processes

MDPI

Article

Design and Implementation of an Optimal Travel Route Recommender System on Big Data for Tourists in Jeju

Lei Hang, Sang-Hun Kang, Wenquan Jin and Do-Hyeun Kim *

Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; hanglei@jejunu.ac.kr (L.H.); myks790@gmail.com (S.-H.K.); wenquan.jin@jejunu.ac.kr (W.J.)

* Correspondence: kimdh@jejunu.ac.kr; Tel.: +82-64-7543658

Received: 24 July 2018; Accepted: 8 August 2018; Published: 17 August 2018

Abstract: A recommender system is currently applied in many different domains, seeking to provide users with recommendation services according to their personalized preferences to relieve rising online information congestion. As the number of mobile phone users is large and growing, mobile tourist guides have attracted considerable research interest in recent years. In this paper, we propose an optimal travel route recommender system by analyzing the data history of previous users. The open dataset used covers the travel data from thousands of mobile tourists who visited Jeju in a full year. Our approach is not only personalized to users' preferences but also able to recommend a travel route rather than individual POIs (Points of Interest). An association rule mining-based approach, which takes into account contextual information (date, season and places already visited by previous users), is used to produce travel routes from the large dataset. Furthermore, to ensure the reasonability of the recommendation, a genetic algorithm optimization approach is proposed to find the optimal route among them. Finally, a mobile tourist case study is implemented in order to verify the feasibility and applicability of the proposed system. This application embeds a graphic map for plotting the travel route and provides detailed information of each travel spot as well. The results of this work indicate that the proposed system has great potential for travel planning preparation for mobile users.

Keywords: travel route recommendation; association rule mining; route optimization; mobile; travel planning

1. Introduction

Great opportunities have been offered with the flourishment of social media, such as Facebook and Twitter, for addressing many challenging problems, such as travel recommendations. Travel recommendation is an important application field in both research and industry. Many travel sites provide information on the proposal, which is written by other users to help the user to create a unique travel plan. For example, TripSpot [1] is a famous travel site featuring more than 100 country guides through which users can get a variety of information, including tourism information, destination ideas, maps, weather forecasts, etc. Nowadays, more and more people join the social media network and share their photos which records users' daily life and travel experience. These photos contain heterogeneous metadata (e.g., tags, date taken, latitude etc.) which are not only useful for reliable POIs (Points of Interest) [2] but also provide a great opportunity to recommend personalized travel routes according to users' preferences.

Two main challenges have to be noted when considering travel recommendations. First, travel recommender systems should provide personalized travel recommendations based on user interest since the preference of POIs varies from user to user [3]. For example, some people may prefer historical sites such as temples and pavilions, while others are interested in city landmarks such as high-rise buildings. Many other attributes including season and visiting times are also helpful to provide personalized travel recommendations, more so than topical travel interest. Secondly, the recommendation of a sequential travel route is far more complex and time-consuming than individual POIs, since the distance between the locations or opening times of different POIs should be considered [4].

Existing approaches on travel recommendation can be classified into two kinds in terms of the services they render, which are destination-based recommendations and route-based recommendations. The former approach is centered on the recommendation of a single destination which best meets the user's interest. One of the typical cases is PersonalTour [5], which is used by travel agencies to help their customers find the best travel packages according to their preferences. Once the recommendation process is completed, a list of rated options is presented to the user. The latter approach not only provides a list of the places that fit better with the user's preferences, but also help tourists create a route containing several destinations. Trip-planning [6] is a time-consuming task and there is always a demand for a system that is able to provide tourist locations according to users' interests and preferences. Photo2Trip [7] leverages geo-tagged photos to suggest customized travel routes according to the user's preference, and it also enables the user to input personal preference in an interactive manner.

However, these existing recommendation approaches only work based on user topic interest mining but without considering other personalized attributes such as preferred seasons. Some of them facilitate route mining using personalized travel attributes, but without validating the reasonability of the route. It cannot be a reasonable recommendation, for example, if the recommended POIs of a route in one day are located in different directions. As a result, high reliable travel route recommendation is still a formidable challenge for the existing travel recommendation approaches.

The contribution of this work can be divided into a three-fold structure: first, we propose a novel route recommendation system that aims to provide travel routes depending on previous users' historical experience. An association rule mining-based approach is utilized, which is used to discover the frequent routes between different tourists in a large dataset. Secondly, a genetic algorithm-based route optimization approach is designed to find the optimal travel route by using the distance fitness function. Lastly, a mobile tourist case study is implemented as the proof of concept to validate the performance of the proposed system. Numerous activities and snapshots of the case study are presented and discussed.

The remainder of this paper is organized as follows: Section 2 discusses some main approaches used in travel recommender systems and overviews some of the similar related research works. Section 3 gives an overview of the proposed system's conceptual architecture and details each component of the system. Section 4 overviews the experimental setup and reports the results of the proposed travel route recommender system. Section 5 provides some insight into the implementation of the case study. Section 6 overviews the implementation of the mobile tourist case study with various snapshots. Section 7 outlines the significance of the proposed work by a comparative analysis of the proposed work with some existing works. Finally, Section 8 concludes the paper and discusses the future direction of research.

2. Related Work

This section overviews some of the recent studies in travel recommendation and points out the differences between the designed work and existing works. General travel recommender systems recommend travel routes in terms of the popularity of POIs or routes [8]. Recently, personalized travel recommendations have attracted more attention. Collaborative Filtering (CF) [9], Markov Chains [10] and matrix factorization [11] are three main approaches, widely used in personalized travel recommendation. The most famous method is location-based collaborative filtering (LCF) which measures similar social users based on the location co-occurrence of previously visited POIs and then ranks POIs based on similar users' visiting records. Clements et al. [12] model the co-occurrence with Gaussian density estimation and travel routes are recommended according to similar users' voting. Cheng et al. [13] built a matrix of user location by using a user check-in information on locations and a multi-center Gaussian model was adopted to model the probability to analyze the user's check-in behavior, including check-in frequency over the spatial, which are used to predict unknown frequency to these unvisited locations. Another project called GoThere [14] is a context and preference aware travel guide that suggests significant tourist destinations based on users' preferences and the current surrounding context generated from the well-known social media repository Flicker [15]. The user preferences on the content of travel spots are distinguished from the user preferences on travel spots themselves, which motivates an independent location model of users' content preferences and a location-aware model of users' location preferences of travel spots. However, the location-based approaches may face the data sparse issue when the users have very few location records so that the mining of similar users could be very difficult. A topic model is effective for solving the data sparse problem, for example, an author topic model-based collaborative filtering method [16] is proposed to mine the category of user topic interest (i.e., cultural, cityscape, landmark, and more) simultaneously so as to facilitate comprehensive recommendations for social users. A location-based and preference-aware travel recommendation system is presented by the authors of [17] by using a weighted category hierarchy to model each individual's personal preferences, from learning an iterative learning model in their offline module.

However, personalized travel route recommendation [18] is more convenient for users as the conventional approaches only provide an individual recommendation. For example, the authors of [19] propose a new trip-planning method to recommend a travel route that is related to users in the given context. Travel preferences from users' travel history in one location are used to recommend tourist locations in other cities. Another project [20] focuses on sentimental attributes of location and proposes a mining-based method. They use a sentiment-based mining algorithm to mine the user interested locations with obvious sentimental attributes so as to recommend the travel routes to other users. Gao et al. present a travel guidance system [21], which can automatically recognize and rank the landmarks for tourists. They applied a novel landmark ranking approach by using the geotag info in Flicker and user knowledge from Yahoo Travel Guide. The proposed method selected the popular landmarks and computed the probability of a tag being a landmark name.

In recent years, studies of the travel route recommendation that contain more context attributes have shown more effective performance than topical interest. For example, Yuan et al. exploited both geographical and temporal influence in the time-aware recommendation. They proposed a geographical-temporal influence-aware graph which is aimed at modeling check-in records, geographical influence, and temporal influence to recommend POIs to a user at a given time [22]. Chen et al. focus on leveraging the tourists' attributes from the photo contents [23]. They mined the demographics for different locations and travel paths and then a Bayesian learning framework was introduced for further giving a mobile recommendation on the spot.

To the best of our knowledge, these existing studies related to travel recommendation either recommend destination considering only user preferences or put effort on route mining considering personal travel attributes. Furthermore, these travel attributes have not been mined automatically and few of them optimize the recommendation to ensure reliability and availability. It is necessary to fully adopt all of these features in designing a travel route recommendation system and thus improve the performance.

