Noname manuscript No.
(will be inserted by the editor)

Two Phase Heuristic Algorithm for the Multiple-Travelling Salesman Problem

Xiaolong Xu · Hao Yuan · Mark Liptrott · Marcello Trovati

Received: date / Accepted: date

Abstract The Multiple-Travelling Salesman problem (MTSP) is a computationally complex combinatorial optimisation problem, with several theoretical and real-world applications. However, many state-of-the-art heuristic approaches intended to specifically solve MTSP, do not obtain satisfactory solutions when considering an optimised workload balance. In this article, we propose a method specifically addressing workload balance, whilst minimising the overall travelling salesman's distance. More specifically, we introduce the *Two Phase Heuristic algorithm* (TPHA) for MTSP, which includes an improved version of the $K-$means algorithm by grouping the visited cities based on their locations based on specific capacity constraints. Secondly, a route planning algorithm is designed to assess the ideal route for each the above sets. This is achieved via the Genetic Algorithm (GA), combined with the roulette wheel method with the elitist strategy in the design of the selection process. As part of the validation process, a mobile guide system for tourists based on the Baidu electronic map is discussed. In particular, the evaluation results demonstrate that TPHA achieves a better workload balance whilst

Xiaolong Xu
of Computer Science Nanjing University of Posts and Telecommunications
Nanjing, China E-mail: xuxl@njupt.edu.cn

Hao Yuan
State Key Laboratory of
Information Security,
Chinese Academy of Sciences Beijing, China E-mail: 1214043029@njupt.edu.cn

Mark Liptrott
Department of Computing,
Edge Hill University
Ormskirk, United Kingdom, E-mail: liptrotm@edgehill.ac.uk

Marcello Trovati
Department of Computing,
Edge Hill University
Ormskirk, United Kingdom, E-mail: trovatim@edgehill.ac.uk

minimising of the overall travelling distance, as well as a better performance in solving MTSP compared to the route planning algorithm solely based on GA.

Keywords Multiple-Travelling Salesman Problem · Route Planning · Heuristic Algorithm

1 Introduction

The Salesman Problem (TSP) is typically an NP-problem [1], whose objective is to determine the shortest path across a set of randomly located cities, each of them visited only once (with the exception of the starting point). From a graph theory perspective, the core task of TSP is to obtain a minimum Hamiltonian cycle. However, some real-world problems cannot be modelled via a traditional simple TSP with one single salesman, including personnel scheduling [2], patrol planning [3], and goods distributing [4] [5]. To address this issue, *Multiple TSP* (MTSP) has been specifically designed to consider a multiple-travelling salesman problem.
The aim of MTSP is to minimise the overall distance across n cities where m salesmen start and complete their journeys at the same city, and the other locations are visited only once. Clearly, TSP is a special case of MTSP for $m = 1$. Heuristic approaches can be utilised to solve MTSP [6], such as the Ant Colony Optimisation algorithm (ACO) [7], Particle Swarm Optimisation algorithm (PSO) [8], and the Genetic Algorithm (GA) [9], to name but a few. However, there are a variety of aspects, which require further improvements. For example, ACO has a slow convergence speed, PSO tends to have local optimisation issues, and GA is prone to be trapped in the premature convergence and heavily depends on the initial population.

In this article, we propose a heuristic MTSP algorithm to obtain an optimised workload balance, whilst minimising the overall salesmen's travelling distance. The main contributions of our method include:

1. An extension of the current research on the balance problem associated with MTSP, which can be regarded as a multi-objective programming problem. In particular, MTSP and its objective function are initialised to consider the most appropriate conditions, which include the shortest distance and the minimised difference of the distance travelled by each salesman.
2. A specific focus on the integration between the workload balance and the minimisation of the overall travelling salesmen's distance. To achieve this, we propose the two phase heuristic algorithm (TPHA) for MTSP. In the first phase, we improve the $K-$means algorithm by grouping all cities into m subsets depending on their locations, based on some capacity constraints. In the second phase, the route planning algorithm based on GA is designed to obtain the ideal route for each city.

3. The roulette wheel method is combined with the elitist strategy to design the GA selection operation. Not only does the operation preserve the best individuals, but it also ensures that the most appropriate fitness of the individual is identified for the next generation. Subsequently, the method of cross operation is utilised to select the modified order crossover (MOC), which is based on the gene segment as the initial step. This operation can produce a new generation, even if the two parent chromosomes are identical. The main benefit of MOC includes an approach to address local optima and premature convergence.
4. The planning of the travel route through specific scenic locations over a period spanning several days was used as testing platform. A mobile guide system for tourists was designed, which is based on the Baidu electronic map and on TPHA. Firstly, the system locates the visitors' positions via GPS, and subsequently its utilises TPHA to provide the appropriate route plan through a balanced number of scenic locations for each day. The overall travelling distance is also minimised.

The rest of the paper is organised as follows: in Section 2, we discuss the current state-of-the-art methods and techniques in the field. Section 3 provides a detailed description of MTSP in detail and the mathematical model used in this context is introduced. Section 4 focuses on the properties of TPHA, and in Section 5 the evaluation results are discussed. Finally, Section 6 concludes the article by summarising its main contributions discussing future research directions.

2 Related Work

There is extensive research on TSP, including TSP with time windows [10], TSP with minimum ratio [14]. Most of the existing solutions consider different constraints, whilst finding a minimum Hamiltonian cycle. Currently, the approaches to TSP are divided into exact and approximate algorithms. The former mainly include dynamic programming [12], branch and bound [13], integer linear programming [15], etc. However, if the scale of the TSP becomes too large, its overall computational time and solution space will increase exponentially.

Inspired by biological activities or natural phenomena, some well-known heuristic algorithms have been developed to solve large scale TSPs, including ACO, PSO, and GA. There are significant research opportunities in the improvement of such heuristic algorithms and their combination. Pham et al. [15] propose two new crossover operators to improve the global ergodic property of GA, which is a better solution for classical TSP, but not for complex TSP with multiple constraints. Chen et al. [16] introduce an improved dynamic programming algorithm to deal with large-scale data, used as crossover and mutation operator in GA. Atif et al. [17] integrate $K-$means algorithm [18] with the greedy algorithm and Lin Kernighan's algorithm [19] to create an enhanced solution for large-scale TSP. However, this method is significantly

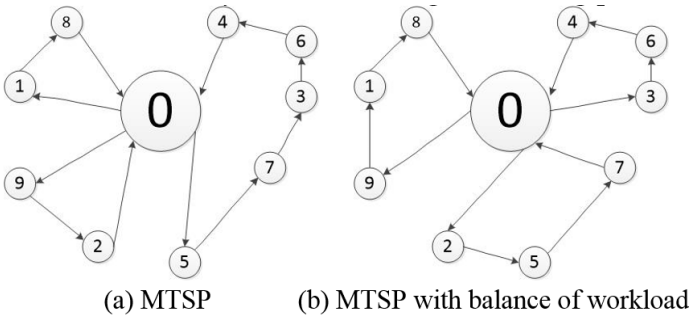


Fig. 1 The example described in Section 3.

affected by the scale of the subset partition.
A *Multi-Travelling Salesman Problem* (MTSP) is characterised by more than one salesman and as a consequence, MTSP typically has a higher level of complexity compared to TSP. MTSP can be converted into TSP, and Gorenstein [20] propose a basic strategy to achieve this, based on m salesmen, and $m-1$ virtual cities. These are used to define a gap between different travelling salesmen, whilst the distance between the virtual cities is considered infinite. Yuan et al. [21] discuss a new crossover operator called two-part chromosome crossover for the genetic algorithm in order to obtain near-optimal solutions of MTSP. However, this method is affected by the growth of the chromosome length and the overall cost of the solution. Kaliaperumal et al. [22] present the *Modified Two-Part Chromosome Crossover* to address MTSP by employing a genetic algorithm for nearby optimal solutions. However, this method allocates a different number of the cities for each salesman, and therefore it cannot successfully address MTSP with workload balance. Osaba et al. [23] propose the *Real-World Dial-a-Ride* problem, which is modelled as a MTSP. In particular, they propose GELS-GA, a new hybrid algorithm, which achivies optimal values even in highly complex scenarios. Finally, Alves et al. [24] consider the workload balance of MTSP, and develop GA to reduce both the overall distance and the difference between the distances travelled by each salesman.

