POI Visits dataset statistics

Theme Park Attraction Visits Dataset [45]: As shown in Table 5, it includes users and their visits to various attractions at some theme parks. The user-attraction visits are calculated using geo-tagged Flickr photographs taken between August 2007 and August 2017 and were collected using the Flickr API.

Theme Park	# POIs	# Users	# POI Visits	# Travel Sequences
Disneyland	31	3,704	119,987	11,758
California Adventure	25	2,593	57,177	6,907
Magic Kingdom	27	3,342	73,994	8,126

Table 5: Theme Park Attraction Visits dataset statistics

Both mentioned datasets include data about POIs, such as their category, popularity, cost, and distance between each two POI in one city. Moreover, there are some details about the photos taken by the users on their visits, such as their time and sequence.

4.3 Experiments

To assess the performance of the algorithm and the quality of its output in different situations, we defined diverse scenarios and conducted experiments in various settings.

Since our algorithm is able to recommend itineraries for a group of tourists, we created various groups with different characteristics. As shown in Table 7, groups are initialized in three sizes 3, 5, and 10. Ten separate groups have been randomly generated from the list of users for each group size. All the algorithms are applied to these groups, and their efficiency is examined using the metrics that will be defined. Moreover, the other variable is the population size of the proposed cultural algorithm, which can vary in three values of 100, 150, and 200. In addition, as elaborated in Chapter.3, our algorithm inputs the maximum number of visiting POIs in a single itinerary, time limit, and budget. In all experiments, the values of these parameters are as presented in Table 6.

Parameter	**Value**	**Description**
MaxPOI	6	Maximum number of POIs
MaxT	8	Time limit (Hours)
MaxB	1000	Maximum budget (Dollars)

Table 6: Input parameters

Moreover, the factor weights in the fitness function, which are user interest, number of POIs in the individual, total popularity, total visit duration, and total entrance cost, are set as $w_1 = 0.5$, $w_2 = 0.25$, $w_3 = 0.1$, $w_4 = 0.1$, and $w_5 = 0.05$, respectively.

All the experiments are repeated 5 times to have a more reliable outcome, and the average, maximum value, and standard deviation of them are reported in the results. Multiple experiments are also conducted to select the best setting for the proposed algorithm.

Variables	Values
Group sizes	3, 5, 10
Groups	10 different groups in each size
Population sizes	100, 150, 200

Table 7: Experiments variables

4.4 Data pre-processing

As mentioned in the previous sections, the datasets contain data about the photos taken in each POI by the users in different time frames. Consequently, inspired from [28], for extracting the user interest in each POI, we needed to measure the total time that each user spent on them. It is calculated by finding the time difference between the last photo taken and the first one in a sequence. Thus, the time of the first taken photo is considered as the user's arrival time and the last one as the departure time. Next, in order to determine the user interest in each category of POIs, not a specific one, the total time that each user spent in all POIs of the same category is divided by the maximum value of that. All the obtained user interests are also normalized to generate a value between 0 and 1. Therefore, we have the interests of all users in categories that there was a visit history for them.

Furthermore, the time needed to visit each POI was required in order to check that the generated itineraries do not exceed the time limit. For this purpose, the average time that each POI was taken from users to visit is calculated and has been set to the POI as its visit duration. Moreover, the travel time between each POI is obtained by dividing the distance between them by the average traveling speed. According to [3], the traveling speed has been set to 4 km per hour as a leisure walking speed. Therefore, the total time required for a POI equals the sum of its visit duration and the travel time from the previous POI.

There are four cities in the first dataset and three theme parks in the second one that are used for our evaluation. Thus, the performance of the proposed algorithm is examined on seven separate locations. Each city or theme park has a various number of POIs differentiated between 25 and 31 that could affect the algorithm accuracy and run-time.

In addition, some sample groups are needed to perform the algorithm on them and assess the results. So, ten groups of each size (3, 5, 10) are randomly generated from users in both datasets separately. All algorithms are applied to these groups to make a comparison of the quality of the recommended itinerary to the same groups. Group analysis has been conducted on the first dataset (Flickr User-POI Visits dataset) to extract the features of each group. For this purpose, the interests of each user in all groups are extracted to find the intersection of their interests. Thus, we determined the common categories of interest between them. The idea behind this is that the more disagreement in a group, the harder it is to recommend an itinerary to satisfy all of them. This information can help us with further investigation. Therefore, the results of this analysis of groups of different sizes are presented in three tables, Table 8, Table 9, and Table 10. These tables include the number of categories that each user is interested in and their intersection, which means the number of categories that all of them are interested in.

Groups	**User 1 # Cats**	**User 2 # Cats**	**User 3 # Cats**	**# Intersection of Cats**
Group 3-0	5	4	3	2
Group 3-1	4	1	3	0
Group 3-2	5	1	1	0
Group 3-3	3	4	2	2
Group 3-4	5	3	4	2
Group 3-5	6	4	3	2
Group 3-6	8	4	2	2
Group 3-7	4	3	8	3
Group 3-8	4	8	3	3
Group 3-9	3	2	2	0

Table 8: The analysis of groups with size 3 (* Cats = Categories)

Groups	User 1 # Cats	User 2 # Cats	User 3 # Cats	User 4 # Cats	User 5 # Cats	# Intersection of Cats
Group 5-0	5	1	7	11	5	1
Group 5-1	4	5	5	2	5	1
Group 5-2	4	5	9	8	3	2
Group 5-3	3	3	2	2	1	0
Group 5-4	4	5	1	3	3	0
Group 5-5	2	8	8	1	5	0
Group 5-6	3	5	2	6	3	2
Group 5-7	4	5	3	2	9	2
Group 5-8	2	1	2	3	3	0
Group 5-9	3	6	4	2	2	1

Table 9: The analysis of groups with size 5 (* Cats = Categories)

Groups	User 1 # Cats	User 2 # Cats	User 3 # Cats	User 4 # Cats	User 5 # Cats	User 6 # Cats	User 7 # Cats	User 8 # Cats	User 9 # Cats	User 10 # Cats	# Intersection of Cats
Group 10-0	2	3	3	2	6	2	9	4	2	6	0
Group 10-1	2	5	9	1	2	4	4	3	2	4	0
Group 10-2	8	1	1	4	4	1	8	9	2	5	0
Group 10-3	9	11	2	4	2	6	1	4	7	4	0
Group 10-4	1	7	5	1	7	4	2	5	3	4	0
Group 10-5	2	9	1	5	3	2	4	2	4	2	0
Group 10-6	5	2	3	6	6	4	8	4	6	5	1
Group 10-7	3	1	7	6	3	5	7	7	4	4	0
Group 10-8	2	3	4	6	1	3	3	8	8	3	1
Group 10-9	1	4	3	4	6	4	2	3	1	3	0

Table 10: The analysis of groups with size 10 (* Cats = Categories)

4.5 Baseline algorithms

In order to make a comparative study, we assess the performance of our proposed algorithm in comparison with six existing state-of-the-art baseline approaches. In this section, we elaborate on each algorithm.

Genetic [8]: It is a standard genetic algorithm that generates a random set of itineraries as an initial population at first. Then, it endeavors to provide variety on that, using crossover and mutation techniques and keeping the best itineraries on each iteration. The best solution is then presented as an algorithm's output after the predefined number of iterations. The fitness function for evaluating each itinerary in the population is the same as our proposed algorithm.

AGAM [28]: It is an adaptive genetic algorithm that recommends a personalized itinerary using dynamic crossover and mutation probabilities. The original algorithm is proposed for recommending an itinerary to an individual, but the fitness function has been customized to make it comparable with our model.

PERSTOUR [4]: It solves the problem as an integer programming problem. This algorithm recommends itineraries based on the popularity of POIs and time-based and frequency-based visitor interest within a given budget. In this experiment, we user time-based version of this algorithm. The value of η indicates the weight given to either POI popularity or user interest. Thus, we use $\eta = 0.5$ to emphasis on optimizing both POI popularity and time-based user interest.

TLR [34]: It is a Transformer based Learning Recommendation using multi-head attention. It combines an encoder and a decoder to record a personalized correlation between POIs that visits the entire series without the need for sequential propagation.

PersQ [45]: This algorithm is a personalised itinerary recommendation based on the implementation of Monte Carlo Tree Search (MCTS). It has four steps: Selection, Expansion, Simulation, and Back-propagation. The core notion is that game play begins with iterations of random node selection to explore moves, with the results of those moves being recorded. Following that, MCTS moves away from random moves and gradually builds on earlier successes by converging to moves that result in win states during following game plays [45].

GreedyFitness: First, a POI is selected randomly and added to the itinerary. Then, following the greedy strategy, it adds a POI that brings the highest fitness to the itinerary. The remaining POIs are then treated in the same way. This approach focuses on adding each POI according to the factors included in calculating the fitness score. It evaluates the POIs step by step when adding each POI to the itinerary. It adds each POI just considering the following POI while not revising its decisions and making the locally optimal choice at each stage.

4.6 Evaluation metrics

The key method of evaluating an itinerary recommendation system is how well the suggested itineraries meet the needs of tourists and their interests. We assess the efficiency of our algorithm and the baselines, which include itinerary planning recommendations. Our algorithm is based on a cultural algorithm to recommend a tour that includes as many POIs as feasible, maximizes the user's visit duration within a time budget, maximizes overall popularity, and maintains the user's cost on POI entrance to a minimum.

Our proposed algorithm is evaluated and compared with other algorithms from different points of view. Thus, a couple of metrics are employed for this evaluation:

Fitness Score: The fitness score of the recommended itineraries by each algorithm is analyzed and compared with others under similar circumstances and variables.

Time Complexity: A time analysis experiment has been conducted to assess the time complexity of each algorithm and check the superiority of our proposed model.

4.7 Results and Analysis

4.7.1 Population size adjustment

As noted earlier, our proposed algorithm has a population size parameter that is effective in its quality and performance. We designed some experiments to find the best size of the population in order to obtain the best result. The experiments are conducted in three values of 100, 150, and 200 for population size, and on two cities of the Flickr User-POI Visits Dataset. Each experiment is repeated five times, and the maximum value, average, and standard deviation of them are reported.