3. Proposed Optimal Travel Route Recommender System Architecture

Figure 1 illustrates the conceptual architecture of the proposed optimal travel route recommender system. In this work, the recommender system is aimed at helping users to make personalized travel decisions based on the data history from previous users. To achieve this purpose, the proposed system automatically analyzes the data resources to mine travel routes according to personal travel interests (i.e., preferred season, weather, etc.). Then the system searches the best travel route and provides the recommendation to mobile users.

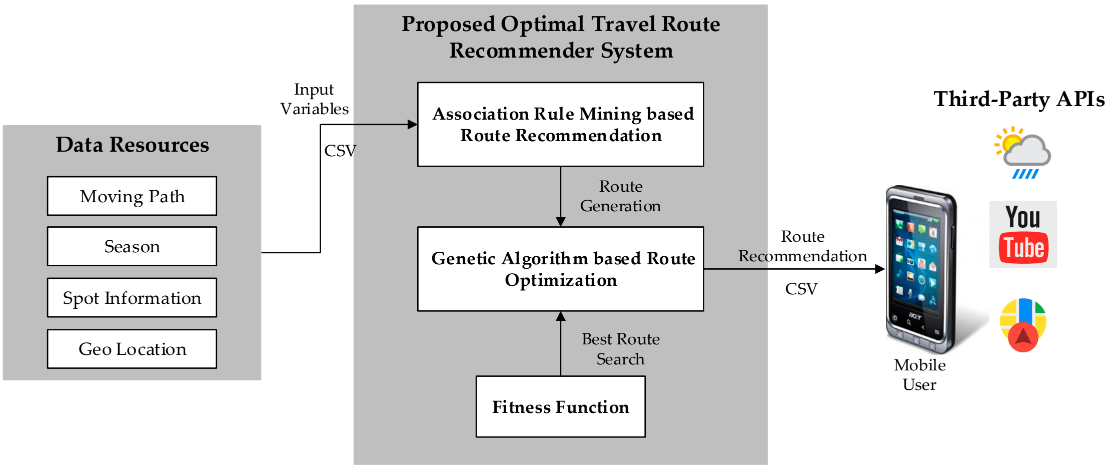


Figure 1. Conceptual architecture of the optimal travel route recommender system.

More specially, the data resources are in Comma-Separated Values (CSV) format, containing multiple values such as moving path, season, geolocation and the information of each travel spot. These values are taken as inputs of the route recommender system on which the association rule mining-based route recommendation approach is applied to find travel routes. The genetic algorithm-based route optimization solution takes these routes to find the optimal routes by referring to the fitness function. These optimal routes are also maintained as a CSV document, which is ingested by the mobile application and thus makes them visible in the intuitive graphical map. Furthermore, various third-party Application Program Interfaces (APIs) are utilized to provide different functionalities such as weather forecast and video searching. The detailed processes of the association rule mining-based route recommendation and the genetic algorithm-based route optimization are discussed in the following subsections.

3.1. *Association Rule Mining Based Travel Route Recommendation*

Association rule mining concentrates on finding rules which will predict the occurrence of an item based on the occurrences of other items in a transaction [24]. The fact that two items are found to be related means co-occurrence but not causality. Association rule mining [25] can be used in basket data analysis, educational data mining, classification, clustering etc. Agrawal et al. introduce association rules [26] to discover regularities between products in large-scale transaction data from supermarkets. For example, the rule {egg} ⇒ {burger, potato} would indicate that is a customer buys eggs, they are very likely to also buy burger meat and potatoes together. In this paper, we apply the principles of association rule mining to discover relations between travel spots in the large travel history dataset, and strong rules are identified using some measures of interest value. Figure 2 gives as overview of the architecture of the proposed association rule mining-based approach for travel route recommendation. The designed application consists of three modules: route extraction module, frequent route mining module, and strong route generation module.

Apriori is a classic algorithm for learning association rules in computer science and data mining. This approach [27] was first designed to operate on databases containing transactions which are composed of a set of items or itemset (for example, collections of items purchased by customers). Some other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi) [28], or having no timestamps (DNA sequencing). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database. Constraints on various measures of significance and interest can be used in order to select credible rules from the set of all possible rules. The best-known constraints are minimum thresholds on support and confidence.

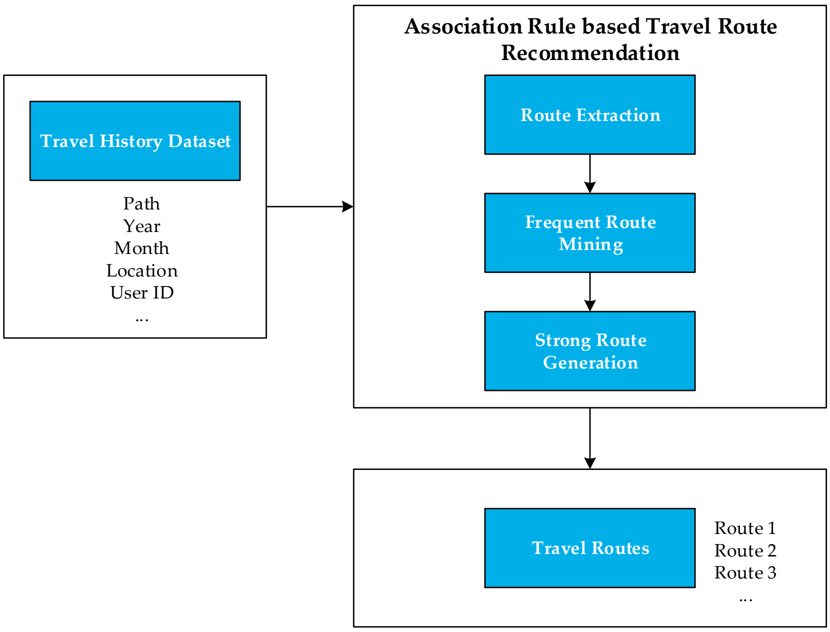


Figure 2. Architecture of the association rule mining travel route recommendation module.

The $supp(X)$ of an itemset X is defined as the proportion of transactions in the data set which contains the itemset.

$$supp(X) = \frac{number\ of\ transactions\ which\ contain\ the\ itemset\ X}{number\ of\ total\ transactions} \tag{1}$$

The confidence of a rule ($x \rightarrow y$) with respect to a set of transactions T, is defined as the proportion of the transactions that contains X which also contains Y.

$$conf(X \rightarrow Y) = supp(X \cup Y)/supp(X) \tag{2}$$

The pseudo code for the algorithm is given below for a set of itemsets, the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. A bottom-up approach is utilized in Apriori, where frequent subsets are extended one item at a time (also known as candidate generation), and groups of candidates are tested against by the data. This process is terminated by the algorithm until no further successful extensions are found. A breadth-first search strategy is used to count the support of itemsets efficiently and the counts of candidate itemsets are stored in a tree structure. Candidate itemsets of length k are generated by joining itemsets of length $k-1$ and those who have an infrequent subpattern are pruned. The candidate itemsets contain all frequent k-length itemsets according to the downward closure lemma. After that, frequent itemsets are determined among the candidates by scanning the transaction database. Association rules are simultaneously required to satisfy a user-specified minimum threshold of support and confidence. The method used to generate association rules can be split up into a two-step process as follows: apply the minimum support threshold to find all frequent itemsets and then form rules from these frequent itemsets in terms of the minimum confidence constraint. We have defined the following terminologies for the pseudo code such as L_1: frequent items, T: database, MS: minimum support, C_g: candidate generated from L_{k-1}, C_k: k size candidate set, t: transaction, C_m: candidate of C_k with minimum support, U_k: counting of the frequent itemsets.

Pseudo Code for Apriori (T, MS)

```
Input ($T$, $MS$)
Output: $U_k$
Begin:
$L_1 \leftarrow \{T\}$
        for $k \leftarrow$ 2 to $L_{k-1}$ do {
$C_k \leftarrow C_g$//cartesian product of $L_{k-1} \times L_{k-1}$ and eliminating any $k-1$ infrequent size items set
        for each $t$ in $T$ do {
$L_k \leftarrow C_m$
        od }
        od }
Return $U_k$
```

Generally, a travel route is a sequence of attractions in a specific order, for example, in terms of the order from left to right, the left-most point within the sequence is considered as the start point and the right-most one as the endpoint. Route extraction is the first step in the proposed travel route recommendation approach, which processes retrieving data out of data sources and recombines these data to generate routes for further data processing or data storage, as shown in Figure 3. In our proposed system, we extract more features such as geo-location, path, user Id, year, and month information from the travel dataset other than individual attractions to generate the route set on which we are applying the Apriori algorithm. The details of the used dataset will be illustrated in the next section.