3 Modelling of the MTSP

Typically, MTSP only aims to obtain the least total cost of distance or time. For example, Figure 1 depicts a scenario defined by 3 salesmen and 9 cities, where node 0 is the starting and ending point.

As shown in Figure 1(a), there are two salesmen traversing two cities respectively, whereas all the other ones are traversed by the third salesman. It is clear that the salesmen's workloads are unbalanced. As discussed above, the main objective of MTSP is to minimise the overall distance travelled by all salesmen, which may cause an unbalanced workload problem. Figure 1(b) shows an ideal solution for MTSP with workload balance. In order to achieve

a balanced allocation of workload, there are typically two different strategies. The first aims to balance the number of cities assigned to each salesman, whereas the second focuses on optimising the balance of the distances travelled by each salesman.

Formally speaking, MTSP with m salesmen and n cities is defined as a complete graph

$$G = G(V, E), \tag{1}$$

where $V = \{v_1, \ldots, v_n\}$ represents cities with the starting location at vertex v_0, and each edge (v_i, v_j) is associated with a weight d_{ij} ($d_{ij} > 0$, $d_{ii} = \infty, v_i, v_j \in V$), which represents the cost of the path (in terms of distance or time) between cities i and j. We define the maximum number of cities Q for each salesman to achieve workloads balance as

$$Q = \lceil \frac{n}{m} \rceil \tag{2}$$

As a consequence, one of the objectives of MTSP is to determine m sequences of Hamiltonian cycles over G, with the least total cost so that each salesman visit one and only one city. The objective function of MTSP is therefore

$$\min Z = \sum_{t=0}^{m} \sum_{i=0}^{n} \sum_{j=0}^{n} r_{ijt}\, d_{ij}, \tag{3}$$

subject to

$$\sum_{t=0}^{m} r_{ti} = 1 \quad i = 1, \ldots, n \tag{4}$$

$$\sum_{i=0}^{n} r_{ijt} = y_{tj} \quad j = 1, \ldots, n \ \forall t \tag{5}$$

$$\sum_{i=0}^{n} r_{ijt} = y_{ti} \quad i = 1, \ldots, n \ \forall t \tag{6}$$

$$\sum_{i=0}^{n} y_{ti} \leq Q \quad t = 1, \ldots, m \tag{7}$$

where

- i refers to a city ($i = 1, \ldots, n$), so that the starting location is 0;
- t refers to a salesman ($t = 1, \ldots, m$), where m is the total number of salesmen;
- d_{ij} represents the distance between the cities i and j;
- Q is the maximum number of cities travelled by each salesman;
- $r_{ijt} \in \{0, 1\}$, such that $r_{ijt} = 1$ if the salesman t travels directly from city i to city j, and $r_{ijt} = 0$, otherwise;
- $y_{ti} \in \{0, 1\}$ and $y_{ti} = 1$ if the salesman t visited the city i, and $y_{ti} = 0$, otherwise.

In particular, Equation 4 ensures that each city must be visited by one salesman exactly once. Equations 5 and 6 indicate that all the salesmen's itineraries must begin and end at the same city. Finally, Equation 7 ensures that the number of cities traversed by each salesman cannot exceed a specific value.

4 Two Phase Heuristic Algorithm for MTSP

As discussed above, TPHA consists of a clustering algorithm, which aims to assign individual cities to m sets, and subsequently, a heuristic algorithm is implemented to plan a route for each city set. $K-$means is a very popular algorithm for partitioning a large dataset into multiple subsets, which is based on the Euclidean distance as the fitness function. This implies that the elements within a cluster are relatively concentrated and therefore meet the MTSP requirements. In this article, we combine $K-$means with the maximal capacity constraint to balance the number of cities belonging to the corresponding subsets. Despite a relatively large variety of clustering algorithms, which could be potentially applied in this context, $K-$means currently offers the most appropriate option. In fact, our proposed method focuses on city locations measured in terms of Euclidean distance, supporting our choice. In future research, we aim to comparatively assess the accuracy and feasibility of other clustering algorithms.
GA is renowned for its global search efficiency, and good scalability. In this work, we utilise the roulette wheel method and the elitist strategy as the selection operation of GA, where MOC is set to be the gene segment associated with the initial step. This operation can produce a new generation, even if the two parent chromosomes are identical. Furthermore, MOC can address the issue of local optima and premature convergence.

4.1 City Clustering Based on Improved $K-$means

In the original $K-$means algorithm, k cities are randomly selected as centres of the corresponding cluster, which subsequently identifies all nearby cities via the fitness function, whilst adjusting the centroid location accordingly. These steps are repeated until the convergence of algorithm is obtained. However, the original $K-$means algorithm cannot meet the objective of achieving a balanced number of cities.
Let $V = \{V_i : i = 1, \ldots, n\}$ be the set of n cities, and assume that the initial cluster set $s = (c_1, \ldots c_k)$, $\bar{v}_j$ has c_j as its centroid, where $\bar{v}_j = v_{i_j}, j = 1, \ldots, k$. Also assume that the number of cities in each cluster q_j is set to 0.
The $K-$means algorithm used in this work includes the following steps:

Step 1: set the capacity of each cluster, where $Q = \lceil \frac{n}{m} \rceil$ is the capacity constraint.

Step 2: calculate the distance of each city $v_i \in V$ to the cluster centre $\bar{v}_j$ with

$$||v_i - \bar{v}_j|| = \sqrt{(x_i - \bar{x}_j)^2 + (y_i - \bar{y}_j)^2}, i = 1, \ldots, n, \quad j = 1, \ldots, k, \quad (8)$$

and sort all the distance values $||v_i - \bar{v}_j||$ in ascending order.

Step 3: if there is a city $v_i \in V$ with minimum distance to the centroid c_j, then let $q_j = q_j + 1$, and then calculate whether the current number of cities in c_j is satisfied with $q_j \leq Q$. If it is, then v_i is assigned to c_i. Otherwise, let the distance value $||v_i - \bar{v}_j|| = \infty$ between the city v_i and the centroid. Repeat the steps until all cities are assigned.

Step 4: the coordinates of the centroid is updated via

$$\begin{cases} \bar{x}_j = \dfrac{1}{|c_j|} \displaystyle\sum_{v_i \in c_j} x_i \\ \bar{y}_j = \dfrac{1}{|c_j|} \displaystyle\sum_{v_i \in c_j} y_i \end{cases} \quad (9)$$

where $|c_j|$ is the number of cities in the cluster c_j. If the coordinates of the centroid are not changed, then the result is assumed to convergence, and move to Step 5; otherwise, go to Step 2.

Step 5: the clustering process is complete with the output cluster $s = (c_1, \ldots, c_k)$.

4.2 Route planning based on GA

The main GA parameters include parameter initialisation, initial population, evaluation of fitness function, the selection and the crossover operation, as well as the mutation operation. Table 1 lists some important parameters related to the algorithm.