The best values of average (Avg), maximum (Max), and standard deviation (std) for each city between three population sizes are shown in bold. Table 11 presents the results of running CA on population sizes of 100, 150, and 200 for groups with size 3. Table 12, and Table 13 show the same experiments results for groups with sizes 5 and 10, respectively. Comparing the quality of the solution when population size is 100, 150, and 200 revealed that CA with a population size of 200 outperforms itself with other population sizes in 83.3% of the cases for groups with size 3, 50% for groups with size 5, and 68.3% for groups with size 10. Consequently, size 200 is selected as the best population size with the highest performance for the following experiments.

		Budapest			Delhi		
		n = 100	n=150	n = 200	n = 100	n = 150	n = 200
group 3-0	Max	3.049E-01	**3.052E-01**	3.043E-01	2.435E-01	2.133E-01	**3.043E-01**
	Std	7.672E-03	**1.981E-03**	3.335E-03	1.506E-02	2.673E-03	**3.335E-03**
	Avg	2.947E-01	3.033E-01	**3.005E-01**	2.108E-01	2.095E-01	**3.005E-01**
group 3-1	Max	2.921E-01	2.912E-01	**2.925E-01**	2.350E-01	2.428E-01	**2.925E-01**
	Std	3.783E-03	3.455E-03	**1.473E-03**	3.263E-02	1.203E-02	**1.473E-03**
	Avg	2.880E-01	2.878E-01	**2.900E-01**	1.965E-01	2.327E-01	**2.900E-01**
group 3-2	Max	**2.885E-01**	2.862E-01	2.866E-01	2.407E-01	2.224E-01	**2.866E-01**
	Std	6.447E-03	4.223E-03	**3.166E-03**	2.862E-02	1.806E-02	**3.166E-03**
	Avg	**2.826E-01**	2.793E-01	2.825E-01	2.026E-01	2.097E-01	**2.825E-01**
group 3-3	Max	2.883E-01	2.898E-01	**2.895E-01**	2.550E-01	2.176E-01	**2.895E-01**
	Std	5.816E-03	4.674E-03	**3.347E-03**	2.305E-02	1.016E-02	**3.347E-03**
	Avg	2.814E-01	2.838E-01	**2.866E-01**	2.089E-01	1.996E-01	**2.866E-01**
group 3-4	Max	2.814E-01	**2.852E-01**	2.846E-01	2.413E-01	2.449E-01	**2.846E-01**
	Std	5.548E-03	**1.707E-03**	3.400E-03	2.849E-02	3.858E-02	**3.400E-03**
	Avg	2.773E-01	**2.826E-01**	2.806E-01	2.007E-01	2.121E-01	**2.806E-01**
group 3-5	Max	2.862E-01	2.853E-01	**2.864E-01**	2.435E-01	2.428E-01	**2.864E-01**
	Std	1.039E-02	8.986E-03	**3.794E-03**	2.783E-02	2.176E-02	**3.794E-03**
	Avg	2.797E-01	2.781E-01	**2.828E-01**	2.031E-01	2.217E-01	**2.828E-01**
group 3-6	Max	3.002E-01	3.035E-01	**3.049E-01**	2.401E-01	2.389E-01	**3.049E-01**
	Std	**1.582E-03**	2.657E-03	1.593E-03	2.198E-02	1.142E-02	**1.593E-03**
	Avg	2.987E-01	3.006E-01	**3.032E-01**	2.037E-01	2.302E-01	**3.032E-01**
group 3-7	Max	2.910E-01	2.885E-01	**2.936E-01**	2.321E-01	2.393E-01	**2.936E-01**
	Std	4.138E-03	**2.874E-03**	2.985E-03	2.756E-02	2.861E-02	**2.985E-03**
	Avg	2.871E-01	2.863E-01	**2.884E-01**	2.005E-01	1.979E-01	**2.884E-01**
group 3-8	Max	2.978E-01	3.015E-01	**3.026E-01**	2.428E-01	2.360E-01	**3.026E-01**
	Std	**3.084E-03**	3.371E-03	3.161E-03	3.321E-02	2.434E-02	**3.161E-03**
	Avg	2.946E-01	2.978E-01	**2.988E-01**	1.914E-01	2.217E-01	**2.988E-01**
group 3-9	Max	3.019E-01	**3.030E-01**	**3.030E-01**	2.451E-01	2.410E-01	**3.030E-01**
	Std	8.011E-03	4.291E-03	**7.887E-04**	3.704E-02	3.202E-02	**7.887E-04**
	Avg	2.956E-01	2.978E-01	**3.023E-01**	1.989E-01	2.056E-01	**3.023E-01**

Table 11: CA result comparison for groups with size 3

		Budapest			**Delhi**		
		n = 100	**n=150**	**n = 200**	**n = 100**	**n = 150**	n = 200
group 5-0	**Max**	2.831E-01	**2.893E-01**	2.866E-01	2.383E-01	**2.433E-01**	2.373E-01
	Std	5.145E-03	3.807E-03	**2.355E-03**	2.789E-02	4.522E-03	**6.989E-03**
	Avg	2.775E-01	**2.862E-01**	2.841E-01	2.163E-01	**2.363E-01**	2.307E-01
group 5-1	**Max**	3.054E-01	**3.059E-01**	3.056E-01	**2.437E-01**	2.430E-01	2.405E-01
	Std	7.808E-03	8.466E-03	**3.295E-03**	3.721E-02	**1.912E-02**	3.036E-02
	Avg	2.989E-01	2.995E-01	**3.023E-01**	2.104E-01	**2.285E-01**	2.040E-01
group 5-2	**Max**	2.801E-01	2.802E-01	**2.825E-01**	**2.410E-01**	2.389E-01	2.305E-01
	Std	4.441E-03	2.391E-03	**2.819E-03**	3.862E-02	2.745E-02	**2.467E-02**
	Avg	2.748E-01	2.785E-01	**2.802E-01**	2.076E-01	1.993E-01	**2.149E-01**
group 5-3	**Max**	3.019E-01	**3.026E-01**	3.016E-01	**2.451E-01**	2.281E-01	2.431E-01
	Std	1.871E-03	**1.713E-03**	7.259E-03	3.016E-02	2.141E-02	**8.596E-03**
	Avg	2.987E-01	**3.011E-01**	2.964E-01	2.151E-01	2.095E-01	**2.318E-01**
group 5-4	**Max**	2.717E-01	2.734E-01	**2.737E-01**	2.393E-01	2.497E-01	**2.638E-01**
	Std	2.585E-03	3.385E-03	**2.458E-03**	**1.934E-02**	2.141E-02	2.323E-02
	Avg	2.678E-01	2.692E-01	**2.703E-01**	2.065E-01	2.141E-01	**2.315E-01**
group 5-5	**Max**	2.802E-01	**2.787E-01**	2.817E-01	2.413E-01	2.416E-01	**2.450E-01**
	Std	2.508E-03	4.151E-03	**1.230E-03**	1.243E-02	**7.152E-03**	8.904E-03
	Avg	2.784E-01	2.751E-01	**2.805E-01**	2.282E-01	2.315E-01	**2.344E-01**
group 5-6	**Max**	**2.777E-01**	2.770E-01	2.775E-01	2.389E-01	**2.510E-01**	**2.510E-01**
	Std	3.040E-03	**1.922E-03**	3.175E-03	2.718E-02	3.055E-02	**3.283E-02**
	Avg	2.739E-01	**2.748E-01**	2.737E-01	2.082E-01	2.115E-01	**2.157E-01**
group 5-7	**Max**	3.004E-01	**3.010E-01**	3.009E-01	2.336E-01	**2.469E-01**	2.360E-01
	Std	1.714E-03	**8.618E-04**	2.040E-03	2.620E-02	1.246E-02	**2.877E-02**
	Avg	2.989E-01	**3.001E-01**	2.992E-01	1.982E-01	**2.353E-01**	2.033E-01
group 5-8	**Max**	2.730E-01	**2.767E-01**	2.763E-01	2.211E-01	**2.537E-01**	2.416E-01
	Std	4.522E-03	**2.255E-03**	4.237E-03	**2.066E-02**	3.411E-02	2.423E-02
	Avg	2.678E-01	**2.745E-01**	2.731E-01	2.009E-01	2.077E-01	**2.153E-01**
group 5-9	**Max**	2.749E-01	2.785E-01	**2.776E-01**	2.409E-01	2.356E-01	**2.432E-01**
	Std	2.371E-03	1.258E-03	**1.772E-03**	2.130E-02	3.419E-02	**6.073E-03**
	Avg	2.718E-01	**2.768E-01**	2.760E-01	2.156E-01	1.915E-01	**2.386E-01**