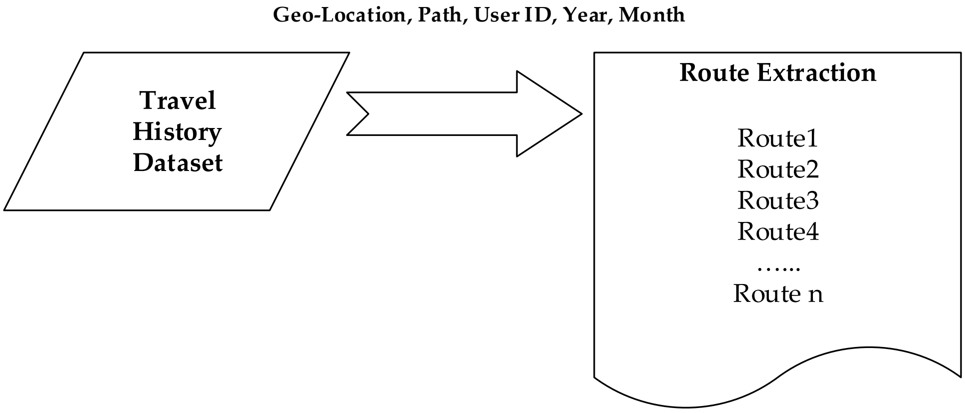


Figure 3. Route extraction from travel dataset.

The task for the frequent route mining is to find all common sets of individual attractions by which to compose travel routes that have at least a minimum support. To generate frequent routes, the following are requirements for an effective candidate generation procedure: First, it should avoid generating too many unnecessary candidates that is the key issue affecting the algorithm performance. Secondly, it must ensure that the candidate route is complete and lastly it should not generate the same route more than once. The Brute-Force [29] method is used in the proposed approach to generate all of the possibilities of routes. Each route in the lattice is a candidate frequent route and the support for each candidate is counted by scanning the dataset in order to match each route against every candidate. For given d items, the number of candidate routes generated at level k is equal to $\left(\frac{d}{k}\right)$, and the overall of this method is defined in Equation (3):

$$O\left(\sum_{k=1}^{d} k \times \left(\frac{d}{k}\right)\right) = O\left(d \cdot 2^{d-1}\right) \tag{3}$$

The set of possible route sets is the power set over I and has the size $2^n - 1$ of which the size grows exponentially in the number of items n in I. An efficient search is desired to minimize the number of candidates or reduce the size of datasets. The most common approach is to reduce the number of candidates using the Apriori principle [30]. This principle can be used to simplify the pattern generation process when mining patterns in data sets. This principle guarantees that if a candidate set is frequent, then all of its subsets must also be frequent and thus no infrequent itemset can be a subset. The Apriori principle holds due to the following property of the support measure as shown in Equation (4):

$$\forall X, Y : (X \in Y) \rightarrow s(X) \geq s(Y) \tag{4}$$

This is verified using the support measure because the support of an itemset never exceeds that of its subsets, which is also known as the anti-monotone property of support [31]. The Apriori is a practical implementation of the principle. The algorithm in this illustration is plotted to mine all frequent routes in the database. Multiple iterations are performed to searches in the database for finding frequent routes where k-item routes are used to generate k + 1-routes. Figure 4 illustrates the generation of the frequent route set using Apriori algorithm. The thresholds of the minimum support and minimum confidence should be predetermined before the algorithm starts to find frequent routes. After that, it scans the dataset to compute the support of each attraction. These attractions, with the support which is equal to or greater than the predefined minimum support, are added to the 1-route frequent sets $R1$ and this process is terminated at the point where no more frequent route sets are generated.

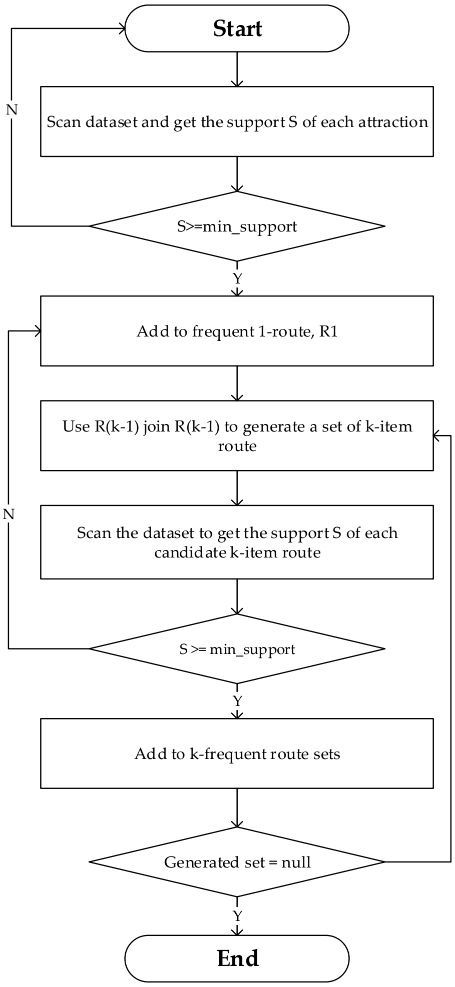

Start
Scan dataset and get the support S of each attraction
N
S>=min_support
Y
Add to frequent 1-route, R1
Use R(k-1) join R(k-1) to generate a set of k-item route
N
Scan the dataset to get the support S of each candidate k-item route
S >= min_support
Y
Add to k-frequent route sets
Generated set = null
Y
End

Figure 4. Flowchart of frequent route set generation.

Given a frequent route set R, the requirement of when the route can be added to the strong route is to find all nonempty subsets which meet the minimum confidence condition. As in the frequent route generation, we also need to look for a way to generate strong routes efficiently. In this paper, we generate strong routes by merging two routes that share the same prefix in the rule consequent. Following rules shown in Figure 5 are used to generate strong routes from the frequent route set:

- For each frequent route set R, find all nonempty subset of R.
- For each non-empty subset S of R, write the strong route $S \rightarrow (R\text{-}S)$ if support count of R/support count of $S \geq$ minimum confidence.

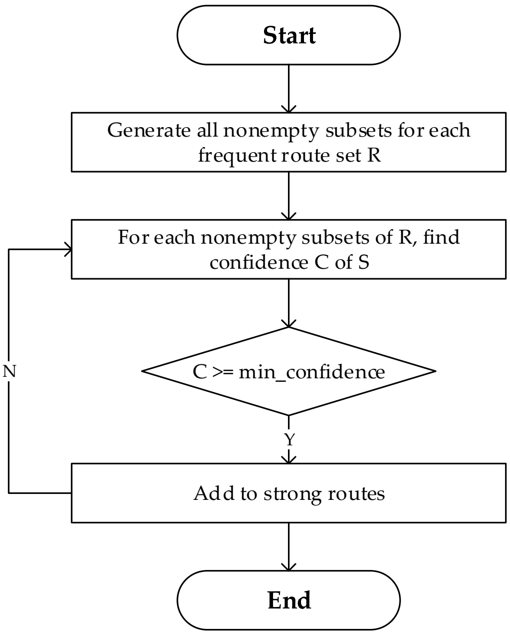

Start
Generate all nonempty subsets for each frequent route set R
For each nonempty subsets of R, find confidence C of S
C >= min_confidence
N
Y
Add to strong routes
End

Figure 5. Flowchart of strong route generation.

3.2. Genetic Algorithm Based Travel Route Optimization

A genetic algorithm [32] is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. It is frequently used to find optimal or near-optimal solutions to difficult problems in research and in machine learning, which would otherwise result in spending a lifetime solving the problem. Basically, a genetic algorithm operates on a finite population of chromosomes or bit strings. The search mechanism [33] is mainly composed of three phases as follows: fitness evaluation of each chromosome, selection of the parent chromosomes, and application of the mutation and recombination operators to the parent chromosomes. The next generation is formed by the new chromosomes derived from these operations, and the process is reduplicated until the system achieves the purpose. This paper utilizes the principles of a genetic algorithm to solve the optimization problem in the travel route recommendation.