Different GA approaches might involve different encoding, crossover and mutation operations, which may lead to a divergence of the iterative process. Therefore, it is necessary to redesign the above operations to ensure that the optimal solution is indeed attained.

4.3 Initial population encoding

In order to provide an integration of the definition of GA with the problem introduced above, it is necessary to identify a suitable encoding operator, which determines the evolution of the population. In any GA approach, a binary representation is generally applied to describe the corresponding target problems. However, according to the properties of TSP, each city can be associated with an integer, which represents a path $1, \ldots, n$, corresponding to a route scheme type. Therefore, solving TSP implies finding the shortest distance between any two numbers associated with two cities [25].

Table 1 Related parameters

Parameters	Description
G	Total number of iterations
P_{size}	Size of population
p_c	Crossover probability
p_m	Mutation probability
K	Number of iterations
X^k	All population of the $k-$th generation
X_i^k	The $i-th$ individual in X^k
D_i^k	The $i-th$ individual's evaluation value in X^k, $D_i^k = D(X_i^k)$
F_i^k	The $i-th$ individual's evaluation value in X^k, $F_i^k = F(X_i^k)$
$[sub\,x_1\ sub\,x_2] = CS(x_1, x_2)$	The new generation $sub\,x_1$ and $sub\,x_2$, which is obtained by the parents of x_1 and x_2 through the method of ameliorate order crossover
$U(0,1)$	Random function generating number subject to $(0,1)$ uniform distribution
X_{best}	The current best individual
D_{best}	The current best individual evaluation value

4.4 Fitness function

The fitness function is used to assess individual elements in the corresponding group. The selection operation based on the fitness value is one of the main steps in GA, and, to a large extent, it determines the performance of GA. In particular, a high value of the fitness evaluation implies that an individual has a high probability of being chosen. The fitness function $F(x)$ is defined as

$$F(x) = \frac{1}{D(x)} \tag{10}$$

where x represents an individual, and $D(x)$ is the distance travelled by the individual x. As can be seen from Equation 10, an individual x follwoing a shorter path will clearly have a high fitness value and will also have a high genetic probability to be selected for the next generation.

4.5 Selection strategy

The selection strategy is an important step in the evolutionary operation of GA, as it affects its efficiency. As discussed above, in this article we combine the roulette wheel method with the elitist strategy as the selection strategy of TPHA. More specifically, the former selects a chromosome in a statistical fashion based solely on its fitness value. On the other hand, the latter moves the individual with the best fitness at the current generation to the next one. Compared to the original roulette wheel method, the TPHA selection strategy allows to retain a "superior" individual. More specifically, we have the following steps:

Individual	Chromosome	Fitness value	Selection probability	Cumulative probability
1	36587142	6	0.200	0.200
2	53168247	5	0.166	0.366
3	87625413	2	0.066	0.432
4	73841625	10	0.333	0.765
5	26481537	7	0.233	1.0

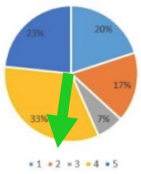


Fig. 2 Roulette selection strategy.

Step 1: assume P_{size} is the size of population, and X_i^k is the $i-th$ individual in X^k. Evaluate the $i-th$ individual's fitness value F_i^k in descending order. Subsequently, the individual with the best fitness at the current generation is moved to the next one.

Step 2: the probability of each individual to be selected is evaluated as

$$p_i = \frac{F_i^k}{\sum_{i=1}^{G} F_i^k} \tag{11}$$

where p_i is assigned to each individual X_i^k based on the order of calculation.

Step 3: the next generation is selected via the roulette wheel strategy by randomly generating a uniformly distributed number δ within $(0, 1)$. If

$$\sum_{j=0}^{i-1} p_j \leq \delta \leq \sum_{j=0}^{i} p_j,$$

then X_i^k is selected to the next generation. Repeat the above steps until all the parent chromosomes have been selected.

As shown in Figure 2, the greater the fitness value, the higher the probability of being selected will be.

4.6 Crossover operation

Two parent chromosomes P_1 and P_2 are selected according to the crossover probability p_c. This generates two intersection points, which identify the corresponding segments Δp_1 and Δp_2. Child 1 is then assigned to Δp_1 as the initial gene, whilst the equivalent components of P_2's chromosome are ignored. Finally, the remaining part is added to Child 1. Child 2 is defined in a similar manner. As an example, Figure 3, depicts the scenario with 8 cities, where the integers in $[0, 7]$ are associated with the two parents' chromosomes. The randomly selected Parent 1's gene segment (e.g. 5213) is used as the initial gene for Child 1, and the identical parts of Parent 2's chromosome are ignored. Finally, the remaining component is added to Child 1. The same procedure also applies to Child 2.

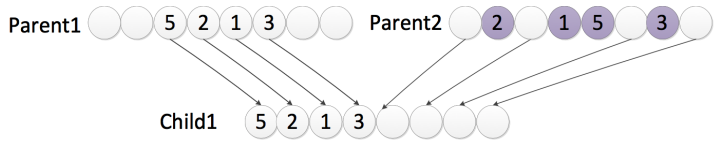


Fig. 3 Implementation process of ameliorate OC operator.

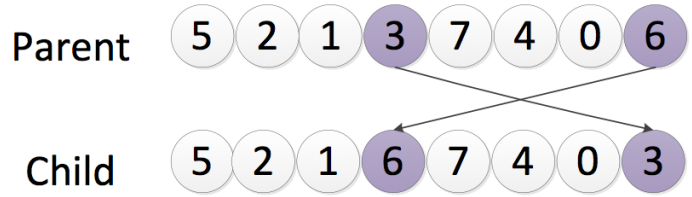


Fig. 4 The mutation operation.

4.7 Mutation operation

The mutation operation plays an important role in improving local search capability, whilst maintaining the variability of the population. It also prevents the premature GA convergence. This work utilises the swapping mutation, as follows: via the mutation probability p_m a chromosome is selected, and then two crossing points are randomly identified, whilst the selected points are exchanged. Figure 4 shows the case with 8 cities again, so that a route scheme is associated with the sequence of integers corresponding to cities $(5, 2, 1, 3, 7, 4, 0, 6)$. Two selected swapping gene points are node 3 and node 6, which are swapped to generate the offspring.
If the scheduled termination condition is satisfied (i.e. the number of iterations is larger than G), then the iteration is halted, and the corresponding path is considered satisfactory.

4.8 Detailed workflow

The detailed workflow includes the following steps:

Step 1: initialisation of the parameters, including G, P_{size}, p_c and p_m, and the generation of initial population X^1. We assume $x_{best} = X_1^1, D_{best} = D(X_1^1)$, $k = 1, j = 2, m = 1$, and $n = 1$.

Step 2: assessment of each individual's evaluation value for each generation, $D_i^k = D(X_i^k)$, where the fitness value $F_i^k = D(X_i^k)$, for $i = 1, \ldots, P_{size}$. For X_{min}^k, if $\min\limits_{x \in X^k} \{D(X)\} = D(X_{min}^k)$, then $D_{min} = D(X_{min})$, and $Y_1 = X_{min}^k$. Otherwise, D_{best} remains unchanged.

Step 3: evaluation of the probability of the corresponding individual to be selected for each generation via

$$p_i = \frac{F_i^k}{\sum_{i=1}^{G} F_i^k}$$

Step 4: if $j > P_{size}$, then move to Step 5. Otherwise, randomly generate $\delta \in U(0,1)$. If

$$\sum_{j=0}^{i-1} p_j \leq \delta \leq \sum_{j=0}^{i} p_j,$$

then X_i^k is selected for the next generation. Let $Y_j = X_i^k, j = j+1$, and repeat this step.