Table 12: CA result comparison for groups with size 5

		Budapest			**Delhi**		
		n = 100	**n=150**	**n = 200**	**n = 100**	**n = 150**	n = 200
group 10-0	**Max**	2.710E-01	2.695E-01	**2.716E-01**	2.238E-01	**2.427E-01**	**2.427E-01**
	Std	4.084E-03	3.699E-03	**2.429E-03**	2.696E-02	2.502E-02	**1.442E-02**
	Avg	2.682E-01	**2.695E-01**	**2.695E-01**	1.933E-01	**2.224E-01**	2.213E-01
group 10-1	**Max**	2.770E-01	2.748E-01	**2.780E-01**	1.896E-01	2.278E-01	**2.540E-01**
	Std	**3.081E-03**	3.148E-03	3.452E-03	**1.096E-02**	2.658E-02	3.245E-02
	Avg	2.722E-01	2.748E-01	**2.754E-01**	1.733E-01	1.977E-01	**2.104E-01**
group 10-2	**Max**	2.811E-01	2.841E-01	**2.842E-01**	**2.437E-01**	**2.437E-01**	2.429E-01
	Std	4.477E-03	3.570E-03	**3.144E-03**	1.611E-02	2.984E-02	**7.436E-03**
	Avg	2.768E-01	**2.841E-01**	2.813E-01	2.192E-01	2.195E-01	**2.371E-01**
group 10-3	**Max**	2.765E-01	2.733E-01	**2.772E-01**	**2.417E-01**	2.409E-01	2.349E-01
	Std	4.411E-03	2.457E-03	**1.469E-03**	3.281E-02	2.968E-02	**5.215E-03**
	Avg	2.714E-01	2.733E-01	**2.748E-01**	2.145E-01	2.065E-01	**2.307E-01**
group 10-4	**Max**	2.810E-01	2.694E-01	**2.817E-01**	2.171E-01	2.362E-01	**2.405E-01**
	Std	**2.694E-03**	3.214E-03	6.602E-03	2.327E-02	3.047E-02	**2.146E-02**
	Avg	**2.790E-01**	2.694E-01	2.767E-01	1.992E-01	2.065E-01	**2.153E-01**
group 10-5	**Max**	2.774E-01	**2.785E-01**	2.781E-01	2.356E-01	**2.470E-01**	2.358E-01
	Std	4.610E-03	3.749E-03	**2.442E-03**	**2.917E-02**	4.311E-02	3.122E-02
	Avg	2.738E-01	**2.785E-01**	2.757E-01	1.910E-01	**2.108E-01**	2.043E-01
group 10-6	**Max**	2.768E-01	2.771E-01	**2.805E-01**	2.418E-01	**2.431E-01**	2.428E-01
	Std	4.176E-03	3.564E-03	**2.034E-03**	3.046E-02	**2.790E-02**	4.073E-02
	Avg	2.715E-01	2.771E-01	**2.788E-01**	2.022E-01	**2.224E-01**	2.106E-01
group 10-7	**Max**	2.699E-01	2.611E-01	**2.724E-01**	2.193E-01	**2.413E-01**	2.351E-01
	Std	7.834E-03	2.356E-03	**1.869E-03**	2.992E-02	8.578E-03	**1.931E-02**
	Avg	2.626E-01	2.611E-01	**2.702E-01**	1.801E-01	**2.289E-01**	2.228E-01
group 10-8	**Max**	2.892E-01	2.809E-01	**2.907E-01**	2.202E-01	2.076E-01	**2.278E-01**
	Std	5.422E-03	2.352E-03	**1.417E-03**	1.906E-02	**1.402E-02**	1.968E-02
	Avg	2.833E-01	2.809E-01	**2.893E-01**	2.025E-01	1.892E-01	**2.056E-01**
group 10-9	**Max**	2.808E-01	2.779E-01	**2.814E-01**	2.360E-01	2.286E-01	**2.430E-01**
	Std	3.114E-03	2.355E-03	**2.016E-03**	3.322E-02	1.620E-02	**1.086E-02**
	Avg	2.782E-01	2.779E-01	**2.790E-01**	1.838E-01	2.033E-01	**2.366E-01**

Table 13: CA result comparison for groups with size 10

4.7.2 Result comparison with baselines

We ran all the algorithms on both mentioned datasets and compared the results. The fitness score shows how well the suggested itinerary corresponds to user preferences. The following tables presents the fitness score of the recommended itinerary by each of the algorithms on Flickr User-POI Visits dataset, as our first dataset. Table 14 shows the comparison of CA and other approaches in four cities from this dataset, including Budapest, Delhi, Glasgow and Vienna in groups with size 3. Table 15 and Table 16 present the same experiment on groups with sizes 5 and 10, respectively.

Flickr User-POI Visits dataset								
City	**Group**	**CA**	**GA**	**PersTour**	**AGAM**	**TLR**	**PersQ**	**GreedyFitness**
Budapest	**Group 3-0**	**3.005E-01**	3.022E-01	2.955E-01	2.655E-01	2.940E-01	2.552E-01	2.733E-01
	Group 3-1	**2.900E-01**	2.864E-01	2.828E-01	2.618E-01	2.835E-01	2.468E-01	2.800E-01
	Group 3-2	2.825E-01	2.787E-01	2.638E-01	2.520E-01	**2.854E-01**	2.588E-01	2.417E-01
	Group 3-3	**2.866E-01**	2.834E-01	2.765E-01	2.539E-01	2.819E-01	2.497E-01	2.182E-01
	Group 3-4	**2.806E-01**	2.788E-01	2.573E-01	2.472E-01	2.797E-01	2.157E-01	2.596E-01
	Group 3-5	**2.828E-01**	2.795E-01	2.696E-01	2.447E-01	2.578E-01	2.209E-01	2.452E-01
	Group 3-6	3.032E-01	2.998E-01	2.922E-01	2.635E-01	**3.044E-01**	2.476E-01	2.992E-01
	Group 3-7	**2.884E-01**	2.886E-01	2.755E-01	2.602E-01	2.859E-01	2.587E-01	2.487E-01
	Group 3-8	**2.988E-01**	2.968E-01	2.763E-01	2.602E-01	2.975E-01	2.271E-01	2.743E-01
	Group 3-9	**3.023E-01**	2.960E-01	2.940E-01	2.757E-01	2.928E-01	2.529E-01	2.759E-01
Delhi	**Group 3-0**	**2.125E-01**	2.111E-01	1.711E-01	2.095E-01	1.933E-01	1.588E-01	1.513E-01
	Group 3-1	**2.005E-01**	1.955E-01	1.697E-01	2.146E-01	1.889E-01	1.588E-01	1.949E-01
	Group 3-2	**2.365E-01**	2.031E-01	1.699E-01	2.223E-01	1.895E-01	1.588E-01	1.519E-01
	Group 3-3	**2.153E-01**	2.078E-01	1.709E-01	2.334E-01	1.927E-01	1.588E-01	1.993E-01
	Group 3-4	**2.098E-01**	1.977E-01	1.702E-01	2.219E-01	1.903E-01	1.588E-01	2.187E-01
	Group 3-5	**2.274E-01**	1.980E-01	1.706E-01	2.450E-01	1.918E-01	1.588E-01	2.089E-01
	Group 3-6	**2.051E-01**	2.054E-01	1.695E-01	2.226E-01	1.884E-01	1.588E-01	1.947E-01
	Group 3-7	**2.159E-01**	1.989E-01	1.695E-01	2.295E-01	1.884E-01	1.588E-01	1.889E-01
	Group 3-8	**2.362E-01**	1.955E-01	1.701E-01	2.126E-01	1.902E-01	1.588E-01	1.866E-01
	Group 3-9	**2.184E-01**	2.021E-01	1.696E-01	2.143E-01	1.886E-01	1.588E-01	1.651E-01
Glasgow	**Group 3-0**	**2.473E-01**	2.400E-01	1.970E-01	2.204E-01	2.302E-01	1.647E-01	1.895E-01
	Group 3-1	2.547E-01	2.428E-01	1.970E-01	2.147E-01	**2.570E-01**	1.658E-01	1.989E-01
	Group 3-2	2.641E-01	2.679E-01	1.970E-01	2.376E-01	**2.723E-01**	2.080E-01	2.101E-01
	Group 3-3	**2.491E-01**	2.495E-01	1.970E-01	2.211E-01	2.290E-01	2.032E-01	2.195E-01
	Group 3-4	2.308E-01	2.240E-01	2.018E-01	2.153E-01	**2.310E-01**	1.646E-01	1.557E-01
	Group 3-5	**2.649E-01**	2.659E-01	2.127E-01	2.289E-01	2.342E-01	1.668E-01	1.638E-01
	Group 3-6	2.228E-01	2.158E-01	2.019E-01	2.075E-01	**2.293E-01**	1.653E-01	1.911E-01
	Group 3-7	**2.605E-01**	2.625E-01	1.990E-01	2.145E-01	2.593E-01	1.772E-01	1.747E-01
	Group 3-8	**2.272E-01**	2.200E-01	1.983E-01	2.129E-01	2.269E-01	1.647E-01	2.058E-01
	Group 3-9	2.397E-01	2.323E-01	1.970E-01	2.166E-01	**2.519E-01**	1.656E-01	1.679E-01
Vienna	**Group 3-0**	2.598E-01	2.593E-01	2.366E-01	2.320E-01	**2.611E-01**	1.774E-01	1.697E-01
	Group 3-1	2.499E-01	2.573E-01	2.397E-01	2.309E-01	**2.567E-01**	1.899E-01	1.998E-01
	Group 3-2	**2.850E-01**	2.864E-01	2.516E-01	2.357E-01	2.716E-01	1.948E-01	1.730E-01
	Group 3-3	**2.491E-01**	2.412E-01	2.293E-01	2.099E-01	2.465E-01	1.727E-01	1.857E-01
	Group 3-4	**2.324E-01**	2.277E-01	2.132E-01	2.074E-01	2.267E-01	1.958E-01	2.078E-01
	Group 3-5	2.219E-01	2.174E-01	2.218E-01	2.034E-01	**2.238E-01**	1.782E-01	1.841E-01
	Group 3-6	**2.564E-01**	2.530E-01	2.072E-01	2.230E-01	2.551E-01	1.916E-01	2.232E-01
	Group 3-7	**2.886E-01**	2.870E-01	2.544E-01	2.288E-01	2.717E-01	1.960E-01	1.853E-01
	Group 3-8	**2.481E-01**	2.452E-01	2.125E-01	2.089E-01	2.431E-01	1.921E-01	1.958E-01
	Group 3-9	2.603E-01	**2.685E-01**	2.338E-01	2.208E-01	2.594E-01	1.875E-01	1.969E-01

Table 14: Comparison between our algorithm and various baselines in terms of the fitness score of the recommended itinerary on groups with size 3 of the first dataset