Figure 6 basically defines the flowchart as an expression of the travel route optimization using the genetic algorithm. The process begins with a set of individuals which is called a population. In the proposed travel route optimization model, each individual is a solution characterized by a set of travel spots known as Genes, which contains the geo-information such as longitude and latitude. For given N travel spots, each place is identified with a unique numeric label in the range of 0 to $N-1$, where the order of the route is defined from left to right. For example, one route contains 4 places (0, 1, 2, 3, 4), where the start location is 0 and the end location is 4, can be encoded as follows:

$$place\ 0 \rightarrow place\ 1 \rightarrow place\ 2 \rightarrow\ place\ 3 \rightarrow place\ 4$$

Evaluation is the core component in the genetic algorithm, which accepts a population to generate a new generation. Selection, crossover and mutation are then determined by evaluation. The fitness function defines a function which takes a candidate solution to the problem as input and produces as output how "fit" the solution is with respect to the problem in consideration. The fitness function allows the system to determine which individuals will be selected for recombination and mutation at each phase of the genetic algorithm, where the basic rule is that the higher the fitness value, the better the individual. In this paper we define the distance fitness function for the solution of travel route optimization. For a given route $p_1, p_2, \cdots, p_n$, the total distance is defined to as shown in Equation (5).

$$total\ distance = \sum_{i=1}^{N} dist(p_i,\ p_{i+1}) \tag{5}$$

where $dist(p_i,\ p_{i+1})$ is the distance between place *i* and place *i* + 1. In order to describe the shorter routes with higher fitness score, a simple fitness function that calculates the inverse value of the route distance is used as shown in Equation (6).

$$fitness\ = \frac{1}{total\ distance} \tag{6}$$

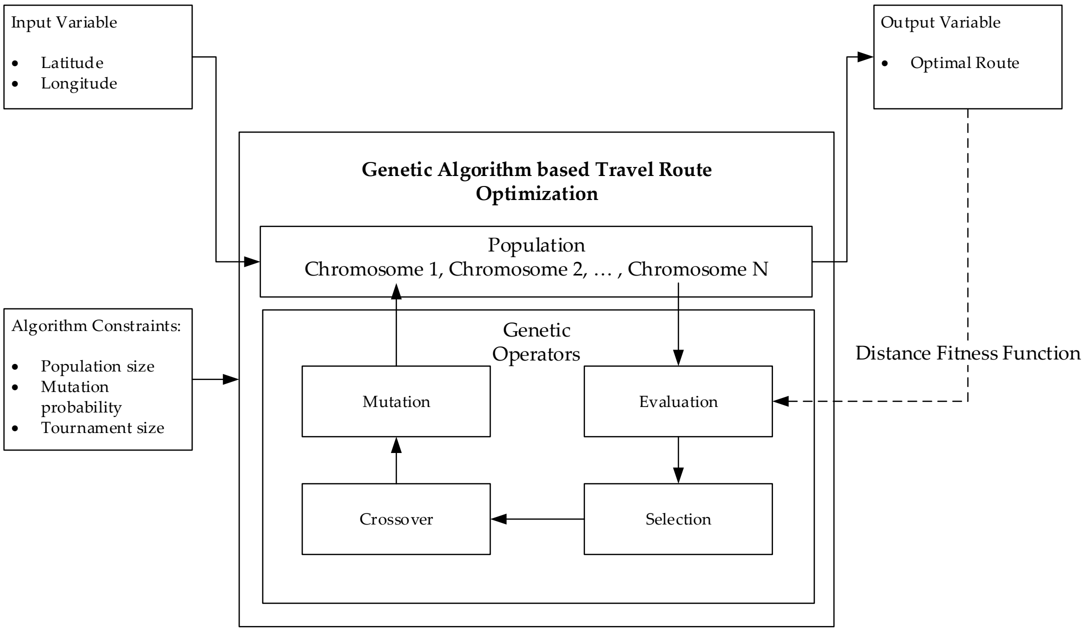


Figure 6. The process flow of genetic algorithm-based travel route optimization.

Parent selection is the process of selecting parents which mate and recombine to create offspring for the next generation. Individual solutions are selected through a stochastic fitness process with the principle that the fitter solutions have higher possibility to be selected as parents. In this paper, we apply the tournament selection method [34] as illustrated in Figure 7, where a number of individuals are selected from the population randomly and the best one is selected as the winner for crossover, the same process is repeated until the next parent is selected. Tournament size is also called tournament pressure as the size grows larger, the less chance of weak individuals to be selected. In this paper, we set the tournament size to 5%, which equals 5% of the population.

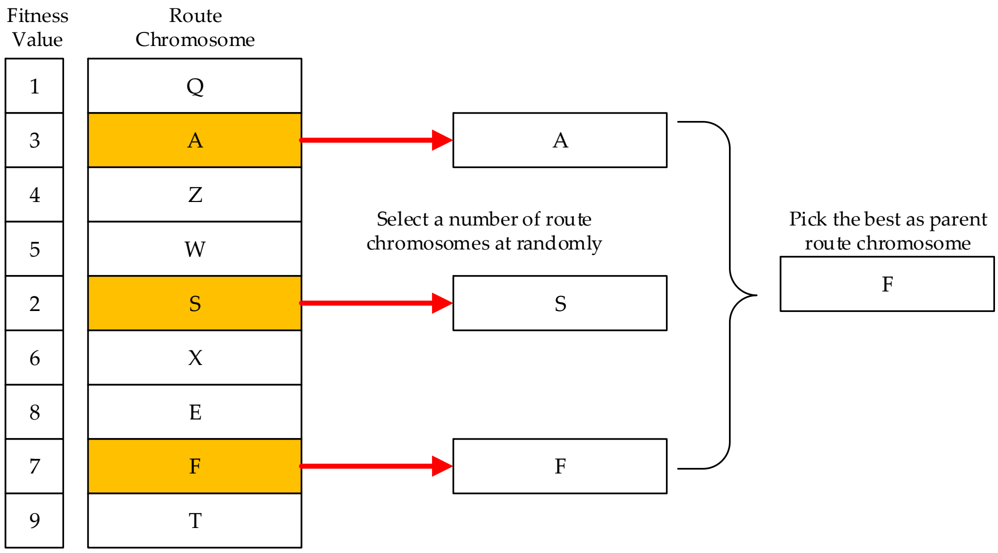
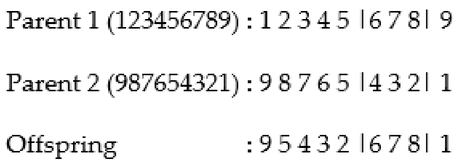

Fitness Value
Route Chromosome
1
Q
3
A
4
Z
5
W
2
S
6
X
8
E
7
F
9
T
A
Select a number of route chromosomes at randomly
S
F
Pick the best as parent route chromosome
F

Figure 7. Tournament selection on population.

Finding a solution to the case of travel route optimization requires a genetic algorithm to be set up in a specialized way. For instance, a valid solution would need to represent a route where every location is included at least, and only, once. A route would not be considered valid if it contains a single location more than once, or is a location is missed out completely. Hence, special types of mutation and crossover methods are needed to ensure the genetic algorithm does indeed meet this requirement. We utilize the order crossover method [35] which is able to generate a valid route. In this crossover method, two random cross points are defined and a subset is selected from the first parent and then copied into the same positions of the offspring. Any missing values are determined by the second parent and only non-repetitive genotype can be added to the offspring. Figure 8 is made so as to give a little clearer an explanation considering the following example, with two random cross points; 6 and 8. Here, a subset of the route is taken from the first parent (6, 7, 8) and copied to the same positions of the offspring. The genotype after the second cross point in the second parent (1) is not in offspring so that it can be added at position 9 of the offspring. Traversing the second parent circularly, we get the genotype (9, 8, 7, 6, 5) where 9 and 5 can be copied to the offspring, skipping the others. Finally, 4, 3, 2 are copied to the offspring at position 3, 4, and 5, respectively. This process is repeated until the offspring has no more empty values and the end result should be a complete route which contains all of the positions from its parents with no missing or duplicated positions.


Parent 1 (123456789) : 1 2 3 4 5 |6 7 8| 9

Parent 2 (987654321) : 9 8 7 6 5 |4 3 2| 1

Offspring : 9 5 4 3 2 |6 7 8| 1

Figure 8. Order crossover on two parent routes.

After the crossover, the mutation function is applied to each of the chromosomes to output the new generation. Mutation is an operator used to maintain the genetic diversity from one generation of a population to the next. Mutation basically modifies one or more genes in an individual from its initial state. The genetic algorithm comes to the better solution as the solution may change entirely from the previous solution by using mutation. However, for a travel route optimization problem, the mutation method should only be capable of shuffling the route without ever adding or removing a location from the route, which otherwise might risk creating an invalid solution. Swap function [36] is one type of appropriate method that can be used, which is the closest mutation operator in philosophy to the original mutation operator since it only shuffles the route rather than modifies the original one. For example, we select two locations from the route at random and simply swap their positions. The mutation occurs in a user-definable probability (in our case 1.5%), which should be set low as the search will turn into a primitive random search when it is set to high. A more transparent explanation is presented in Figure 9 by the following example, where swap mutation is applied to the given route, here, a new route with the same values is created in a different order by switching the positions of 2 and 6. The proposed mutation method would never create a route having missing or duplicate values compared with the original one as it only swaps the pre-existing values, which is exactly what the travel route optimization module is supposed to do.