Step 5: the sequence A is obtained by the random number of integers $1, \ldots P_{size}$. The individuals in population Y are subsequently re-arranged according to A to obtain the population Z. Set $X^k = Y$.

Step 6: if $m > P_{size}$ then $m = 1$, and move to Step 8. Otherwise, randomly generate a number $r \in U(0,1)$, and move to Step 7.

Step 7: if $r < p_c$, then $[sub\, x_1\ sub\, x_2] = CS(x_m^k, x_{m+1}^k)$, and

$$S = [sub\, x_1\ sub\, x_2, X_m^k, X_{m+1}^k].$$

Select two chromosomes x_1 and x_2 with a smaller evaluation value via

$$min_{x \in S}\{D(x)\} = D(x_1)$$

and

$$min_{x \in S \setminus x_1}\{D(x)\} = D(x_2).$$

Let the next generation $X_m^{k+1} = x_1$ and $X_{m+1}^{k+1} = x_2$, and move to Step 6.

Step 8: if $n > P_{size}$, let $n = 1$ and move to Step 10. Otherwise, randomly generate a number $r \in U(0,1)$, and then move to Step 9.

Step 9: if $r < p_m$, then execute the mutation operation on X_n^k to obtain V_n^{k+1}, $X_n^{k+1} = V_n^{k+1}$. Let $n = n+1$ and then move to Step 8.

Step 10: set $k = k+1$. If $k < G$, then move to Step 2. Otherwise, terminate the algorithm, and output X_{best} and D_{best}.

The detailed flowchart of TPHA is shown in Figure 5.

5 Applications and Experiments

Tourism route planning is a well-known application of MTSP, where a tourist needs to traverse various scenic locations subject to minimising the travelled distance. Due to the constraint of a limited daily travel time, the number of locations needs to be balanced. Suppose there are n scenic locations, therefore a tourist will spend m days visiting all of them. Another important requirement

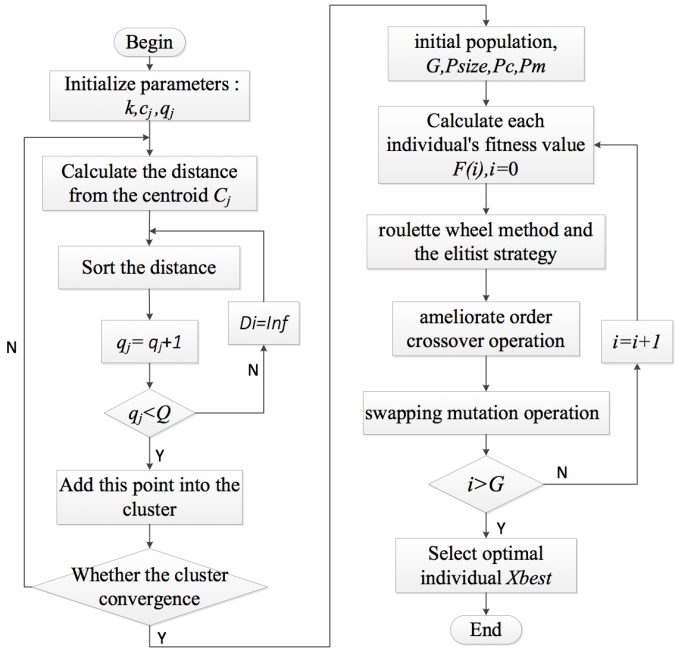


Fig. 5 Flow chart of the TPHA.

is minimising the time spent travelling between different locations. Furthermore, it is assumed that a tourist starts and completes his/her journey at the same hotel every day. In other words, we have the following constraints:

- The tourist will not change his/her hotel accommodation during his staying in a city.
- Each location is visited once and only once.
- The daily travel time is fixed, which limits the number of locations visited each day.
- The tourist will return to his/her hotel every day.

As part of the evaluation process, sixteen scenic areas in Nanjing city (China) were selected as shown on the Baidu map in Figure 6. The hotel accommodation was set as the Phoenix Universal Hotel. Table 2 shows a detailed description of the locations, and Figure 7 shows their corresponding latitude and longitude coordinates.

5.1 Experiments

First of all, by using TPHA, specific locations are assigned to different clusters based on their latitude and longitude coordinates, and let the number of days $m = k = 4$. The number of (daily visited) areas Q, $(Q = \lceil \frac{n}{m} \rceil)$ is eval-

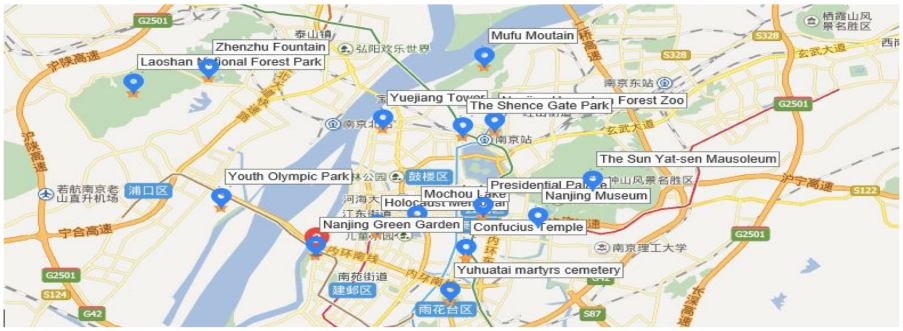


Fig. 6 Scenic spots.

Table 2 Scenic Locations

Serial number	Scenic locations	Abbreviation	Latitude and longitude coordinates
1	Laoshan National Forest Park	LSFP	(118.600362, 32.102836)
2	Zhenzhu Fountain	ZF	(118.665367, 32.128073)
3	Yuejiang Tower	YJT	(118.753228, 32.099638)
4	Youth Olympic Games Sports Park	YOGP	(118.671459, 32.055351)
5	Swallow rocky Park	SRP	(118.823312, 32.153159)
6	Mufu Mountain	MM	(118.784569, 32.121674)
7	Nanjing Hongshan Forest Zoo	HSZ	(118.808457, 32.102101)
8	The Shence Gate Park	SGP	(118.793415, 32.090744)
9	Yuhuatai martyrs cemetery	YMC	(118.785541, 32.009374)
10	Memorial Hall of the Victims in Nanjing Massacre by Japanese Invaders	MHVNM	(118.752705, 32.040775)
11	Mochou Lake	MCL	(118.765195, 32.043606)
12	Presidential Palace	PP	(118.803388, 32.049069)
13	Nanjing Museum	NM	(118.831876, 32.045068)
14	Sun Yat-sen Mausoleum	SYM	(118.859411, 32.064341)
15	Confucius Temple	CT	(118.795398, 32.026971)
16	Nanjing Green Garden	NGG	(118.795398, 32.026971)

uated, which determines the capacity constraints for each day. The original $K-$means algorithm randomly selects four locations as the initial centroid, it calculates the distance between them and the corresponding centroid, and finally if utilises the fitness function to arrange them to join the appropriate cluster, as shown in Figure 8. However, our improved $K-$means algorithm includes the capacity constraints of each cluster set, allowing the number of locations in each cluster to be more balanced, as shown in Figure 9.

In the second phase, TPHA uses the re-designed GA to plan routes for the four clusters. In order to test its performance, we downloaded the standard dataset from TSPLIB (`http://comopt.ifi.uniheidelberg.de/software/TSPLIB95/`).