Flickr User-POI Visits dataset								
City	**Group**	**CA**	**GA**	**PersTour**	**AGAM**	**TLR**	**PersQ**	**GreedyFitness**
Budapest	**Group 5-0**	2.841E-01	2.798E-01	2.623E-01	2.618E-01	**2.864E-01**	2.209E-01	2.601E-01
	Group 5-1	**3.023E-01**	3.028E-01	2.975E-01	2.769E-01	2.963E-01	2.662E-01	2.728E-01
	Group 5-2	**2.802E-01**	2.782E-01	2.573E-01	2.391E-01	2.801E-01	2.246E-01	2.585E-01
	Group 5-3	**2.964E-01**	2.953E-01	2.950E-01	2.608E-01	2.944E-01	2.584E-01	2.822E-01
	Group 5-4	**2.703E-01**	2.687E-01	2.460E-01	2.453E-01	2.690E-01	2.294E-01	2.481E-01
	Group 5-5	**2.805E-01**	2.804E-01	2.688E-01	2.530E-01	2.784E-01	2.315E-01	2.786E-01
	Group 5-6	2.737E-01	2.725E-01	2.555E-01	2.435E-01	**2.756E-01**	2.169E-01	2.494E-01
	Group 5-7	**2.992E-01**	2.986E-01	2.926E-01	2.593E-01	2.919E-01	2.574E-01	2.717E-01
	Group 5-8	**2.731E-01**	2.719E-01	2.634E-01	2.474E-01	2.677E-01	2.318E-01	2.516E-01
	Group 5-9	2.760E-01	2.754E-01	2.649E-01	2.480E-01	**2.763E-01**	2.320E-01	2.754E-01
Delhi	**Group 5-0**	**2.307E-01**	2.017E-01	2.623E-01	2.618E-01	1.888E-01	1.597E-01	2.601E-01
	Group 5-1	**2.040E-01**	1.976E-01	2.975E-01	2.769E-01	1.887E-01	1.588E-01	2.728E-01
	Group 5-2	**2.149E-01**	1.966E-01	2.573E-01	2.391E-01	1.883E-01	1.588E-01	2.585E-01
	Group 5-3	**2.318E-01**	1.953E-01	2.950E-01	2.608E-01	1.883E-01	1.588E-01	2.822E-01
	Group 5-4	**2.315E-01**	2.022E-01	2.460E-01	2.453E-01	1.899E-01	1.588E-01	2.481E-01
	Group 5-5	**2.344E-01**	1.992E-01	2.688E-01	2.530E-01	1.888E-01	1.588E-01	2.786E-01
	Group 5-6	**2.157E-01**	1.908E-01	2.555E-01	2.435E-01	1.893E-01	1.588E-01	2.494E-01
	Group 5-7	**2.033E-01**	1.980E-01	2.926E-01	2.593E-01	1.884E-01	1.588E-01	2.717E-01
	Group 5-8	**2.153E-01**	2.011E-01	2.634E-01	2.474E-01	1.893E-01	1.588E-01	2.516E-01
	Group 5-9	**2.386E-01**	2.010E-01	2.649E-01	2.480E-01	1.882E-01	1.588E-01	2.754E-01
Glasgow	**Group 5-0**	2.322E-01	2.260E-01	2.027E-01	2.211E-01	**2.403E-01**	1.798E-01	2.055E-01
	Group 5-1	**2.281E-01**	2.274E-01	1.971E-01	2.236E-01	2.233E-01	1.825E-01	2.236E-01
	Group 5-2	2.344E-01	2.245E-01	2.025E-01	2.170E-01	**2.441E-01**	1.716E-01	2.057E-01
	Group 5-3	2.182E-01	2.124E-01	1.974E-01	2.110E-01	**2.264E-01**	1.644E-01	1.866E-01
	Group 5-4	**2.452E-01**	2.377E-01	1.976E-01	2.256E-01	2.421E-01	1.804E-01	2.037E-01
	Group 5-5	2.228E-01	2.168E-01	1.977E-01	2.159E-01	**2.298E-01**	1.686E-01	2.147E-01
	Group 5-6	**2.381E-01**	2.244E-01	2.164E-01	2.182E-01	2.305E-01	1.676E-01	1.498E-01
	Group 5-7	2.214E-01	2.178E-01	1.971E-01	2.109E-01	**2.293E-01**	1.700E-01	2.076E-01
	Group 5-8	2.375E-01	2.326E-01	1.986E-01	2.160E-01	**2.456E-01**	1.765E-01	2.315E-01
	Group 5-9	**2.329E-01**	2.321E-01	2.165E-01	2.232E-01	2.269E-01	1.922E-01	1.807E-01
Vienna	**Group 5-0**	**2.359E-01**	2.351E-01	2.108E-01	2.108E-01	2.326E-01	1.825E-01	1.856E-01
	Group 5-1	**2.714E-01**	2.766E-01	2.201E-01	2.305E-01	2.671E-01	1.745E-01	1.967E-01
	Group 5-2	2.363E-01	2.322E-01	2.162E-01	2.103E-01	**2.369E-01**	1.817E-01	2.020E-01
	Group 5-3	**2.627E-01**	2.650E-01	2.157E-01	2.255E-01	2.618E-01	1.824E-01	2.061E-01
	Group 5-4	2.298E-01	2.320E-01	2.311E-01	2.218E-01	**2.339E-01**	1.784E-01	1.963E-01
	Group 5-5	2.362E-01	2.372E-01	2.072E-01	2.217E-01	**2.417E-01**	1.825E-01	1.852E-01
	Group 5-6	**2.345E-01**	2.286E-01	2.161E-01	2.041E-01	2.303E-01	1.836E-01	1.971E-01
	Group 5-7	**2.609E-01**	2.632E-01	2.146E-01	2.168E-01	2.522E-01	1.807E-01	1.906E-01
	Group 5-8	2.352E-01	2.299E-01	2.206E-01	2.095E-01	**2.382E-01**	1.768E-01	2.131E-01
	Group 5-9	**2.390E-01**	2.325E-01	2.024E-01	2.076E-01	2.384E-01	1.825E-01	1.797E-01

Table 15: Comparison between our algorithm and various baselines in terms of the fitness score of the recommended itinerary on groups with size 5 of the first dataset

City	Group	CA	GA	PersTour	AGAM	TLR	PersQ	GreedyFitness
Flickr User-POI Visits dataset								
Budapest	**Group 10-0**	**2.695E-01**	2.690E-01	2.583E-01	2.505E-01	2.680E-01	2.285E-01	2.699E-01
	Group 10-1	2.754E-01	2.740E-01	2.518E-01	2.500E-01	**2.759E-01**	2.226E-01	2.478E-01
	Group 10-2	2.813E-01	2.824E-01	2.587E-01	2.468E-01	**2.828E-01**	2.314E-01	2.538E-01
	Group 10-3	**2.748E-01**	2.743E-01	2.630E-01	2.485E-01	2.726E-01	2.281E-01	2.748E-01
	Group 10-4	2.767E-01	2.751E-01	2.572E-01	2.435E-01	**2.812E-01**	2.238E-01	2.559E-01
	Group 10-5	**2.757E-01**	2.756E-01	2.656E-01	2.549E-01	2.696E-01	2.326E-01	2.499E-01
	Group 10-6	**2.788E-01**	2.778E-01	2.674E-01	2.524E-01	2.710E-01	2.336E-01	2.514E-01
	Group 10-7	**2.702E-01**	2.668E-01	2.544E-01	2.400E-01	2.698E-01	2.196E-01	2.670E-01
	Group 10-8	**2.893E-01**	2.859E-01	2.810E-01	2.560E-01	2.824E-01	2.496E-01	2.577E-01
	Group 10-9	**2.790E-01**	2.800E-01	2.683E-01	2.438E-01	2.778E-01	2.344E-01	2.792E-01
Delhi	**Group 10-0**	**2.213E-01**	1.920E-01	1.694E-01	2.088E-01	1.882E-01	1.588E-01	1.651E-01
	Group 10-1	**2.104E-01**	1.984E-01	1.695E-01	2.030E-01	1.885E-01	1.588E-01	1.938E-01
	Group 10-2	**2.371E-01**	1.987E-01	1.695E-01	2.275E-01	1.883E-01	1.588E-01	1.502E-01
	Group 10-3	**2.307E-01**	1.819E-01	1.695E-01	2.304E-01	1.883E-01	1.598E-01	1.658E-01
	Group 10-4	2.153E-01	1.965E-01	1.695E-01	**2.308E-01**	1.883E-01	1.588E-01	1.502E-01
	Group 10-5	**2.043E-01**	1.854E-01	1.695E-01	2.033E-01	1.884E-01	1.594E-01	1.944E-01
	Group 10-6	2.106E-01	1.891E-01	1.695E-01	**2.303E-01**	1.884E-01	1.588E-01	1.938E-01
	Group 10-7	**2.228E-01**	1.998E-01	1.695E-01	2.162E-01	1.882E-01	1.588E-01	1.651E-01
	Group 10-8	**2.056E-01**	2.016E-01	1.695E-01	2.015E-01	1.884E-01	1.588E-01	1.938E-01
	Group 10-9	**2.366E-01**	1.893E-01	1.695E-01	2.260E-01	1.882E-01	1.588E-01	1.651E-01
Glasgow	**Group 10-0**	2.377E-01	2.294E-01	1.975E-01	2.233E-01	**2.480E-01**	1.846E-01	2.220E-01
	Group 10-1	**2.319E-01**	2.247E-01	1.976E-01	2.176E-01	2.282E-01	1.721E-01	1.898E-01
	Group 10-2	2.342E-01	2.305E-01	2.002E-01	2.182E-01	**2.498E-01**	1.699E-01	1.593E-01
	Group 10-3	2.343E-01	2.261E-01	2.003E-01	2.147E-01	**2.452E-01**	1.654E-01	2.055E-01
	Group 10-4	**2.446E-01**	2.256E-01	2.073E-01	2.157E-01	2.435E-01	1.657E-01	1.768E-01
	Group 10-5	2.346E-01	2.303E-01	1.972E-01	2.240E-01	**2.490E-01**	1.781E-01	2.057E-01
	Group 10-6	**2.426E-01**	2.230E-01	1.971E-01	2.159E-01	2.412E-01	1.688E-01	2.053E-01
	Group 10-7	2.332E-01	2.255E-01	2.019E-01	2.065E-01	**2.302E-01**	1.794E-01	2.107E-01
	Group 10-8	**2.331E-01**	2.236E-01	1.971E-01	2.130E-01	2.321E-01	1.736E-01	2.203E-01
	Group 10-9	**2.415E-01**	2.200E-01	2.015E-01	2.132E-01	**2.394E-01**	1.678E-01	2.053E-01
Vienna	**Group 10-0**	**2.403E-01**	2.292E-01	2.179E-01	2.141E-01	2.392E-01	1.760E-01	1.675E-01
	Group 10-1	**2.308E-01**	2.219E-01	2.173E-01	2.099E-01	2.288E-01	1.774E-01	1.924E-01
	Group 10-2	2.363E-01	2.360E-01	2.199E-01	2.123E-01	**2.413E-01**	1.786E-01	2.000E-01
	Group 10-3	2.375E-01	2.363E-01	2.148E-01	2.090E-01	**2.395E-01**	1.770E-01	2.122E-01
	Group 10-4	**2.390E-01**	2.347E-01	2.131E-01	2.141E-01	2.344E-01	1.767E-01	1.997E-01
	Group 10-5	**2.399E-01**	2.367E-01	2.188E-01	2.190E-01	2.321E-01	1.738E-01	2.157E-01
	Group 10-6	**2.429E-01**	2.369E-01	2.155E-01	2.098E-01	2.337E-01	1.893E-01	2.162E-01
	Group 10-7	2.294E-01	2.275E-01	2.032E-01	2.146E-01	**2.316E-01**	1.769E-01	1.978E-01
	Group 10-8	2.527E-01	2.543E-01	2.117E-01	2.195E-01	**2.561E-01**	1.762E-01	1.848E-01
	Group 10-9	**2.510E-01**	2.382E-01	2.087E-01	2.202E-01	2.444E-01	1.762E-01	2.169E-01