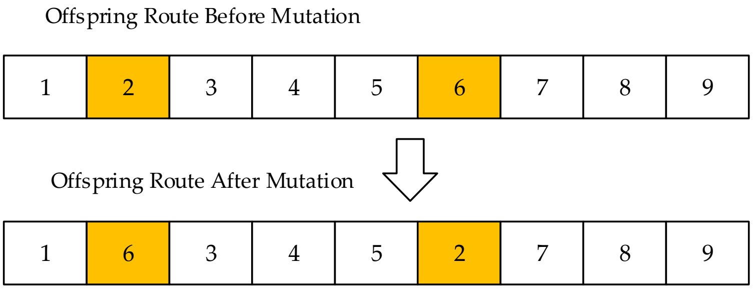

Offspring Route Before Mutation
1 2 3 4 5 6 7 8 9
Offspring Route After Mutation
1 6 3 4 5 2 7 8 9

Figure 9. Swap mutation on the offspring route.

The principle of survivor selection [37] is to take the good with the bad, or in other words, to determine which individuals should be kicked out and which should be kept in the next generation. This paper employs Elitism, which guarantees the fittest member of the current population is always propagated to the next generation and therefore, under no circumstance can the fittest member of the current population be replaced. We utilize the fitness-based selection, where the offspring tend to replace the least fit individual in the new population. As depicted in Figure 10, the generated offspring replaces the least fit individual P1 of the population in the original position. The termination condition is important as it determines when to stop running the algorithm. In this paper, the termination criterium is defined by using an absolute number of generations (in our case 1000) and thus the algorithm terminates until the counter reaches the predetermined count.

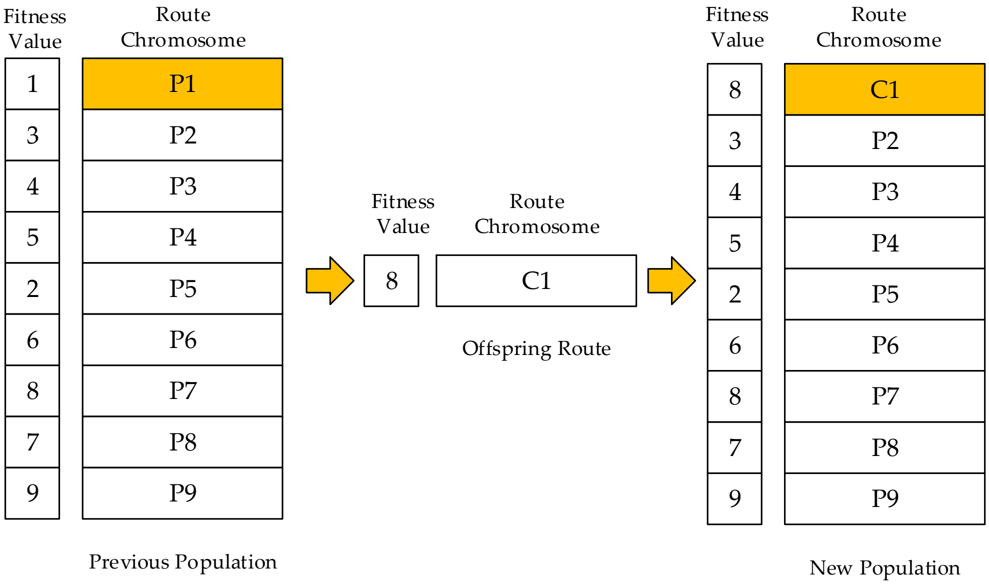
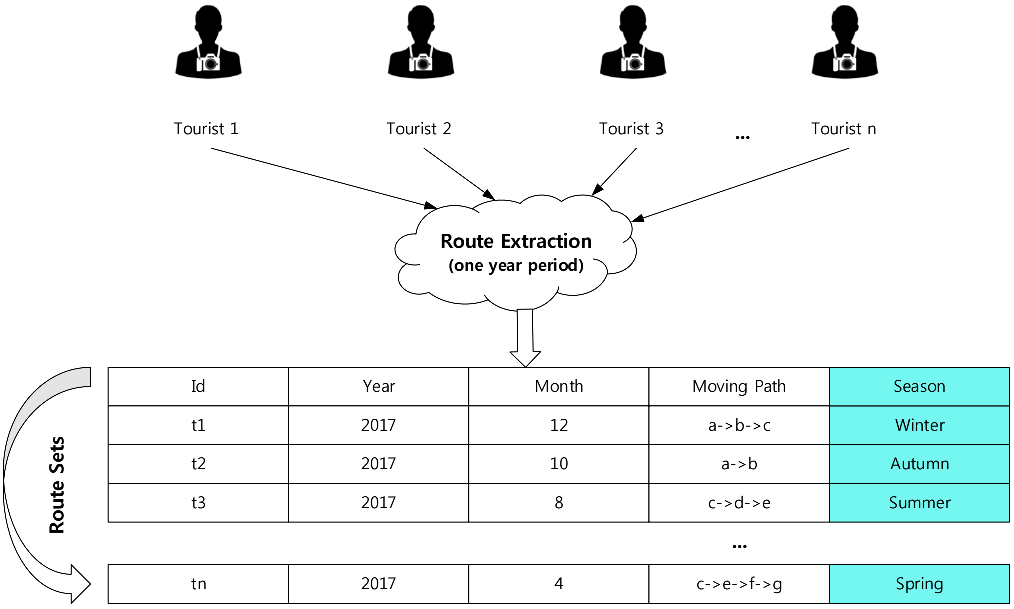


Figure 10. Fitness based selection for generating a new population.

4. Dataset and Experiment

This section represents the dataset that has been used for the experiment and reports the results. We use the open data which collected the travel information from 36,548 tourists who have been to Jeju in 2017. The Open Data Portal [38] releases this open data to the public, covering more than 130 competitive destinations throughout Jeju visited by these tourists. It includes moving path records from each tourist in the specific month of the year. Each path contains starting and destination points along with other travel spots between the two points. In general, the recommendation provided by a travel recommender system can be quite different with seasons. For example, a tourist is pleased to visit the beach in summer, but not so much in winter, when the weather is unpleasant due to cold climates. The extracted route set presented in Figure 11 contains various routes, which is composed of a moving path and some other features; for example, consider the route t1 as shown in the following sample, containing the user Id, year, month and a moving path of three locations (a, b, c). In keeping with the seasonality requirements of the recommender system, we assume that a single 'season' feature is specified in advance. This feature can be easily classified according to the month in the current year (i.e., December represents winter and August represents summer) and thus the route set can be categorized into four subsets.



Id	Year	Month	Moving Path	Season
t1	2017	12	a->b->c	Winter
t2	2017	10	a->b	Autumn
t3	2017	8	c->d->e	Summer
...				
tn	2017	4	c->e->f->g	Spring

Figure 11. A sample of the extracted route set from the dataset.

Table 1 reports the experimental results of the association rule based recommendation without considering seasonality. For this analysis, eight cases with different minimum support values from 0.03 to 0.1 are performed by the proposed system ten times at randomly selected system resource utilization levels. In this paper, the min support is specified in percentage instead of a single count due to the large volume of data. We set the minimum confidence to a constant (0.9) which indicates the high strength of rules generated. It can be seen from the table that the number of rules is decreased with the increment of the minimum support value. The number of best rules is 125 when the min support is 0.03 while the number of rules reduced to 29 when the min support is 0.1. Moreover, the minimum support has a great effect on the number of cycles which determines how long it takes to run. For example, the average run time taken is recorded to be 9.674 s when the minimum support is 0.03 and decreased by 39.4% when the minimum support is 0.04.

Table 1. Travel route recommendation performance evaluation.

Min Supp	Min Conf	No. Cycles	No. Best Rules	Average Run Time
0.03	0.9	8	125	9.674 s
0.04	0.9	7	125	5.860 s
0.05	0.9	7	61	5.336 s
0.06	0.9	7	61	5.036 s
0.07	0.9	5	61	4.272 s
0.08	0.9	5	29	2.420 s
0.09	0.9	5	26	2.264 s
0.1	0.9	5	29	2.073 s

Table 2 presents another experiment test considering seasonal factors, in which the original dataset is separated into four subsets according to the seasons. In this test, all cases are performed by the proposed system ten times at randomly selected system resource utilization levels under the same conditions, where minimum support is 0.07 and minimum confidence is set to 0.9. It is clear to see that the recommender system generates 125 rules for autumn, which is the most in all seasons. The main reason is that most of the tourists prefer to go sightseeing in the autumn since the climate is cooler and more comfortable than other seasons. Spring is the next after autumn with 61 rules, summer is the third with 29 rules and winter is the least with only 13 rules.

Table 2. Travel route recommendation performance evaluation in different seasons.