The experimental results of the improved GA are subsequently compared with

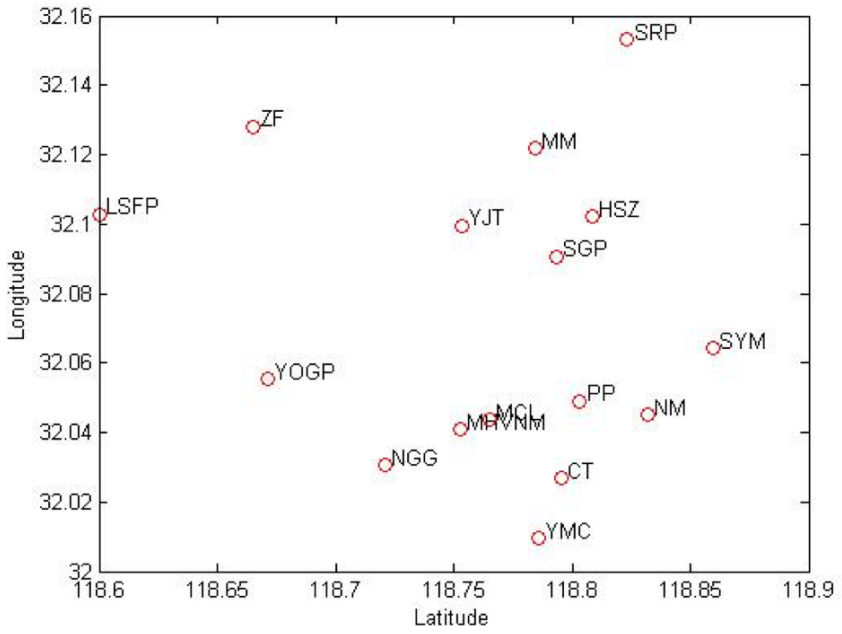


Fig. 7 Coordinates of the locations described in Table 2.

other typical algorithms. The related parameters are set via experiments as follows: $G = 2000$, $P_{size} = 500$, $p_c = 0.80$, and $p_c = 0.1$ The experimental results are shown in Figure 10. Figure 11 depicts the comparison of the GA experimental results with the Ant Colony Algorithm (ACA) [26] and the Nearest Neighbour Method (NN) [27]. OHC is the best known solution, and the deviation of the best found solution to the best known solution err is defined as follows:

$$err = \frac{w(Best) - w(OHC)}{w(OHC)} \tag{12}$$

Note that a small error rate indicates that a better solution of the method has been achieved.

As shown in Figure 11, the improved GA version clearly demonstrates that better results are obtained compared to the Nearest Neighbour Method. Although the error rate of ACA is lower than the improved GA, the time complexity of ACA is $O(Gn^2m)$, where G represents the total number of iterations, n is the number of cities, and m the number of ants. On the other hand, the time complexity of the improved GA is $O(GP_{size}$, which is lower. The integration of TPHA with the improved GA is utilised for the second phase of the route plan. The related parameters are set as follows: $G = 100$, $P_{size} = 100$, $p_c = 0.90$, and $p_m = 0.05$.

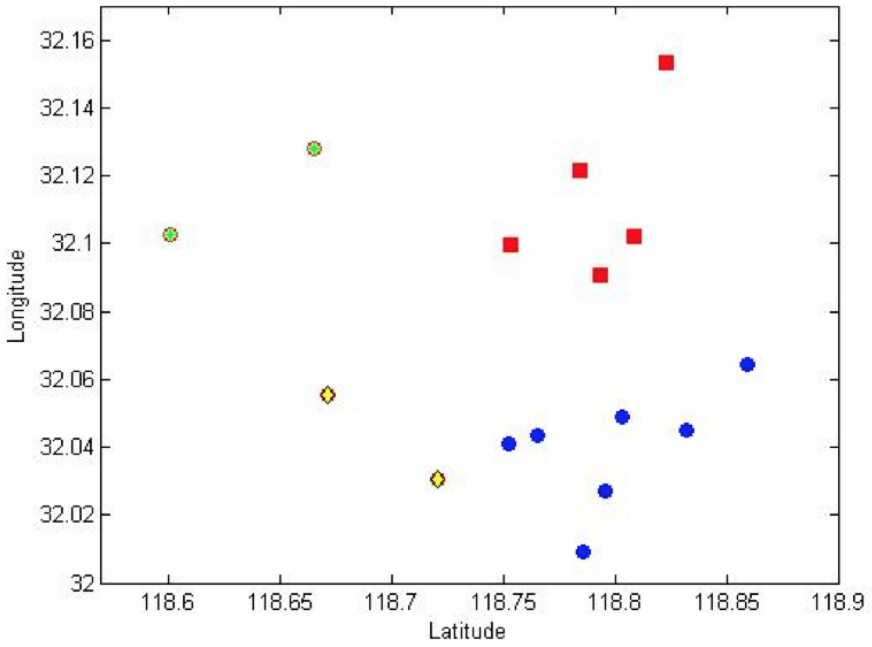


Fig. 8 Clustering of locations with the original $K-$means algorithm.

Table 3 Performance of TPHA and GA algorithms.

Algorithm	Time	Distance
TPHA	2.442360s	0.9941
GA	10.698919s	1.1648

Figures 13 and 14 show that with the original GA the length of chromosome is longer, not only increasing the computation time and reducing the convergence speed, but also increasing the total distance, as shown in Table 3.

5.2 Design of the Prototype

In this section, the design of a mobile guide system for tourists based on the Baidu electronic map SDK and Android 4.2 mobile platform is discussed.

As shown in Figure 14, the mobile guide system consists of the following classes:

- Class `MainActivity` describes the completion of the Baidu electronic map loading, including the verification of API key, the detection of network states and the management of map life cycle.
- Class `TPHA` invokes the class `MKsearchListener`, which obtains the location information, whilst GA carries out the path planning.

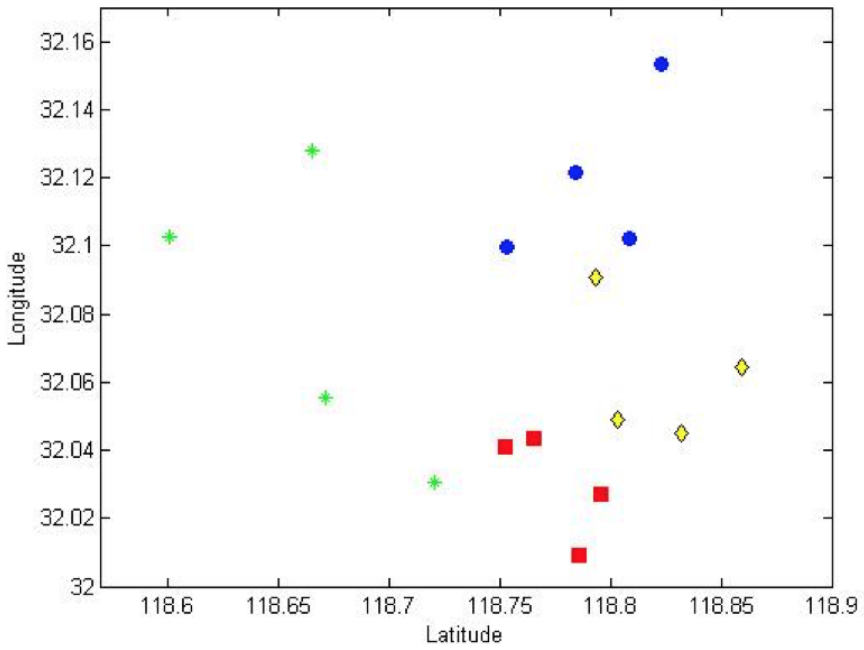


Fig. 9 Clustering of scenic spots with the improved $K-$means algorithm.

