Optimization for Vehicle Routing Based on Taxi Pick-up Models

LING LIU, YIXIAO LU and CHENYANG XU

ABSTRACT

With the increase of traffic congestion, public transport is gradually rising and energy saving and environmental protection is more and more advocated. According to the taxi pick-up model, the paper will optimize the path of a taxi running and analysis to figure out that the cost of the total useless cost depends on the empty running cost of the taxi for a certain period of time. On this basis, the original taxi pick-up problem is transformed into a classic TSP problem. The corresponding mathematical model is established and an improved adaptive genetic algorithm is designed.

KEYWORDS

taxi pick-up, vehicle path, genetic algorithm

INTRODUCTION

With the development of social economy, the rapid growth of the number of motor vehicles, resulting in increasingly congested urban traffic, frequent traffic accidents, road and vehicle conflicts deteriorate and seriously affecting the public travel. It has become a hot spot for domestic and foreign scholars that how to plan a reasonable driving path of the complex urban road network. Among them, in Shanghai, the large city in China, has begun to control the number of motor vehicles, for that a large number of people began to rely on public transport. It has a significant significance of the people's livelihood and the effect of energy-saving and emission reduction to make researches of taxi research vehicle planning with the use of more and more taxis. In recent years, with the GPS and Beidou and other positioning equipment used and popularized in the taxi, the location data of the taxi can be achieved steadily [1]. The general taxi data contains vehicle information, pick-up time, drop-off time, travel distance, the pick-up latitude and longitude and the drop-off latitude and longitude [2,3].
**Problem Description**

Node $N(i)$ is the taxi pick-up location, that the node $N(i + n)$ is the passengers drop-off place in the index $i$. The taxi starting places is 0. This distinguishes the starting point, the picking point and the location where the passengers get off. All passengers on and off the location $N = \{0,1,...,n,...2n\}$ is a node set. $p = \{1,2,...,n\}$ is an another node set for the passenger on the taxi and $D = \{n + 1, n + 2,...,2n\}$ is for the passenger off the taxi. It is worth noting that different points may have the same physical location, that is to say different passengers may get on or off at the same location. The vehicle route problem with time window has a hard time window $[e_i,l_i]$ at each task point [4,5,6]. However, it is possible to ignore the influence of the time window because the time for passengers on or off is too small compared to the time between two points. The cost and time of taxi driving between two points are $c_{ij}, t_{ij}$, $q_i$ is the number of passengers requested by the customer $i$ from the node $N(i)$ to the node $N(n + i)$, the middle process is not allowed to receive other passengers. $d_{ij}$ is distance between any two points [7].

**Concepts and Assumptions**

1. Only consider the situation of a single taxi
2. In a certain period of time there is a demand for passengers waiting for this taxi, the driver will only send the next batch of passengers after sending passengers to the destination
3. The cost and time of travel are proportional to the distance.

**Strategies and models**

According to the above concepts and assumptions, when the taxi received the passengers, then the next point of the taxi driving must be the passenger's destination point. Therefore, for a taxi, the total loss will depend on the vehicle's empty running costs. So the node $N(i)$ and its destination node $N(i + n)$ can be combined to form a new node $N'(i)$, as shown in the figure to build a new model, the transformation of the problem is a typical TSP problem [8,9].

![Figure 1. Picture of conversion.](image-url)
In the new directed graph after conversion \( G' = \{N', D'\} \), \( N' = \{0, 1, 2, ..., n\} \), node 0 is still expressed as the starting position of the taxi, and the other nodes are transformed into the demand node in the TSP. \( D' \) is a distance between any two connection points. Above all, the classic TSP model can be solved by the following method.

Decision variables: \( x_{ijk} = \begin{cases} 1 \text{ taxi } k \text{ from } i \text{ to } j, \\ 0 \text{ otherwise} \end{cases} \)

\[
y_{ik} = \begin{cases} 1 \text{ the task in } i \text{ is completed by taxi } k, \\ 0 \text{ otherwise} \end{cases}
\]

\[
\min f = \sum_{i \in N} \sum_{j \in N} c_{ij} \sum_{k=1}^{K} x_{ijk} + \lambda \sum_{k=1}^{K} y_{0k} \quad (1)
\]

\[
\sum_{k=1}^{K} y_{ik} = 1 \quad \forall i \in N[0] \quad (2)
\]

\[
\sum_{i \in N} x_{ijk} = \sum_{j \in N} x_{ijk} = y_{ik} \quad \forall i \in N, \forall k \quad (3)
\]

\[
\sum_{i \in S} \sum_{j \in S} x_{ijk} < |S|, \quad \forall S \subseteq N[0], |S| \geq 2, \forall k \quad (4)
\]

\[
x_{ijk}, y_{ik} = 0 \text{ or } 1, \forall i, j, k \quad (5)
\]

The objective function (1) is to make the total cost to the smallest; the constraint condition (2), (3) indicates that each passenger point is accessible and can only be accessed once. The constraint (4) prevents the sub-loop between the customers.