Season	Min Supp	Min Conf	No. Cycles	No. Best Rules	Average Run Time
Spring	0.07	0.9	4	61	1.102 s
Summer	0.07	0.9	5	29	0.581 s
Autumn	0.07	0.9	6	125	4.313 s
Winter	0.07	0.9	5	13	0.453 s

A sample list of recommended routes according to seasons is presented, as shown in Table 3. It can be seen from the table that the recommendation contains more outdoor venues in spring than in summer, which is reasonable due to the hot climate, and people are more likely to stay indoors. It is obvious to see that these routes contain the same locations, but in a different order. For example, we can generate 6 different routes in the case of 3 locations which can be simply calculated by the factorial of 3. However, if the number of locations in the route grows, the factorial will be gigantic, which makes it almost impractical to find the shortest route among such a huge number of different routes.

Table 3. A sample of the recommendation in spring and summer.

Season	Id	Moving Path
Spring	1	Jeongbang Falls, Chilshimni Food Street, Seobok Exhibition Hall, Jungmun Tourist Complex
	2	Seobok Exhibition Hall, Jeongbang Falls, Chilshimni Food Street, Jungmun Tourist Complex
	3	Chishimni Food Street, Jeongbang Falls, Seobok Exhibition Hall, Jungmun Tourist Complex
Summer	1	Play K-Pop Jeju, Teddy Bear Museum, Pacific Land, Cheonjeyeon Falls
	2	Teddy Bear Museum, Cheonjeyeon Falls, Pacific Land, Play K-Pop Jeju
	3	Cheonjeyeon Falls, Pacific Land, Play K-Pop Jeju, Teddy Bear Museum

In this paper, we use the genetic algorithm as the optimization technique which is capable of discovering the optimal solution to this problem. In order to demonstrate the performance of the proposed route optimization approach, we apply a test agenda with four instances as illustrated in Table 4. Each instance contains a list of 10 to 55 places. The values of some of the relevant parameters for the genetic algorithms are defined as follows: the population size is set to 100, the max generation is set to 1000, and the mutation rate is set at 1.5%.

Table 4. Test agenda for travel route optimization.

No	No. Places	Population Size	Max Generation	Mutation Rate
1	10	100	1000	1.5%
2	25	100	1000	1.5%
3	40	100	1000	1.5%
5	55	100	1000	1.5%

Selection Type: Tournament Selection; Crossover Type: Order Crossover; Mutation Type: Swap Mutation.

The performance has been analyzed and the results are reported in Table 5. For this analysis, four route sets of 10, 25, 40, 55 places are provided to the proposed route optimization module. Each of the sets is performed 20 times independently at randomly selected system resource utilization levels. The best result, average result, and optimum value are recorded. For the set with 10 places, the best result is recorded to be 789, averaging at the 801 and the optimum is 740. For the set with 25 places, the best result is recorded to be 1550, averaging at the 1620 and the optimum is 1475. Similarly, the test results for the rest instances are recorded, respectively, and all these results are visually demonstrated in Figure 12.

Table 5. Test results for each test instance.

No	No. Places	Best Result	Average Result	Optimum
1	10	789	801	740
2	25	1550	1620	1475
3	40	3460	3505	3250
4	55	7620	7720	7420

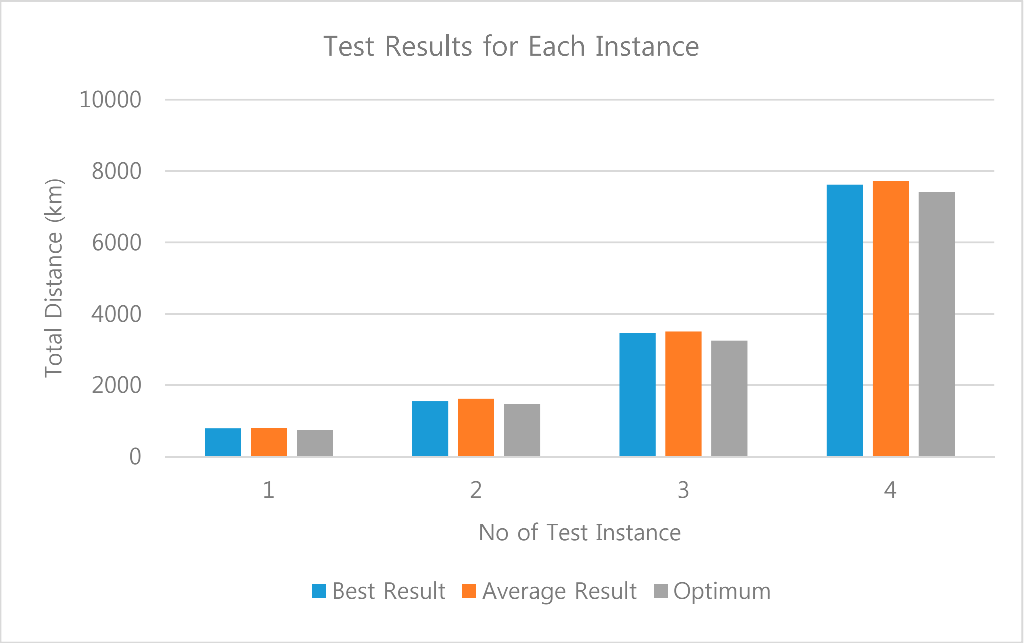


Figure 12. Performance analysis graph of different test instances.

5. Implementation Details

This section illustrates the development tools, hardware, and technologies used in the implementation. The proposed work comprises of two main components as shown in Figure 1 so that the development stacks are independently summarized into two tables for each of components.

Table 6 depicts the development toolkits and technologies used for the optimal travel route recommender system. The implementation is performed on the development machine with Intel(R) Core i3-3220 CPU at 3.30 GHz, 12 GB memory and Windows 7 Ultimate 64 bits operating system. The implementation platform is Eclipse Luna IDE using Java programming language. The dataset used is formed in the CSV format so that the opencsv is used, which is a Java specified CSV parser library for reading or writing CSV files.

Table 6. Development environment of the optimal travel route recommender system.

Component	Characterization
Operating System	Windows 7 Ultimate 64 bits
CPU	Intel(R) Core i3-3220
Memory	12 GB
IDE	Eclipse Luna (4.4.2)
Library and Framework	opencsv
Programming Language	Java

Table 7 introduces the development stack for developing the mobile tourist application. This mobile application is built in Java and XML programming languages using Android Studio IDE. SQlite 3 is a widespread database engine used for an embedded system such as mobile phones. It is used to store the routes generated from the recommender system. Daum MAP API provides various functionalities to customize maps with users' own content and imagery for display on mobile devices. YouTube Data API allows the user to add a variety of YouTube features to the application, for instance, it supports the functionalities to search for videos matching specific search terms, topics, and locations. The kma API captures the real-time weather data for any location including wind, temperature, humidity, and more. This application is tested in the Galaxy S4 with Octa-core (8-core CPU), 2GB memory and Android 4.2.2.

Table 7. Development stack of the mobile tourist application.

Component	Characterization
Model Name	Galaxy S4 SHV-E300K
Operating System	Android 4.2.2
CPU	Octa-core (4 × 1.6 GHz Cortex-A15 & 4 × 1.2 GHz Cortex-A7)
Memory	2 GB
IDE	Android Studio 3.1.2
DBMS	SQlite3
Library and Framework	Daum Map API, kma API, YouTube Data API
Programming Language	Java, XML

6. Use Case Study: Mobile Tourist

In order to demonstrate the usability of the proposed system, a mobile tourist case study has been implemented as part of the experiment. This application ingests the travel routes generated from the travel route recommender system and these routes are stored in the local database for further processing and various services. The main interface as shown in Figure 13 provides an overview of this application, which mainly shows a map of Jeju. The real-time weather information is displayed in different directions on the map. There are also several options at the bottom of the application, which provide the entries for different function interfaces.