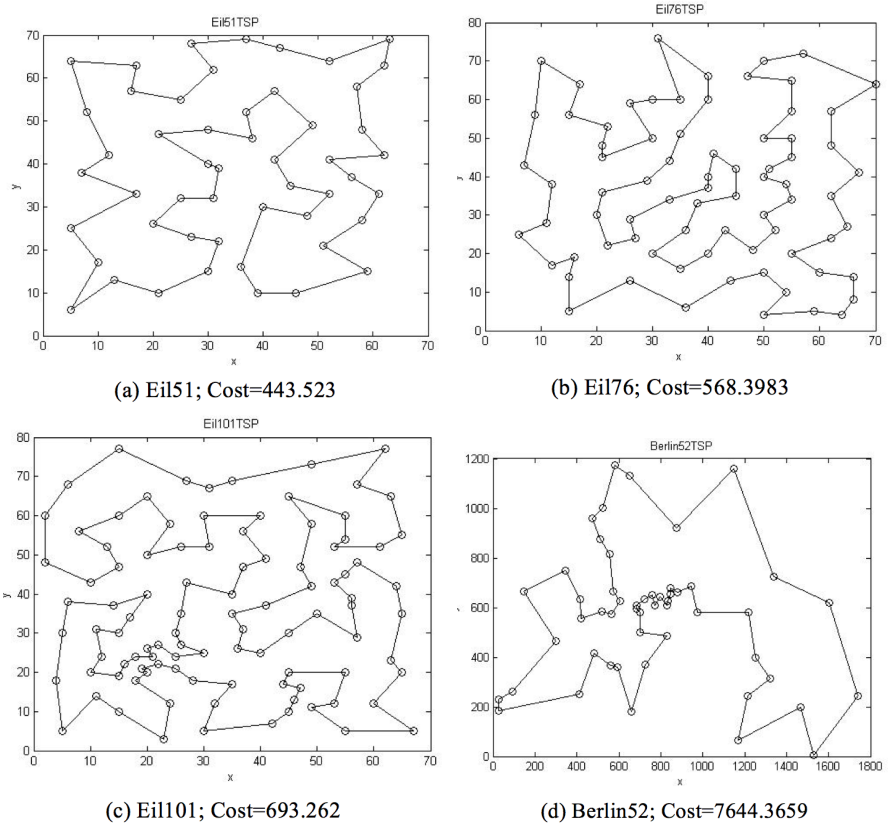
- Class `ItemizedOverlay` invokes the class `GpsActivity` to obtain the current location, and marks it on the map.
- Class `RouteOverlay` is used to identify the route
- `GraphicsOverlay` is utilised to show the path information on the Baidu electronic map.
- Finally, `getDistance()` is used to obtain the associated cost.

Figures 15 and 16 show the four-days travel path planned with TPHA and GA, respectively, demonstrating that TPHA has a better performance.

As shown in Figure 17, the distance of total route planned with TPHA is 150.7 km, while the distance of total route planned with GA is 159.4 km.

6 Conclusions

In this article, we have discussed the significance of MTSP, both from a theoretical and real-world applications point of view. In order to address the issues raised by MTSP, we propose TPHA, which integrates improved $K-$means and the re-designed GA to obtain the balanced and short-distance routes. Furthermore, this work specifically focuses on the workload balance subject to minimising the overall salesmen's travelling distance. Experimental results show that the proposed TPHA has a better performance in solving MTSP



(a) Eil51; Cost=443.523

(b) Eil76; Cost=568.3983

(c) Eil101; Cost=693.262

(d) Berlin52; Cost=7644.3659

Fig. 10 Best routes with the improved GA: (a) Eil51, (b) Eil76, (c), Eil101, and (d) Berlin52.

TSP instance	OHC	ACA		NNM		The improved GA	
		Best	err	Best	err	Best	err
Eil51	426	435	2.11	453	6.34	443	3.74
Eil76	538	551	2.41	582	8.18	568	5.57
Eil101	629	682	8.42	715	13.6	693	9.65
Berlin52	7542	7543	0.01	7976	5.57	7644	1.35

Fig. 11 Comparison of the improved GA and other algorithms.

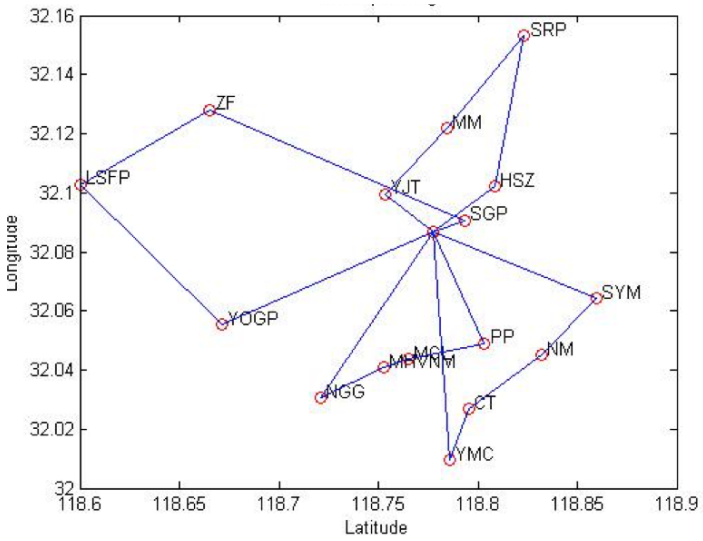


Fig. 12 Routes planned for the 4 clusters with the original GA

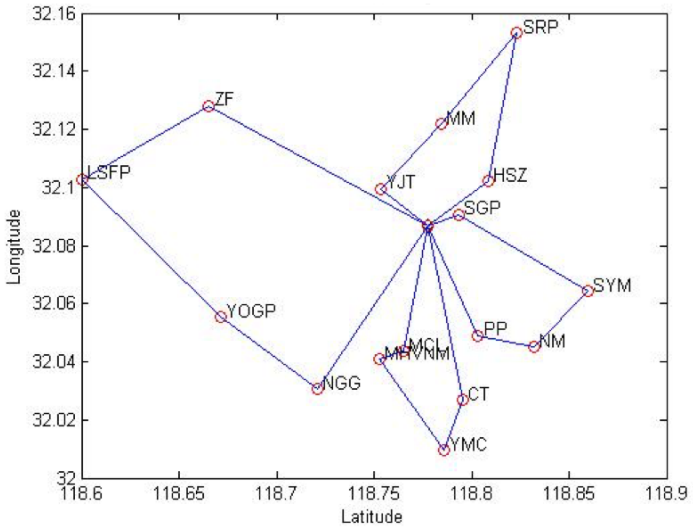


Fig. 13 Routes planned for the 4 clusters with TPHA.

with lower system overheads, and a mobile guide system for tourists was implemented to further demonstrate this.