Table 16: Comparison between our algorithm and various baselines in terms of the fitness score of the recommended itinerary on groups with size 10 of the first dataset

As shown in tables 14, 15, and 16, the results of applying all algorithms on the first dataset revealed that our proposed algorithm (CA) surpasses other baselines in 72.5% of the cases for groups with size 3, 67.5% for groups with size 5, and 65% for groups with size 10.

Comparing the obtained results for each city revealed that the best solution quality is for Budapest. Further analysis depicted that the category of the most popular POIs in this city is cultural, which is the same as the intersection of interests of group members in some of the groups. Thus, it is much easier for the algorithm to provide a satisfying itinerary for the generated groups in this city, as it is more likely to find a cultural touristic spot in this city than others. Moreover, Delhi is the only city where our proposed algorithms could beat all the other baselines when recommending an itinerary for all groups with sizes 3 and 5. The reason behind that could be the fact that in Delhi, there is no cultural POI, and it can be derived that CA performs more efficiently when running in the situation that any of the group members have no interest in the target region. Furthermore, considering the cases where other algorithms performed better, TLR had the best performance after CA, with at least 25% of cases recommending the best solution.

Moreover, similar experiments are conducted on three regions from the Theme Part Attraction Visits dataset as our second dataset, including Disneyland, California Adventure, and Magic Kingdom. Table 17 presents the results of running CA and all the other algorithms on these regions for groups with size 3. Table 18 and Table 19 contain the results of a similar experiment for groups with sizes 5 and 10, respectively.

Theme Park Attraction Visits dataset								
City	**Group**	**CA**	**GA**	**PersTour**	**AGAM**	**TLR**	**PersQ**	**GreedyFitness**
Disland	**Group 3-0**	2.137E-01	**2.156E-01**	1.435E-01	1.481E-01	1.385E-01	1.606E-01	1.657E-01
	Group 3-1	2.744E-01	**2.769E-01**	2.089E-01	2.116E-01	1.690E-01	1.922E-01	2.769E-01
	Group 3-2	**2.286E-01**	2.186E-01	1.694E-01	1.588E-01	1.485E-01	1.693E-01	1.983E-01
	Group 3-3	**2.586E-01**	**2.586E-01**	1.719E-01	1.805E-01	1.510E-01	1.874E-01	1.823E-01
	Group 3-4	**2.200E-01**	2.181E-01	1.678E-01	1.667E-01	1.562E-01	1.782E-01	2.173E-01
	Group 3-5	**2.564E-01**	2.554E-01	1.739E-01	1.770E-01	1.599E-01	2.065E-01	2.554E-01
	Group 3-6	2.686E-01	**2.691E-01**	1.817E-01	1.748E-01	1.648E-01	2.164E-01	2.691E-01
	Group 3-7	**2.544E-01**	2.538E-01	1.733E-01	1.685E-01	1.587E-01	2.053E-01	2.538E-01
	Group 3-8	**2.752E-01**	2.607E-01	1.797E-01	1.731E-01	1.621E-01	2.140E-01	2.657E-01
	Group 3-9	**2.474E-01**	2.374E-01	1.710E-01	1.750E-01	1.642E-01	2.007E-01	2.474E-01
CaliAdv	**Group 3-0**	**2.044E-01**	2.029E-01	1.840E-01	1.588E-01	2.071E-01	1.460E-01	1.920E-01
	Group 3-1	2.092E-01	**2.120E-01**	1.917E-01	1.775E-01	2.117E-01	1.487E-01	1.931E-01
	Group 3-2	**2.365E-01**	2.348E-01	1.887E-01	1.826E-01	2.201E-01	1.727E-01	1.787E-01
	Group 3-3	**2.368E-01**	2.293E-01	2.010E-01	1.841E-01	2.186E-01	1.625E-01	1.920E-01
	Group 3-4	**2.414E-01**	**2.414E-01**	1.901E-01	1.831E-01	2.237E-01	1.468E-01	1.756E-01
	Group 3-5	**2.652E-01**	2.550E-01	2.118E-01	1.752E-01	2.282E-01	1.482E-01	2.058E-01
	Group 3-6	2.354E-01	**2.360E-01**	2.067E-01	2.020E-01	2.163E-01	1.553E-01	1.922E-01
	Group 3-7	2.301E-01	**2.333E-01**	2.017E-01	1.735E-01	2.184E-01	1.556E-01	2.201E-01
	Group 3-8	**2.948E-01**	2.941E-01	2.141E-01	1.984E-01	2.415E-01	1.446E-01	1.704E-01
	Group 3-9	**3.044E-01**	3.031E-01	2.181E-01	1.835E-01	2.451E-01	1.486E-01	1.709E-01
MagicK	**Group 3-0**	**2.423E-01**	2.396E-01	2.355E-01	1.886E-01	2.403E-01	1.880E-01	2.316E-01
	Group 3-1	**2.955E-01**	2.944E-01	2.789E-01	2.063E-01	2.804E-01	2.321E-01	2.445E-01
	Group 3-2	2.319E-01	2.265E-01	2.409E-01	2.112E-01	**2.424E-01**	2.054E-01	2.220E-01
	Group 3-3	**2.525E-01**	2.510E-01	2.484E-01	2.159E-01	2.516E-01	1.882E-01	1.931E-01
	Group 3-4	**2.342E-01**	2.309E-01	2.471E-01	2.111E-01	2.284E-01	2.094E-01	2.303E-01
	Group 3-5	**2.758E-01**	**2.758E-01**	2.581E-01	2.209E-01	2.686E-01	2.144E-01	2.586E-01
	Group 3-6	**2.832E-01**	2.828E-01	2.616E-01	2.156E-01	2.765E-01	2.235E-01	2.685E-01
	Group 3-7	2.618E-01	2.656E-01	2.534E-01	2.159E-01	**2.680E-01**	2.174E-01	2.574E-01
	Group 3-8	**2.798E-01**	2.772E-01	2.588E-01	2.175E-01	2.745E-01	2.220E-01	2.660E-01
	Group 3-9	**2.614E-01**	2.611E-01	2.546E-01	2.048E-01	2.604E-01	2.127E-01	2.528E-01

Table 17: Comparison between our algorithm and various baselines in terms of the fitness score of the recommended itinerary on groups with size 3 of the second dataset

Theme Park Attraction Visits dataset								
City	**Group**	**CA**	**GA**	**PersTour**	**AGAM**	**TLR**	**PersQ**	**GreedyFitness**
Disland	**Group 5-0**	**1.954E-01**	1.935E-01	1.469E-01	1.435E-01	1.415E-01	1.489E-01	1.610E-01
	Group 5-1	**1.916E-01**	1.869E-01	1.494E-01	1.485E-01	1.435E-01	1.548E-01	1.875E-01
	Group 5-2	**2.180E-01**	2.180E-01	1.623E-01	1.631E-01	1.541E-01	1.758E-01	2.180E-01
	Group 5-3	**2.117E-01**	2.117E-01	1.570E-01	1.525E-01	1.498E-01	1.715E-01	2.117E-01
	Group 5-4	**2.235E-01**	2.227E-01	1.605E-01	1.570E-01	1.523E-01	1.791E-01	2.227E-01
	Group 5-5	**2.375E-01**	**2.375E-01**	1.659E-01	1.654E-01	1.573E-01	1.892E-01	2.375E-01
	Group 5-6	2.041E-01	**2.049E-01**	1.600E-01	1.569E-01	1.479E-01	1.653E-01	2.056E-01
	Group 5-7	**2.110E-01**	2.105E-01	1.581E-01	1.560E-01	1.504E-01	1.710E-01	2.110E-01
	Group 5-8	**2.904E-01**	2.834E-01	1.871E-01	1.865E-01	1.740E-01	2.209E-01	2.834E-01
	Group 5-9	**2.931E-01**	2.925E-01	1.896E-01	1.985E-01	1.754E-01	2.275E-01	2.931E-01
CaliAdv	**Group 5-0**	**2.044E-01**	2.029E-01	1.840E-01	1.588E-01	2.071E-01	1.460E-01	1.920E-01
	Group 5-1	2.092E-01	**2.120E-01**	1.917E-01	1.775E-01	2.117E-01	1.487E-01	1.931E-01
	Group 5-2	**2.365E-01**	2.348E-01	1.887E-01	1.826E-01	2.201E-01	1.727E-01	1.787E-01
	Group 5-3	**2.368E-01**	2.293E-01	2.010E-01	1.841E-01	2.186E-01	1.625E-01	1.920E-01
	Group 5-4	**2.414E-01**	**2.414E-01**	1.901E-01	1.831E-01	2.237E-01	1.468E-01	1.756E-01
	Group 5-5	**2.652E-01**	2.550E-01	2.118E-01	1.752E-01	2.282E-01	1.482E-01	2.058E-01
	Group 5-6	2.354E-01	**2.360E-01**	2.067E-01	2.020E-01	2.163E-01	1.553E-01	1.922E-01
	Group 5-7	2.301E-01	**2.333E-01**	2.017E-01	1.735E-01	2.184E-01	1.556E-01	2.201E-01
	Group 5-8	**2.948E-01**	**2.941E-01**	2.141E-01	1.984E-01	2.415E-01	1.446E-01	1.704E-01
	Group 5-9	**3.044E-01**	3.031E-01	2.181E-01	1.835E-01	2.451E-01	1.486E-01	1.709E-01
MagicK	**Group 5-0**	2.16E-01	2.18E-01	2.35E-01	1.99E-01	**2.28E-01**	1.87E-01	2.17E-01
	Group 5-1	**2.22E-01**	2.13E-01	2.30E-01	1.94E-01	2.20E-01	1.93E-01	2.07E-01
	Group 5-2	**2.32E-01**	**2.32E-01**	2.43E-01	2.14E-01	2.25E-01	2.04E-01	2.28E-01
	Group 5-3	**2.29E-01**	2.28E-01	2.38E-01	2.06E-01	2.21E-01	2.04E-01	2.24E-01
	Group 5-4	**2.42E-01**	2.36E-01	2.40E-01	2.13E-01	2.41E-01	2.06E-01	2.31E-01
	Group 5-5	**2.51E-01**	**2.51E-01**	2.49E-01	2.16E-01	**2.51E-01**	2.17E-01	2.41E-01
	Group 5-6	**2.24E-01**	2.21E-01	2.35E-01	2.05E-01	2.18E-01	2.02E-01	2.17E-01
	Group 5-7	2.26E-01	2.27E-01	2.39E-01	2.11E-01	**2.42E-01**	2.04E-01	2.23E-01
	Group 5-8	**2.97E-01**	2.92E-01	2.69E-01	2.05E-01	2.79E-01	2.28E-01	2.73E-01
	Group 5-9	**3.07E-01**	3.04E-01	2.73E-01	2.23E-01	2.84E-01	2.32E-01	2.80E-01