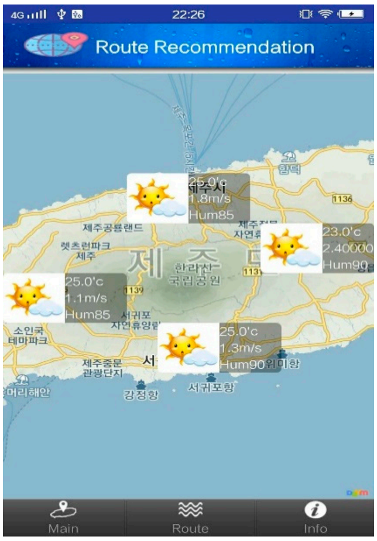
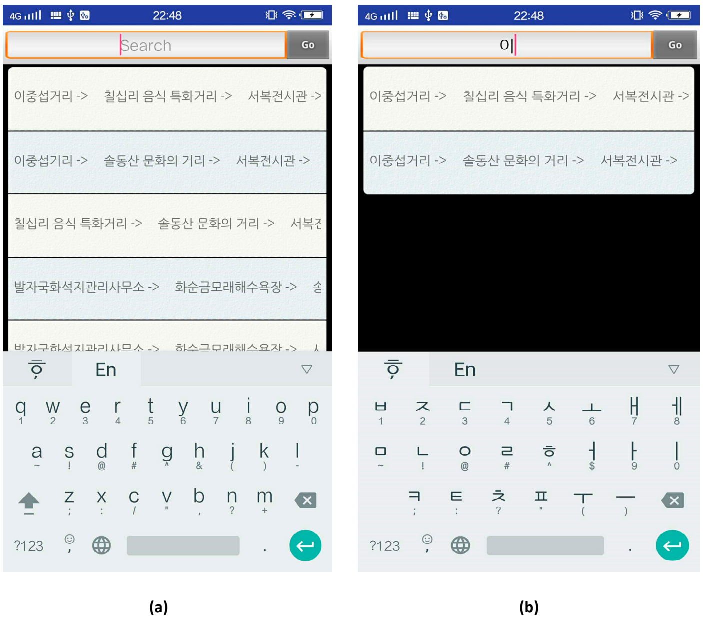


Figure 13. Mobile tourist application main interface.

The main function of this application is to provide travel routes depending on the start point entered by the user. Figure 14 presents the interface of route searching, in which users can retrieve all available routes or search the specific route by keywords.



이중섭거리 -> 칠십리 음식 특화거리 -> 서복전시관 ->

이중섭거리 -> 솔동산 문화의 거리 -> 서복전시관 ->

칠십리 음식 특화거리 -> 솔동산 문화의 거리 -> 서복?

발자국화석지관리사무소 -> 화순금모래해수욕장 -> 송

이중섭거리 -> 칠십리 음식 특화거리 -> 서복전시관 ->

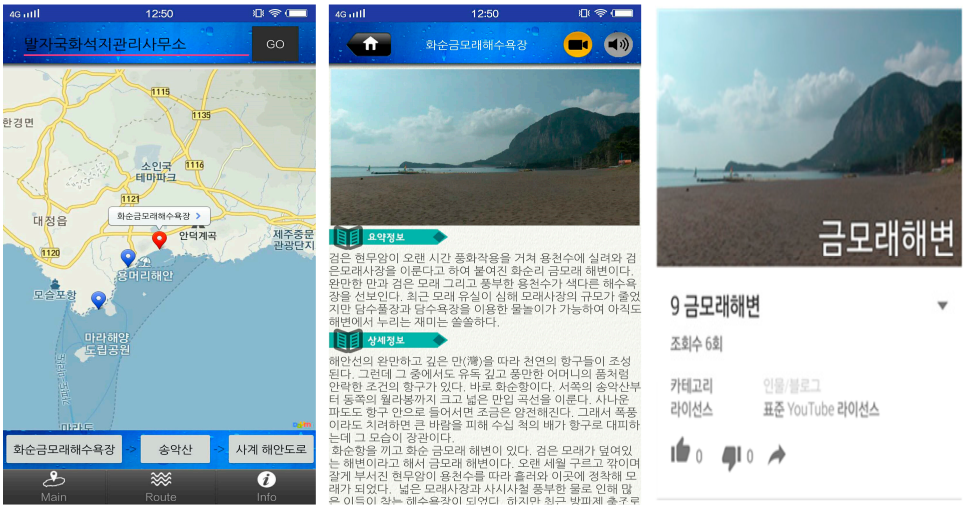
이중섭거리 -> 솔동산 문화의 거리 -> 서복전시관 ->

(a)

(b)

Figure 14. (**a**) travel route list form; (**b**) search route using keywords.

Figure 15 shows some snapshots of various interfaces for travel routes and spots. Figure 15a shows the overview of travel routes located on the map along with the complete route information which is displayed at the bottom of the map. Each travel spot from the route is labeled with a balloon marker and it is necessary to notice that the balloon marker in red represents the current location while the marker in blue stands for the next destination of the route. Furthermore, the map also renders a bubble above the marker that displays the name of a given spot. By clicking the given marker, this application redirects to another interface as represented in Figure 15b, where the tourism information of the selected travel spot is detailed. Figure 15c presents the interface for displaying the related introductory videos of the travel spot from YouTube.



검은 현무암이 오랜 시간 풍화작용을 거쳐 용천수에 실려와 검은모래사장을 이룬다고 하여 붙여진 화순리 금모래 해변이다. 완만한 만과 검은 모래 그리고 풍부한 용천수가 색다른 해수욕장을 선보인다. 최근 모래 유실이 심해 모래사장의 규모가 줄었지만 담수풀장과 담수욕장을 이용한 물놀이가 가능하여 아직도 해변에서 누리는 재미는 쏠쏠하다.

해안선의 완만하고 깊은 만(灣)을 따라 천연의 항구들이 조성된다. 그런데 그 중에서도 유독 깊고 풍만한 어머니의 품처럼 안락한 조건의 항구가 있다. 바로 화순항이다. 서쪽의 송악산부터 동쪽의 월라봉까지 크고 넓은 만입 곡선을 이룬다. 사나운 파도도 항구 안으로 들어서면 조금은 얌전해진다. 그래서 폭풍이라도 치려하면 큰 바람을 피해 수십 척의 배가 항구로 대피하는데 그 모습이 장관이다.

화순항을 끼고 화순 금모래 해변이 있다. 검은 모래가 덮여있는 해변이라고 해서 금모래 해변이다. 오랜 세월 구르고 깎이며 잘게 부서진 현무암이 용천수를 따라 흘러와 이곳에 정착해 모래가 되었다. 넓은 모래사장과 사시사철 풍부한 물로 인해 많은 이들이 찾는 해수욕장이 되었다. 하지만 최근 방파제 축조로

(**a**) (**b**) (**c**)

Figure 15. (**a**) travel route with markers; (**b**) travel spot detailed info; (**c**) travel spot introductory video.

7. Comparison and Significance

This section provides a comparative analysis of the proposed travel route recommender system with some of the similar works mentioned in the Related Work Section. A benchmark study was carried out using the following features to reveal the superiority and applicability of the proposed work and the evaluation results are shown in Table 8. It can be seen from the table that the proposed system performs better than other systems in many respects. Most of the systems only provide an individual recommendation, neither utilize personal attributes nor supports route generation, which greatly reduces the conveniences for users. The work [24] is a somewhat similar approach with the highest variety of travel recommendation related services. This system utilizes the personal attributes-based mining approach to generate travel routes and shows the recommendation results in smartphones. However, route optimization is not considered which might cause the recommendation to be not reasonable. This limitation is a common issue that existed among most of the current systems. Moreover, many systems do not support mobile applications, which is another major limitation.

Table 8. Comparative analysis of the proposed system with the related systems.

Name	Personal Attributes	Route Generation	Data Store	Route Optimization	Mobile Support
[12]	No	No	Yes	No	No
[13]	No	No	Yes	No	No
Gothere [14]	No	No	Yes	No	No
[16]	No	No	Yes	No	No
[17]	No	No	Yes	No	Yes
[19]	No	Yes	Yes	No	No
[20]	No	Yes	Yes	No	Yes
W2go [21]	No	Yes	Yes	No	No
[22]	Yes	Yes	Yes	No	No
[23]	Yes	Yes	Yes	No	Yes
Proposed System	Yes	Yes	Yes	Yes	Yes

This work proposes a real-life case study for tourists in Jeju to verify the feasibility and practicability of the proposed approach. The travel and tourism industry is huge and it was estimated to be a 6.3 trillion dollar industry in 2015. The impact of the proposed solution can be summarized as follows: first, it can change the conventional tourism industry since tourists do not need to check guidebooks or periodicals for planning their upcoming tours. Nowadays, more than 150 million tourists use smartphones, and for this reason, they can use the mobile tourist application with just a single click. Moreover, the proposed solution can greatly affect the on-site travel behavior and serve as a basis for future enhancements in the tourism industry since it provides personalized services by analyzing travelers' experience during their trip. We hold the opinion that the designed system has the potential to be expanded in other scenarios such as hotel and restaurant recommendation, which can additionally benefit from the significance of the work. Furthermore, the demands to have a reliable and user-friendly travel guide application that offers personal attributes mining, travel route generation, and travel route optimization are growing rapidly, and this paper aims to reveal the potential to solve all these issues mentioned above.