Future research efforts will include the investigation of the time cost, the traffic and other constraints in order to model more complex and dynamic MTSP, whilst improving the performance and function of the route plan-

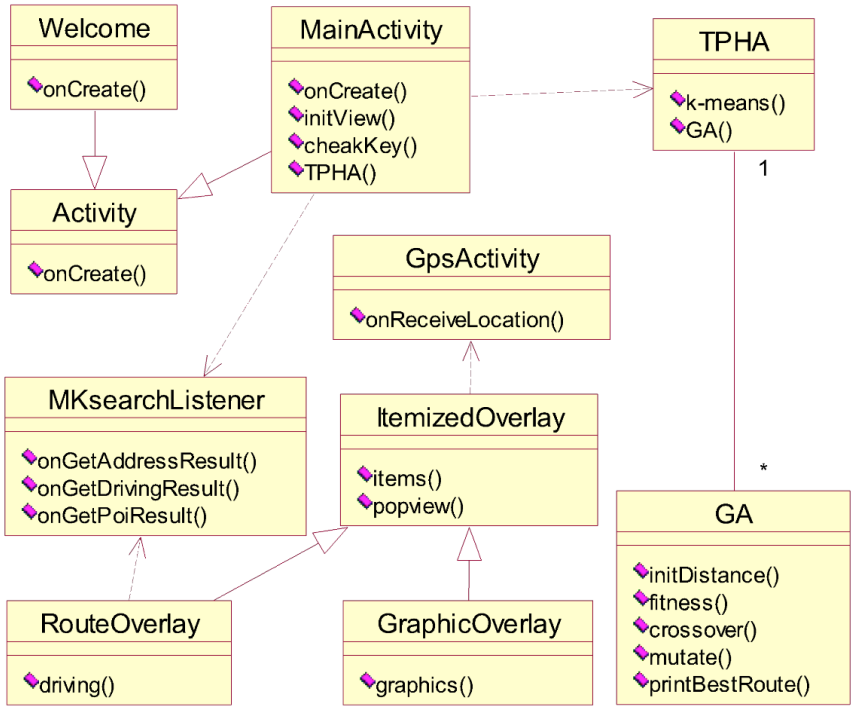


Fig. 14 Classes of our mobile guide system.

ning algorithm. Furthermore, we are aiming to consider different locations by analysing the corresponding datasets whenever available. This will be also facilitated by integrating Text Mining techniques to extract information from social media, blogs and general websites to validate the accuracy of the identified routes.

Acknowledgements We would like to thank the reviewers in advance for their comments to help us improve the quality of this paper. This work was jointly sponsored by the National Natural Science Foundation of China under Grant 61472192 and the Scientific and Technological Support Project (Society) of Jiangsu Province under Grant BE2016776.

Compliance with Ethical Standards

Funding.

This study was jointly sponsored by the National Natural Science Foundation of China under Grant 61472192 and the Scientific and Technological Support Project (Society) of Jiangsu Province under Grant BE2016776.

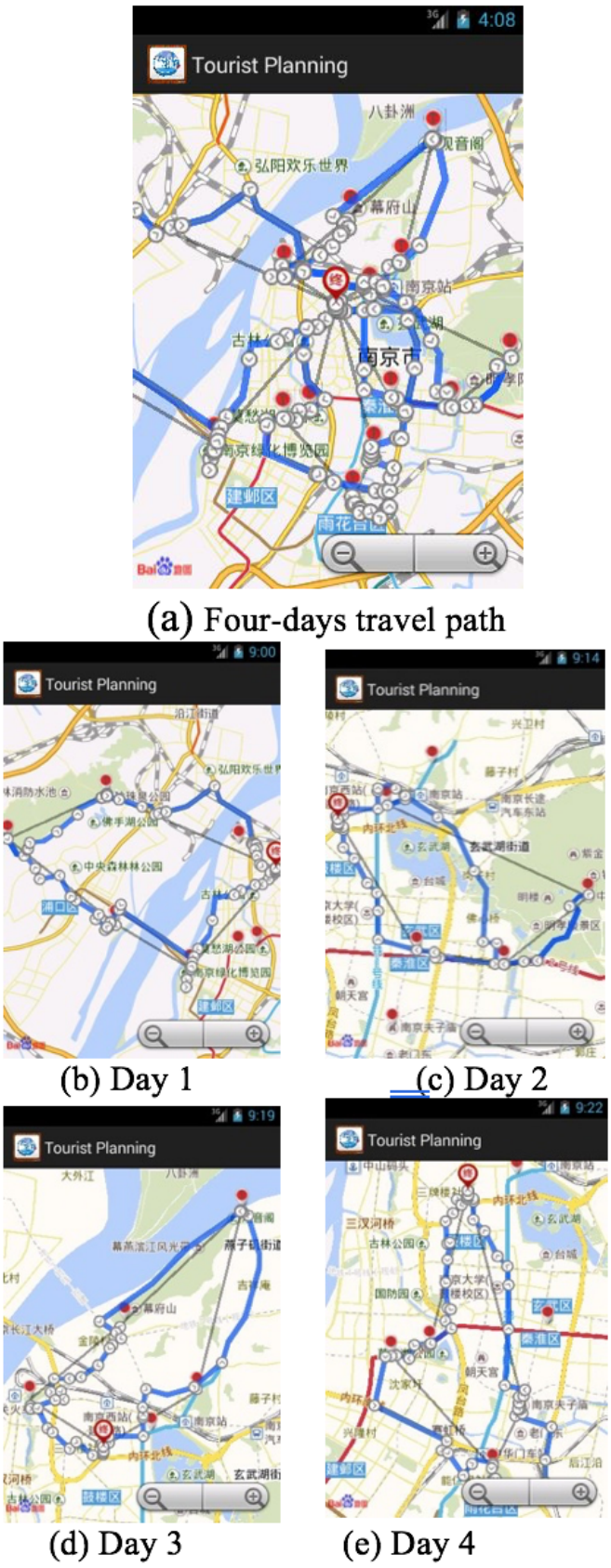


Fig. 15 Route planning with TPHA.

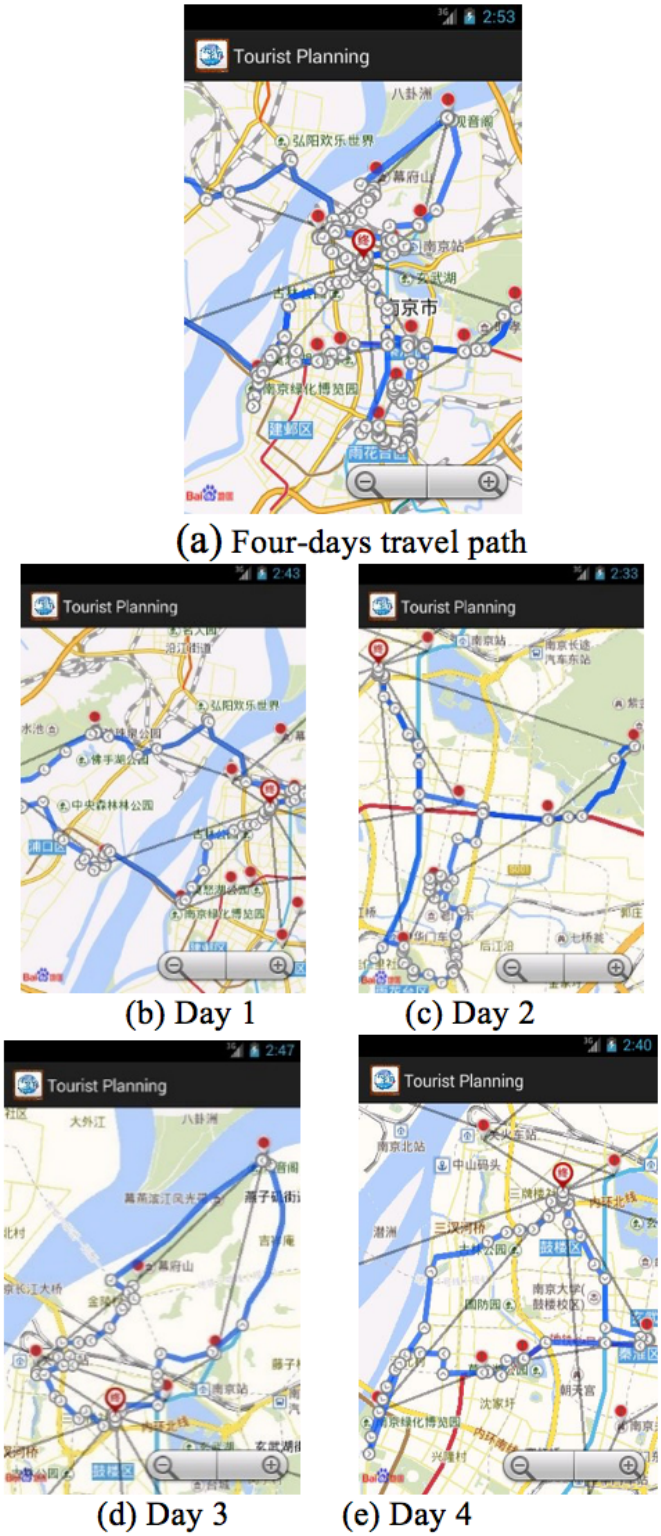


Fig. 16 GA scenic route plan.