Table 18: Comparison between our algorithm and various baselines in terms of the fitness score of the recommended itinerary on groups with size 5 of the second dataset

Theme Park Attraction Visits dataset								
City	**Group**	**CA**	**GA**	**PersTour**	**AGAM**	**TLR**	**PersQ**	**GreedyFitness**
Disland	**Group 10-0**	**1.977E-01**	1.970E-01	1.520E-01	1.535E-01	1.461E-01	1.604E-01	1.977E-01
	Group 10-1	**2.049E-01**	2.030E-01	1.567E-01	1.549E-01	1.495E-01	1.638E-01	2.029E-01
	Group 10-2	1.846E-01	**1.850E-01**	1.466E-01	1.635E-01	1.415E-01	1.515E-01	1.846E-01
	Group 10-3	**2.528E-01**	**2.528E-01**	1.770E-01	1.806E-01	1.663E-01	1.975E-01	2.528E-01
	Group 10-4	**2.384E-01**	**2.384E-01**	1.698E-01	1.766E-01	1.610E-01	1.878E-01	2.384E-01
	Group 10-5	**2.879E-01**	2.874E-01	1.924E-01	1.888E-01	1.795E-01	2.211E-01	2.879E-01
	Group 10-6	1.924E-01	**1.931E-01**	1.501E-01	1.464E-01	1.442E-01	1.568E-01	1.924E-01
	Group 10-7	**2.626E-01**	2.588E-01	1.819E-01	1.515E-01	1.699E-01	2.040E-01	2.626E-01
	Group 10-8	**2.099E-01**	2.093E-01	1.571E-01	1.541E-01	1.505E-01	1.685E-01	2.099E-01
	Group 10-9	**2.155E-01**	2.150E-01	1.604E-01	1.625E-01	1.527E-01	1.723E-01	2.155E-01
CaliAdv	**Group 10-0**	**2.227E-01**	2.209E-01	1.987E-01	1.942E-01	2.160E-01	1.633E-01	2.112E-01
	Group 10-1	2.189E-01	**2.247E-01**	1.920E-01	1.904E-01	2.177E-01	1.500E-01	1.795E-01
	Group 10-2	**2.046E-01**	2.026E-01	1.892E-01	1.891E-01	2.111E-01	1.448E-01	1.758E-01
	Group 10-3	**2.692E-01**	2.689E-01	2.019E-01	2.008E-01	2.366E-01	1.526E-01	1.697E-01
	Group 10-4	2.571E-01	**2.633E-01**	1.961E-01	1.954E-01	2.306E-01	1.592E-01	1.797E-01
	Group 10-5	**3.057E-01**	2.955E-01	2.159E-01	2.231E-01	2.486E-01	1.493E-01	1.693E-01
	Group 10-6	2.165E-01	**2.203E-01**	1.857E-01	1.846E-01	2.144E-01	1.596E-01	1.715E-01
	Group 10-7	**2.830E-01**	2.707E-01	2.057E-01	2.104E-01	2.401E-01	1.474E-01	1.694E-01
	Group 10-8	**2.309E-01**	2.279E-01	1.899E-01	1.901E-01	2.204E-01	1.469E-01	1.769E-01
	Group 10-9	2.296E-01	**2.357E-01**	1.889E-01	1.872E-01	2.232E-01	1.539E-01	1.755E-01
MagicK	**Group 10-0**	**2.223E-01**	2.130E-01	2.117E-01	2.011E-01	2.216E-01	1.989E-01	2.124E-01
	Group 10-1	2.259E-01	2.239E-01	**2.364E-01**	2.006E-01	2.162E-01	2.003E-01	2.159E-01
	Group 10-2	2.123E-01	2.125E-01	2.262E-01	1.948E-01	**2.269E-01**	1.918E-01	2.113E-01
	Group 10-3	**2.714E-01**	2.611E-01	2.569E-01	2.217E-01	2.614E-01	2.219E-01	2.495E-01
	Group 10-4	**2.603E-01**	2.506E-01	2.497E-01	2.115E-01	2.541E-01	2.163E-01	2.398E-01
	Group 10-5	**3.040E-01**	3.016E-01	2.726E-01	2.283E-01	2.791E-01	2.376E-01	2.731E-01
	Group 10-6	2.199E-01	2.112E-01	2.298E-01	2.053E-01	**2.309E-01**	1.949E-01	2.089E-01
	Group 10-7	**2.780E-01**	2.763E-01	2.619E-01	2.060E-01	2.663E-01	2.263E-01	2.561E-01
	Group 10-8	**2.392E-01**	2.287E-01	2.368E-01	2.117E-01	2.317E-01	2.032E-01	2.206E-01
	Group 10-9	2.392E-01	2.392E-01	**2.402E-01**	2.047E-01	2.325E-01	2.057E-01	2.244E-01

Table 19: Comparison between our algorithm and various baselines in terms of the fitness score of the recommended itinerary on groups with size 10 of the second dataset

The comparison results on the second dataset in tables 17, 18, and 19 disclosed that our algorithm outperforms in 73% of the experiments in groups with size 3, 80% in groups with size 5, and 66% in groups with size 10. Whereas the statistics of each theme park are different from others, the proposed algorithm had a consistent and stable proficiency in all three regions and all the generated groups. Analyzing the results of other algorithms revealed that GA had the best performance after our proposed algorithm with at least 26.6% of the cases for all group sizes, which shows that the evolutionary approaches are more productive than other methodologies in solving this kind of problem.

4.7.3 Non-parametric statistical analysis

There are two types of statistical processes used to perform the statistical analysis: parametric and non-parametric [53]. The results of the parametric tests are reported in the previous section. Now according to the principles of [54], we use some non-parametric tests to demonstrate that the proposed algorithm has consistent results and can also deliver better results than the other algorithms. Compared to the parametric test, which makes assumptions about a population's parameters, non-parametric tests do not assume anything about the underlying distribution of the variables being assessed.

The first non-parametric analysis is conducted to check the consistency of the results of our algorithm. For this purpose, two non-parametric statistical tests for multiple comparisons were used in particular. One of the most frequently used tests is Friedman test [55, 56]. The null hypothesis for this test is that the populations' medians are equal. The alternative hypothesis is non-directional since it is defined as the negation of the null hypothesis [53]. Another utilized non-parametric test is Kruskal test. The null hypothesis of this pre-hoc test is that the variables come from the same distribution. This test uses a rank system as well, but it can handle more than two groups at once [57].

All the experiments on the proposed algorithm are repeated five times in each group, and the average fitness score of the solution is reported in the previous section. Using the non-parametric tests, we check the stability of the results in all these five experiments for different groups. The null hypothesis is that there is a same distribution between each CA experiment with a significance level of $\alpha = 0.05$. As reported in Table 20, the obtained p-value from both Friedman and Kruskal tests are greater than α. Therefore, it failed to reject the null hypothesis, which is interpreted strongly suggesting that the results are drawn from the same distributions.

CA	Friedman Test		Kruskal Test	
	Statistic	p-value	Statistic	p-value
	5.200	0.267	2.525	0.640

Table 20: Non-parametric statistical analysis on CA results ($\alpha = 0.05$)

Furthermore, another test is run on the results obtained from all examined algorithms to check if there is a considerable difference between their output. Wilcoxon test is employed for this analysis which is a non-parametric approach that uses a two-sample design in hypothesis testing situations. It seeks to find significant differences between two samples means that the behaviours of two algorithms are not similar [53]. The null hypothesis in this test is that CA and other solution approaches produce similar results with no statistically significant differences, considering a level of significance $\alpha = 0.05$.

Table 21 shows the corresponding results, which include the test statistic value and the p-value derived from each test that compared CA with each of the baselines. In this table, the p-value for all algorithms are less that α. Therefore, the test results indicate that we can correctly reject the null hypothesis and it implies a difference in accuracy between the CA and all the compared approaches.