8. Conclusions and Future Direction

This work outlines a novel procedure for the design and implementation of a travel route recommender system to help aid in both the trip planning as well as instantaneous travel information. An association rule mining-based approach which considers personal travel attributes is proposed to produce a travel route including various destinations. The biggest innovation of the presented work is the utilization of a genetic algorithm to optimize the travel route, by referring to the distance fitness function in order to find the optimal route. A series of experiments are performed using the open access travel data in Jeju from January to December 2017, permitting efficient and reliable route recommendation and optimization according to user personal attributes. Furthermore, a mobile tourist application has been implemented as the proof of concept and experimentally tested in Jeju, Korea to validate the performance of the proposed approach. The significance of this work has been highlighted by a comparative analysis of the designed system with existing approaches, and the result demonstrates that the designed system outperforms other systems in many aspects. It is the goal of this work to fill gaps in mobile-based travel planning applications, to make the life of a tourist easier, and to accelerate tourism industry developments. Moreover, the proposed work can be expanded to many other application scenarios such as hotel reservations and car rentals on the basis of the original application. Future research directions include deploying the designed application into a larger application domain for all kind of travel purposes while using a much wider range of datasets that cover the whole of Korea.

Author Contributions: L.H. conceived the idea for this paper, designed the experiments and wrote the paper; S.-H.K. implemented the mobile tourist application for the use case study; W.J. offered the dataset and the theory support; D.-H.K. conceived the overall idea of travel route recommendation for mobile tourists, and proof-read the manuscript, and was correspondence related to this paper.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2016-0-00313) supervised by the IITP (Institute for Information & communications Technology Promotion), and this research (paper) was performed for the Development of Radar Payload Technologies for Compact Satellite in Korea Aerospace Research Institute, funded by the Ministry of Science and ICT. Any correspondence related to this paper should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. TripSpot. Available online: http://www.tripspot.com/ (accessed on 8 January 2018).
2. Sang, J.; Mei, T.; Sun, T.J.; Li, S.; Xu, C. Probabilistic sequential POIs recommendation via check-in data. In Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 402–405.
3. Bao, J.; Zheng, Y.; Wilkie, D.; Mokbel, M. Recommendations in location-based social networks: A survey. *Geoinformatica* **2015**, *19*, 525–565. [CrossRef]

4. Zhao, S.; King, I.; Lyu, M.R. A Survey of Point-of-interest Recommendation in Location-based Social Networks. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
5. Lorenzi, F.; Loh, S.; Abel, M. PersonalTour: A recommender system for travel packages. In Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligent Agent Technology (IAT 2011), Lyon, France, 22–27 August 2011.
6. Resnick, P.; Varian, H.R. Recommender systems. *Commun. ACM* **1997**, *40*, 56–58. [CrossRef]
7. Lu, X.; Wang, C.; Yang, J.; Pang, Y.; Zhang, L. Photo2trip: Generating travel routes from geo-tagged photos for trip planning. In Proceedings of the ACM Multimedia 2010 International Conference, Firenze, Italy, 25–29 October 2010; pp. 143–152.
8. Shi, Y.; Serdyukov, P.; Hanjalic, A.; Larson, M. Personalized landmark recommendation based on geo-tags from photo sharing sites. In Proceedings of the 5th AAAI Conference Weblogs Social Media, Barcelona, Spain, 17–21 July 2011; Volume 11, pp. 622–625.
9. Huang, H.; Gartner, G. Using trajectories for collaborative filtering–based POI recommendation. *Int. J. Data Min. Model. Manag.* **2014**, *6*, 333–346. [CrossRef]
10. Cheng, C.; Yang, H.; Lyu, M.R.; King, I. Where you like to go next: Successive point-of-interest recommendation. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2605–2611.
11. Kurashima, T.; Iwata, T.; Irie, G.; Fujimura, K. Travel route recommendation using geotags in photo sharing sites. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010; pp. 579–588.
12. Clements, M.; Serdyukov, P.; de Vries, A.; Reinders, M. Personalised travel recommendation based on location co-occurrence. *arXiv*, 2011.
13. Cheng, C.; Yang, H.; King, I.; Lyu, M.R. Fused matrix factorization with geographical and social influence in location-based social networks. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; Volume 12, pp. 17–23.
14. Majid, A.; Chen, L.; Chen, G.; Mirza, H.; Hussain, I. Gothere: Travel suggestions using geotagged photos. In Proceedings of the 21st International Conference on World Wide Web, Hyderabad, India, 28 March–1 April 2011; pp. 577–578.
15. “Flickr”. Available online: https://www.flickr.com/ (accessed on 10 April 2018).
16. Jiang, S.; Qian, X.; Shen, J.; Fu, Y.; Mei, T. Author topic model based collaborative filtering for personalized POI recommendation. *IEEE Trans. Multimedia* **2015**, *17*, 907–918. [CrossRef]
17. Bao, J.; Zheng, Y.; Mokbel, M.F. Location-based and preference-aware recommendation using sparse geo-social networking data. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 199–208.
18. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 791–800.
19. Majid, A.; Chen, L.; Chen, G.; Mirza, H.T.; Hussain, I.; Woodward, J. A context-aware personalized travel recommendation system based on geotagged social media data mining. *Int. J. Geogr. Inform. Sci.* **2013**, *27*, 662–684. [CrossRef]
20. Lou, P.; Zhao, G.; Qian, X.; Wang, H.; Hou, X. Schedule a rich sentimental travel via sentimental POI mining and recommendation. In Proceedings of the 20th ACM International Conference Multimedia Big Data, Taipei, Taiwan, 20–22 April 2016; pp. 33–40.
21. Gao, Y.; Tang, J.; Hong, R.; Dai, Q.; Chua, T.; Jain, R. W2go: A travel guidance system by automatic landmark ranking. In Proceedings of the International Conference Multimedia, Firenze, Italy, 25–29 October 2010; pp. 123–132.
22. Yuan, Q.; Cong, G.; Sun, A. Graph-based point-of-interest recommendation with geographical and temporal influences. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 659–668.
23. Chen, Y.Y.; Cheng, A.J.; Hsu, W.H. Travel Recommendation by Mining People Attributes and Travel Group Types from Community-Contributed Photos. *IEEE Trans. Multimedia* **2013**, *15*, 1283–1295. [CrossRef]

24. Mishra, V.; Mishra, T.K.; Mishra, A. Algorithms for Association Rule Mining: A General Survey on Benefits and Drawbacks of Algorithms. *Int. J. Adv. Res. Comput. Sci.* **2013**, *4*, 8.
25. Abdel-Basset, M.; Mohamed, M.; Smarandache, F.; Chang, V. Neutrosophic Association Rule Mining Algorithm for Big Data Analysis. *Symmetry* **2018**, *10*, 106. [CrossRef]
26. Agrawal, R.; Imieliński, T.; Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD '93), Washington, DC, USA, 25–28 May 1993; p. 207.
27. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, 12–15 September 1994; Bocca, J.B., Jarke, M., Zaniolo, C., Eds.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1994; pp. 487–499.
28. Mannila, H.; Toivonen, H.; Inkeri Verkamo, A. Discovery of Frequent Episodes in Event Sequences. *Data Min. Knowl. Discov.* **1997**, *1*, 259–289. [CrossRef]
29. Bayardo, R.J., Jr. Brute-Force Mining of Highkonfidence Classification Rules. In Proceedings of the KDD 97 Proceedings, Newport Beach, CA, USA, 14–17 August 1997; pp. 123–126.
30. Agrawal, R.; Srikant, R. Mining sequential patterns. In Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, 6–10 March 1995; pp. 3–14.
31. Pei, J.; Han, J.; Lakshmanan, L.V.S. Mining frequent item sets with convertible constraints. In Proceedings of the Proceedings 17th International Conference on Data Engineering (ICDE'01), Heidelberg, Germany, 2–6 April 2001; pp. 433–442.
32. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [CrossRef]
33. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1996; ISBN 9780585030944.
34. Coello, C.A.C.; Montes, E.M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv. Eng. Inform.* **2002**, *16*, 193–203. [CrossRef]
35. Ono, I.; Yamamura, M.; Kobayashi, S. A genetic algorithm for job-shop scheduling problems using job-based order crossover. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 547–552.
36. Louis, S.J.; Tang, R. Interactive genetic algorithms for the traveling salesman problem. In Proceedings of the 1999 Genetic and Evolutionary Computing Conference (GECCO 1999), Orlando, FL, USA, 13–17 July 1999; pp. 385–392.
37. Ting, C.K.; Ko, C.F.; Huang, C.H. Selecting survivors in genetic algorithm using tabu search strategies. *Memet. Comput.* **2009**, *1*, 191–203. [CrossRef]
38. Open Data Portal. Available online: https://www.data.go.kr/main.do?lang=en (accessed on 15 May 2018).