**ALGORITHM DESIGN**

Because put the point that passenger on the taxi and its off point into a task point to build a model, the need to do some changes for the original transport network node distance to renew a new taxi distance matrix.

**Task Distance Matrix Generation Algorithm [10]**

**Input:** the original taxi network status parameter.

**Output:** transformed transport mission network \( G' = \{N', D'\} \).

**Start:**

Determine the location of each passenger into or off the taxi to form a pick-up-passenger property table; number the task of picking up; create an empty matrix \( M \).

**Start 1:**

For each picking-up task, calculate the distance between the drop-off location of the task and the other pick-up locations, and record the results to the corresponding rows of the matrix \( M \) by numbers.

**Return to 1:**

Output the task distance matrix \( M \).

**End**
After obtaining the taxi pick-up distance matrix, the genetic algorithm is used to solve the optimal completion order of the guest task. Genetic algorithm is a random search algorithm with natural selection and natural genetic mechanism in biological evolution process, which has stronger adaptability to solving complex and nonlinear optimization problems than traditional search algorithms [11,12].

**Optimal Completion Sequence Generation Algorithm of the Pick-up Task**

Input: task distance matrix genetic algorithm control factors GEN_MAX, POPU, etc., other calculation factors such as: vehicle driving costs and so on.  
Output: the completed path of the pick-up  
Start:  
Determine the coding scheme,  
Generate the initial population.  
When the number of iterations gen < GEN_MAX:  
Start 1:  
When the population m < POPU:  
Start 2  
Calculate the population fitness value,  
Choose operation,  
Cross operation,  
Mutation operation,  
Evolutionary reversal operation.  
Return to 2:  
The judgment of the algorithm terminates.  
Return to 1:  
Draw the iterative schematic, the objective function change graph, and the path graph.  
End

**Algorithm Key Parameter Designs**

(1) Choose operation  
Choose operation is that select good individuals out with a certain probability from the original group to form a new population to get the next generation of individual by breeding. The higher the fitness value of an individual, the higher the probability of being chosen. The selection operator is the selection strategy for calculating the fitness value ratio, selection operator:

\[ p_i = \frac{F_i}{\sum_{i=1}^{N} F_i} \]  

Where: \( F_i \) is the fitness value for individual \( i \) and \( N \) is the number of individuals of the population.

(2) cross operation  
Cross operations is that randomly select two individuals from the population and make the exchange of the two chromosomes to inherit the excellent properties of the parent to the offspring, resulting in new outstanding individuals. As the cross operation uses real cross method.
(3) mutation operation
The mutation operation is mainly to maintain the diversity of the population. Perform a mutation operation by first randomly selecting an individual from the population and selecting a point in the individual gene for variation to produce a better individual.

(4) evolutionary reversal operation
After the selection, crossover and mutation, multiple unidirectional reversal operation is carried out to improve the local search ability of the genetic algorithm. After the reversal operation, only the improvement of the fitness value is accepted, otherwise the reversal is invalid. For example, in [1, 10] randomly generate two numbers 8 and 7 and determine the two positions in a chromosome, then exchange the positions.

2 6 9 | 7 1 8 | 10 4 5 3
After the reverse operation becomes:
2 6 9 | 8 1 7 | 10 4 5 3

(5) determine the termination conditions
Because the genetic algorithm is a random search algorithm, it is difficult to find a clear termination condition. The general solution is to set the number of fixed evolution iterations $G_{\text{max}} (G_{\text{max}} \in [100, 1000])$, or to view the current optimization value which did not change after the evolution of several generations as a termination condition of the algorithm, this paper chooses to set the number of fixed iterations as the termination condition of the algorithm.

SUMMARY

(1) The pick-up task is exclusive to the taxi, the analysis of the total transport costs will depend on the size of the taxi’s empty running costs.
(2) Put forward to a new strategy that puts the original taxi pick-up task model into a classic TSP problem;
(3) This paper designs an improved genetic algorithm to solve the corresponding optimization and can be more quickly calculate the preferred taxi driving path.

REFERENCES