		GA	**PersTour**	**AGAM**	**TLR**	**PersQ**	**Greedy**
WilcoxonTest	**Statistic**	3.000	0.000	0.000	7.000	0.000	0.000
	p-value	0.010	0.002	0.002	0.037	0.002	0.002

Table 21: Non-parametric statistical analysis comparing CA and baselines ($\alpha = 0.05$)

4.7.4 Run-time analysis

We have previously demonstrated the efficacy of our method. This section compares the run time of our approach with the baselines. Figure 8 presents the run-time comparison of all the algorithms on groups of different sizes.

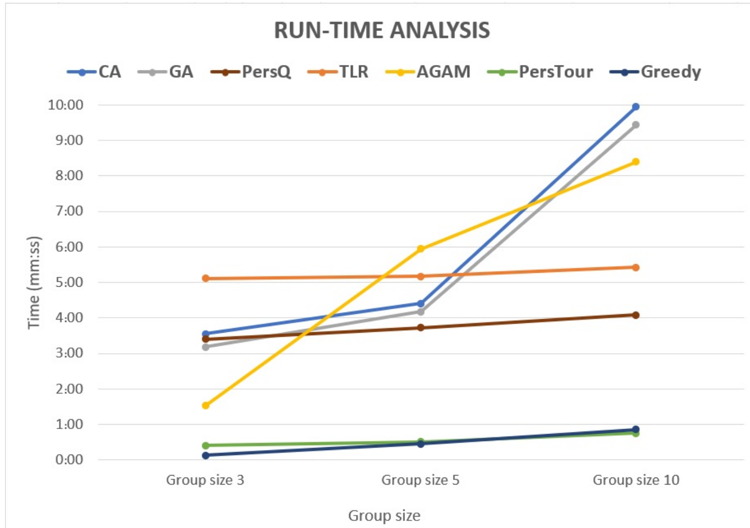

Figure 8: Run-time analysis

We witnessed that the time-complexity of the algorithm follows a linear manner while increasing the group size. Moreover, its run-time is acceptable compared with other baselines such as TLR and AGAM, especially when group size is 3 and 5, whereas it exceeds them when applied to groups with size 10.

A comparison of the algorithms in terms of the evolution process is depicted in Figure 9. As we are witnessing in these charts, the quality of the solutions generated by CA is evolving to the final iterations, while GA has a consistent quality after one-third of the iterations. This shows that the evolution in CA could keep on if the termination criteria have not stopped the process. Therefore, 50 iterations lead to a better result than 30, letting the algorithm continue the evolution process, and the final result is much more satisfying than GA.

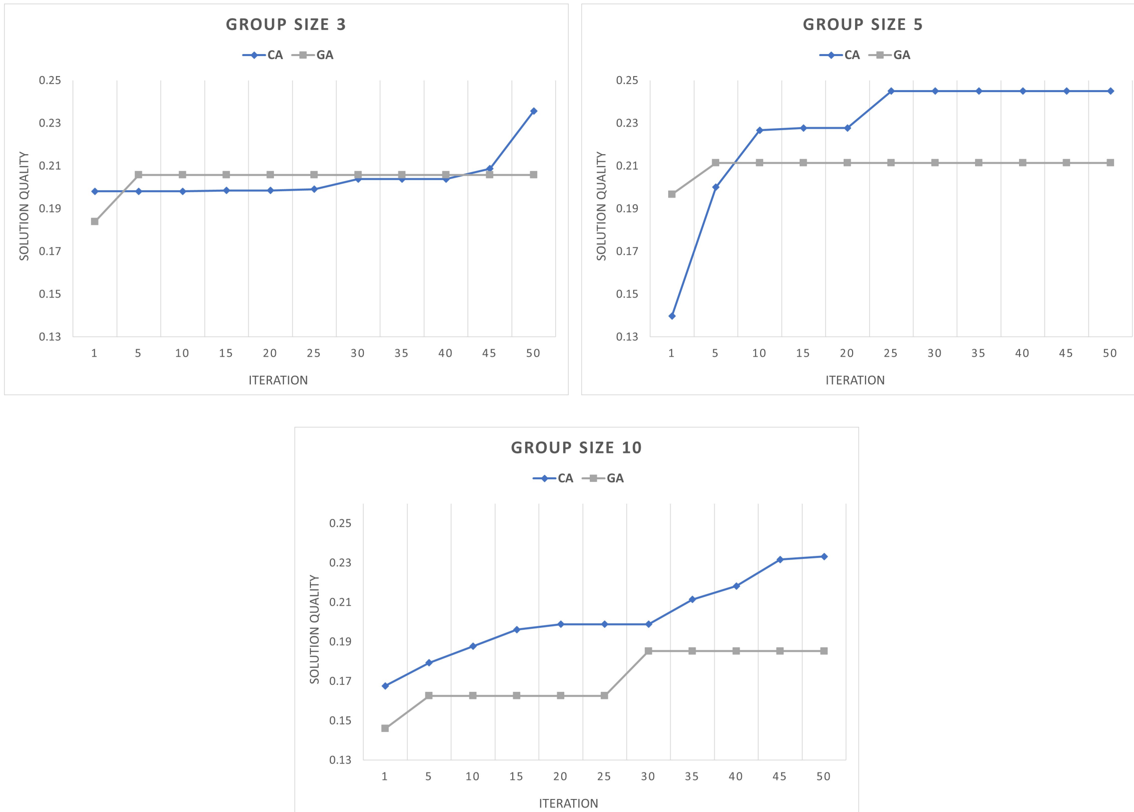

Figure 9: Comparison of the evolution of our proposed algorithm with Genetic algorithm, considering the solution of both algorithms for Delhi on population size 200 on different group sizes

4.8 Discussion

The importance of tour planning research has increased the widespread use of personalized itinerary recommendation systems. Therefore, in this research study, we proposed a personalized group itinerary recommendation algorithm using Cultural Algorithm. The algorithm recommends an itinerary corresponding to the interests of all tourists in a group and their limitations, such as their time, budget, and the maximum number of points they tend to visit. To select the best itinerary which matches users' needs, we focus on users' interests into different categories of POIs, POIs popularity, visit duration, entrance cost, the distance between POIs, and the number of POIs covered in the itinerary.

In this algorithm, the main objective is to enhance the accuracy of the recommended solution by using the knowledge gained during the evolution process. This is why it can reduce the search space and head to the final result faster than other baselines while improving the effectiveness of the solution.

The proposed algorithm has been applied to two real-world datasets to test the performance and efficiency. Some experiments have been conducted to select the best population size for the CA algorithm. The results revealed that the bigger the population, the better the fitness score. As a result, between three population sizes of 100, 150, and 200, population size 200 was selected for the remaining experiments.

Moreover, CA was compared to some state-of-the-art algorithms related to this topic. Six other algorithms were selected from different categories of methodology, such as evolutionary, deep learning, MCTS, and integer programming approaches, to make a comprehensive comparison. The results of the mentioned experiments depicted that our proposed algorithm is more efficient than the baseline algorithms in most of the experiments. Moreover, the non-parametric tests indicated that there is consistency in the results of our algorithm in a different situation, and there is also a notable difference between CA and other baselines.

Time analysis revealed that the proposed algorithm has better or similar run time compared with others, mainly when applied to groups with sizes 3 and 5. However, it takes longer than others to recommend an itinerary for bigger groups. Whereas the improvement gained in the quality of the final solution for all group sizes compared with baselines outweigh the time difference in larger groups. In addition, studying the evolution process of CA and other evolutionary algorithms, such as GA, expressed that CA pursues its progress to accomplish a more efficient result while GA stops the evolution process in the early iterations.

CHAPTER 5

Conclusion and Future Work

The objective of itinerary recommendation is to offer a plan for a tourist or a group of tourists to visit some travel spots, depending on their interests and taking into account specific constraints such as time, budget, and the number of points of interest (POIs) to visit. Considering the visit history of each user, their interests in each category of POIs were extracted and utilized for the itinerary recommendation.

In this thesis, we have proposed a personalized group itinerary recommendation algorithm using the Cultural Algorithm to generate an itinerary that satisfies all group members to some extent while does not exceed their limitations. The proposed algorithm can solve the group itinerary recommendation problem as a multi-objective optimization problem with some characteristics, such as Variable size of the solution, and important order and sequence of the selected items in the solution.

Cultural Algorithms can be considered as an extension of Genetic Algorithms, in which the belief space serves as a knowledge conduit between successive generations that are being evolved. Thus, the knowledge extracted from the population in each iteration is stored in two matrices designed for belief space. First, a random population is generated. Then, itineraries with high quality are selected to form the belief space. One of the matrices collects the POIs that have occurred more than others, and the other one keeps the sequence of the POIs in those solutions. Consequently, the following populations are generated from the belief space or by using crossover and mutation methodologies.

The proposed algorithm has been evaluated on two real-world datasets: the Flickr User-POI Visits dataset and Theme Park Attraction Visits Dataset. We have conducted multiple experiments to analyze the best settings for the proposed algorithms and also evaluated its performance on some groups of different sizes of 3, 5, and 10. For further assessment, we compared the performance of our algorithms with six state-of-the-art algorithms which used various methodologies. As for the evaluation metrics, the quality of the recommended itineraries has been assessed by measuring their fitness score. Moreover, a run-time comparison has been made on all algorithms.

The results revealed that our proposed algorithm beat alternative baselines on the Flickr User-POI Visits dataset in 72.5%, 67.5%, and 65% of situations for groups of sizes 3, 5, and 10, respectively. Additionally, CA outperformed the existing algorithms in 73% of the experiments on groups of size 3, 80% on groups of size 5, and 66% on groups of size 10 when applying to the Theme Park Attraction Visits dataset. Moreover, non-parametric tests revealed that this algorithm produces consistent results in various circumstances and differs significantly from previous algorithms. In addition, the run-time analysis revealed that our algorithm has acceptable and better time complexity compared with other algorithms especially in groups with size 3 and 5, while it could be enhanced when working with larger groups.