With TPHA		
Route	Scenic spots	Distance
Day 1	Starting point, NGG, YOGP, LSFP, ZF	58.4km
Day 2	Starting point, PP, NM, SYM, SGP	30.2km
Day 3	Starting point, HSZ, SRP, MM, YJT	27.4km
Day 4	Starting point, CT, YMC, MHVNM, MCL	34.7km
With GA		
Route	Scenic spots	Distance
Day 1	Starting point, SGP, ZF, LSFP, YOGP	60.3km
Day 2	Starting point, SYM, NM, CT, YMC	42.2km
Day 3	Starting point, HSZ, SRP, MM, YJT	27.4km
Day 4	Starting point, PP, MCL, MHVNM, NGG	29.5km

Fig. 17 Detailed travel itinerary.

Conflict of Interest

- Xiaolong Xu declares that he has no conflict of interest
- Hao Yuan declares that he has no conflict of interest
- Mark Liptrott declares that he has no conflict of interest
- Marcello Trovati declares that he has no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. M.R Garey, D.S Johnson, Computers and intractability: A guide to the theory of NP-completeness, in Computers and Intractability, vol. 24,New York, 1979. pp. 90-91
2. M. Masmoudi and R. Mellouli, MILP for Synchronized-mTSPTW: Application to Home HealthCare Scheduling, in Proc. CoDIT, Metz, 2014, pp. 297-302
3. W. Ghadiry, J. Habibi and A.G. Aghdam, Generalized Formulation for Trajectory Optimization in Patrolling Problems, in Proc. CCECE, Halifax, NS, 2015, pp. 231-236
4. M. Liu and P. Y. Zhang, New Hybrid Genetic Algorithm for Solving the Multiple Traveling Saleman Problem: An Example of Distribution of Emergence Materials, Journal of Systems and Management, vol. 23, no. 02, pp.247-254, Mar. 2014.
5. H. An and L. Wei, Synthetically improved genetic algorithm on the traveling salesman problem in material transportation, in Proc. EMEIT, Harbin, Heilongjiang, 2011, pp. 3386-3371
6. A. Kiraly and J. Abonyi, Optimization of Multiple Traveling Salesmen Problem by a Novel Representation Based Genetic Algorithm, in International Computational in Engineering. Vol. 366, 2011, pp. 241-269
7. G. Singh and R. Mehta, Implementation of Travelling Salesman Problem Using ant Colony Optimization, Journal of Engineering Research and Applications, vol. 6, no. 3, pp. 385-389, Jun. 2014

8. X. S. Yan, C. Zhang, W. J. Luo, et al., Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm, International Journal of Computer Science Issues, vol. 9, no. 6, pp. 264-271, Nov. 2012.
9. R. Avin, D. Agin and M. Adnan, Solving TSP using Genetic Algorithms - Case of Kosovo, Advances in Computer Science, pp. 256-260, Nov. 2012.
10. R. Falcon and A. Nayak, The One-Commodity Traveling Salesman Problem with Selective Pickup and Delivery:an Ant Colony Approach, in Proc. CEC, Barcelona, Spain, 2010, pp. 4326-4333
11. Z. Y. Li, L. Ma and H. Z. Zhang, Discrete bat algorithm for solving minimum ratio traveling salesman problem, Application Research of Computers, vol. 32, no. 2, pp. 356-359, Feb. 2015.
12. J. Li, An Improved Dynamic Programming Algorithm for Bitonic TSP, in Proc. ISCCCA-13, Paris, France, 2013, pp. 24-27
13. S. S. Mahfoudh, W. Khaznaji and M. Bellalouna, A branch and bound algorithm for the porbabilistic traveling salesman problem, in Proc. SNPD, Takamatsu, 2015, pp. 1-6
14. K. P. Ghadle and Y. M. Muley, An Application of Assignment Problem in Traveling Salesman Problem (TSP), Journal of Engineering Research and Applications, vol. 4, no. 1, pp. 169-172, Jan. 2014
15. D. T. Pham and T. T. B. Huynh, New Mechanism of Combination Crossover Operators in Genetic Algorithm for Solving the Traveling Salesman Problem, in Knowledge and Systems Engineering, 2nd ed., vol. 326, Switzerland, 2015, pp. 367-379
16. C. Ye, Z. C. Yang and T. X. Yan, An Efficient and Scalable Algorithm for the Traveling Salesman Problem, in Proc. ICSESS, Beijing, 2014, pp. 335-339
17. A. K. Atif, U. K. Muhammad and I. Muneeb, Multilevel Graph Partitioning Scheme To Solve Traveling Salesman Problem, in Proc. ITNG, Las Vegas, NV, 2012, pp. 458-463
18. C. Q. Hu, A K-means algorithm, Journal of Changchun University of Technology, vol. 35, no. 2, pp. 139-142, Apr. 2014
19. K. Helsgun, Solving the Equality Generalized Traveling Salesman Problem Using the Lin-Kernighan-Helsgaun Algorithm, Mathematical Programming Computation, vol. 7, no. 3, pp. 269-287, Aug. 2014
20. S. Gorenstein, Printing press scheduling for multi -edition periodicals, Management Science, vol. 16, no. 6, pp. 373-383, Feb. 1970
21. S. Yuan, B. Skinner, S. Huang, D. Liu, A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms, European Journal of Operational Research, vol. 228, no. 1, pp. 72-82, Jul. 2013
22. R. Kaliaperumal, A. Ramalingam and J. Sripriya, A Modified Two Part Chromosome Crossover for Solving MTSP Using Genetic Algorithms, in Proc. ICARCSET 2015, New York, 2015, pp. 1-4.
23. A. A. R. Hosseinabadi, M. Kardgar, M. Shojafar, et al., GELS-GA: Hybrid metaheuristic algorithm for solving Multiple Travelling Salesman Problem, in Proc. ISDA, Okinawa, 2014, pp. 76-81.
24. R. M. F. Alves, C. R. Lopes, Using Genetic Algorithms to minimize the distance and balance the routes for the multiple Traveling Salesman Problem, in Proc. CEC, Sendai, 2015, pp. 1-8
25. Chen P. An improved genetic algorithm for solving the Traveling Salesman Problem, in Proc. ICNC, Shenyang, 2013, pp. 397-401
26. Liu Y, Shen X, Chen H. An adaptive ant colony algorithm based on common information for solving the Traveling Salesman Problem, in Proc. ICSAI, Shandong 2012, pp.763-766.
27. Wang Y. A Nearest Neighbor Method with a Frequency Graph for Traveling Salesman Problem, in Proc. IHMSC. Hangzhou, 2014, pp.335-338.