Overall, the proposed algorithm could beat other baseline algorithms in terms of the quality of the recommended itinerary and provide more satisfactory solutions for tourists. It plans the itineraries considering users' interests and takes into account their limitations.

5.1 Future works

In our future work, we intend to address other aspects of the itinerary recommendation problem. We count on user interests extracted from their travel history in this work. Thus, addressing the cold-start problem for recommending an itinerary to a user with no travel history is one of the aspects of extending our research.

In addition, as revealed in run-rime analysis, the time complexity of our proposed algorithm is higher than other baselines while recommending an itinerary for groups with a size 10. Therefore, it can be improved to complete in a sufficient time, especially for groups of considerable size. Moreover, it could be enhanced with more real-life constraints, such as POIs queuing and opening times.

Finally, testing the performance of the proposed algorithm in solving similar real-life problems, such as educational planning and scheduling, and with other input parameters is a part of our future works.

REFERENCES

[1] Joy Lal Sarkar, Abhishek Majumder, Chhabi Rani Panigrahi, and Sudipta Roy. Multitour: A multiple itinerary tourists recommendation engine. *Electronic Commerce Research and Applications*, 40:100943, 2020.

[2] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. Tour recommendation and trip planning using location-based social media: A survey. *Knowledge and Information Systems*, 60(3):1247–1275, 2019.

[3] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems*, 54(2):375–406, 2018.

[4] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. Personalized tour recommendation based on user interests and points of interest visit durations. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[5] Wouter Souffriau, Pieter Vansteenwegen, Joris Vertommen, Greet Vanden Berghe, and Dirk Van Oudheusden. A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*, 22(10):964–985, 2008.

[6] Wouter Souffriau, Pieter Vansteenwegen, Greet Vanden Berghe, and Dirk Van Oudheusden. The multiconstraint team orienteering problem with multiple time windows. *Transportation Science*, 47(1):53–63, 2013.

[7] Serhan Kotiloglu, Theodoros Lappas, Konstantinos Pelechrinis, and PP Repoussis. Personalized multi-period tour recommendations. *Tourism Management*, 62:76–88, 2017.

[8] Budhi S Wibowo and Monica Handayani. A genetic algorithm for generating travel itinerary recommendation with restaurant selection. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 427–431. IEEE, 2018.

[9] Aris Anagnostopoulos, Reem Atassi, Luca Becchetti, Adriano Fazzone, and Fabrizio Silvestri. Tour recommendation for groups. *Data Mining and Knowledge Discovery*, 31(5):1157–1188, 2017.

[10] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*. John Wiley & Sons, 2004.

[11] Heiko Rieger. Optimization problems and algorithms from computer science. *Optimization*, 2:0, 2008.

[12] Jing J Liang, BY Qu, Ponnuthurai Nagaratnam Suganthan, and Alfredo G Hernández-Díaz. Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, 201212(34):281–295, 2013.

[13] Xinjie Yu and Mitsuo Gen. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[14] Seyedali Mirjalili. Genetic algorithm. In *Evolutionary algorithms and neural networks*, pages 43–55. Springer, 2019.

[15] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[16] Mehdi Kargar and Zhibin Lin. A socially motivating and environmentally friendly tour recommendation framework for tourist groups. *Expert Systems with Applications*, 180:115083, 2021.

[17] World tourism day – tourism and jobs: a better future for all – richmond economic development. https://businessinrichmond.ca/sector-profile-tourism-and-jobs-a-better-future-for-all/_ftn1, 2019.

[18] Zhu Zhongming, Lu Linong, Yao Xiaona, Zhang Wangqiang, Liu Wei, et al. Oecd tourism trends and policies 2020. 2020.

[19] Phatpicha Yochum, Liang Chang, Tianlong Gu, Manli Zhu, and Hongliang Chen. A genetic algorithm for travel itinerary recommendation with mandatory points-of-interest. In *International Conference on Intelligent Information Processing*, pages 133–145. Springer, 2020.

[20] WANG Yongzhen, CHEN Yan, and Yu Yingying. Improved grouping genetic algorithm for solving multiple traveling salesman problem. , 39(1):198–205, 2017.

[21] Gorka Kobeaga, María Merino, and Jose A Lozano. An efficient evolutionary algorithm for the orienteering problem. *Computers & Operations Research*, 90:42–59, 2018.

[22] Chen Yuan and Minoru Uehara. Improvement of multi-purpose travel route recommendation system based on genetic algorithm. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, pages 305–308. IEEE, 2019.

[23] Robert A Walker and Samit Chaudhuri. Introduction to the scheduling problem. *IEEE Design & Test of Computers*, 12(2):60–69, 1995.

[24] Kalyani Selvarajah, Pooya Moradian Zadeh, Mehdi Kargar, and Ziad Kobti. Identifying a team of experts in social networks using a cultural algorithm. *Procedia Computer Science*, 151:477–484, 2019.

[25] Budhi S Wibowo and Monica Handayani. A genetic algorithm for generating travel itinerary recommendation with restaurant selection. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 427–431. IEEE, 2018.

[26] Kwan Hui Lim, Xiaoting Wang, Jeffrey Chan, Shanika Karunasekera, Christopher Leckie, Yehui Chen, Cheong Loong Tan, Fu Quan Gao, and Teh Ken Wee. Perstour: A personalized tour recommendation and planning system. In *HT (Extended Proceedings)*, 2016.

[27] Phatpicha Yochum, Liang Chang, Tianlong Gu, Manli Zhu, and Hongliang Chen. A genetic algorithm for travel itinerary recommendation with mandatory points-of-interest. In *International Conference on Intelligent Information Processing*, pages 133–145. Springer, 2020.

[28] Phatpicha Yochum, Liang Chang, Tianlong Gu, and Manli Zhu. An adaptive genetic algorithm for personalized itinerary planning. *IEEE Access*, 8:88147–88157, 2020.

[29] Joy Lal Sarkar and Abhishek Majumder. A new point-of-interest approach based on multi-itinerary recommendation engine. *Expert Systems with Applications*, 181:115026, 2021.

[30] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[31] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73, jan 2016.

[32] Manousos Rigakis, Dimitra Trachanatzi, Magdalene Marinaki, and Yannis Marinakis. Tourist group itinerary design: When the firefly algorithm meets the n-person battle of sexes. *Knowledge-Based Systems*, 228:107257, 2021.

[33] Elif Erbil and Wolfgang Wörndl. Personalization of multi-day round trip itineraries according to travelers' preferences. In *ENTER22 e-Tourism Conference*, pages 187–199. Springer, 2022.

[34] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. Transformer-based multi-task learning for queuing time aware next poi recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 510–523. Springer, 2021.

[35] Lei Chen, Lu Zhang, Shanshan Cao, Zhiang Wu, and Jie Cao. Personalized itinerary recommendation: Deep and collaborative learning with textual information. *Expert Systems with Applications*, 144:113070, 2020.

[36] Zahra Abbasi-Moud, Hamed Vahdat-Nejad, and Wathiq Mansoor. Detecting tourist's preferences by sentiment analysis in smart cities. In *2019 IEEE Global Conference on Internet of Things (GCIoT)*, pages 1–4. IEEE, 2019.

[37] Zahra Abbasi-Moud, Hamed Vahdat-Nejad, and Javad Sadri. Tourism recommendation system based on semantic clustering and sentiment analysis. *Expert Systems with Applications*, 167:114324, 2021.

[38] Jiqing Gu, Chao Song, Wenjun Jiang, Xiaomin Wang, and Ming Liu. Enhancing personalized trip recommendation with attractive routes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 662–669, 2020.

[39] Chenyi Zhang, Hongwei Liang, Ke Wang, and Jianling Sun. Personalized trip recommendation with poi availability and uncertain traveling time. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 911–920, 2015.

[40] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. Efficient itinerary recommendation via personalized poi selection and pruning. *Knowledge and Information Systems*, 64(4):963–993, 2022.

[41] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems*, 54(2):375–406, 2018.

[42] Prarthana Padia, Bhavya Singhal, and Kwan Hui Lim. User-relative personalized tour recommendation. In *IUI Workshops*, 2019.

[43] Mehdi Kargar and Zhibin Lin. A socially motivating and environmentally friendly tour recommendation framework for tourist groups. *Expert Systems with Applications*, 180:115083, 2021.

[44] Joy Lal Sarkar and Abhishek Majumder. gtour: Multiple itinerary recommendation engine for group of tourists. *Expert Systems with Applications*, 191:116190, 2022.

[45] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. Personalized itinerary recommendation with queuing time awareness. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 325–334, 2017.

[46] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[47] Lei Chen, Lu Zhang, Shanshan Cao, Zhiang Wu, and Jie Cao. Personalized itinerary recommendation: Deep and collaborative learning with textual information. *Expert Systems with Applications*, 144:113070, 2020.

[48] Nuno Luz, Ricardo Anacleto, Constantino Martins, Ana Almeida, and Joel Pinho Lucas. Lightweight tourism recommendation. *International Journal of Web Engineering and Technology 9*, 8(2):106–123, 2013.

[49] Robert G Reynolds. An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*, volume 24, pages 131–139. World Scientific, 1994.

[50] Alireza Maheri, Shahin Jalili, Yousef Hosseinzadeh, Reza Khani, and Mirreza Miryahyavi. A comprehensive survey on cultural algorithms. *Swarm and Evolutionary Computation*, 62:100846, 2021.

[51] Carlos A Coello Coello and Ricardo Landa Becerra. Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization*, 36(2):219–236, 2004.

[52] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. Towards next generation touring: Personalized group tours. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 26, pages 412–420, 2016.

[53] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[54] Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information sciences*, 180(10):2044–2064, 2010.

[55] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[56] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[57] William H Kruskal and W Allen Wallis. Errata: Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 48(264):907–911, 1953.

VITA AUCTORIS

NAME:	Farzaneh Jouyandeh
PLACE OF BIRTH:	Iran
YEAR OF BIRTH:	1991
EDUCATION:	
	University of Tehran, Tehran, Tehran, Iran Bachelor's of Science, Computer Science, 2009 - 2013
	University of Windsor, Windsor, ON, Canada Master's of Science, Computer Science, 2021 - 2022