Multi-objective optimization intelligent path planning for autonomous driving

T Z Ma, H Chen, K Li and M Peng

Hunnan Campus of Northeastern University, Shenyang City, Liaoning Province, China
Email: mtz982437365@163.com

Abstract. In order to better solve the path planning problem of autonomous driving, optimize vehicle scheduling, and minimize the total distance travelled by vehicles and the total number of vehicles, this paper proposes a multi-objective optimized intelligent path planning algorithm V-MOEA. Firstly, this paper briefly introduces the relevant problem background of autonomous driving, and introduces the model to solve the path planning problem under this background. Secondly this paper puts forward intelligent path planning algorithm for multi-objective optimization of V-MOEA, the algorithm is used to insert heuristic algorithm population structure, using the variable probability of individual optimization interchange local search method, tested the algorithm of solution and the optimal solution to the existing deviation is very small, and its efficiency and effectiveness is better than there are a lot of algorithm. Finally, the adaptability, further improvement and application of the algorithm are discussed.

1. Introduction

1.1. Research background and significance

1.1.1. Brief Description of Autopilot Technology. Autopilot [1], also known as driverless or intelligent driving, refers to a ground wheeled vehicle that relies on intelligent algorithms and sensor technology to automatically sense the surrounding environment, automatically make motion decisions, and implement driving without human intervention. Autopilot combines a variety of cutting-edge intelligent technologies, such as artificial intelligence algorithm, Internet of things technology, microprocessor, to realize the perception of the vehicle driving environment, identify the road position and the information of driving obstacles, and make real-time judgment to achieve the control of the speed and direction of the vehicle and the driving route.

1.1.2. Development status at home and abroad. With the development of autonomous driving technology, many countries have carried out the research, in which the development of autonomous driving technology in western developed countries is rapid, and the development of automatic driving technology in China is relatively late. Nowadays, Google and Tesla in the United States have achieved great research results. Many companies such as Baidu and Momenta in China have made great achievements in the research and development of autonomous driving technology. In 2009, Google began the development of unmanned drones. In 2014, it launched a brakeless, steering-free autonomous unmanned vehicle. Tesla's practice in the market is more advanced than the results of
Google's technology research and development. In 2017, Baidu launched the “Apollo” program, and Baidu CEO Li Yanhong personally took the Baidu Apollo self-driving car from Baidu to the National Convention Centre.

1.2. Brief introduction of route planning technology in autonomous driving
As an important aspect of autonomous driving technology, route planning can be divided into global route planning and partial route planning. Common route planning steps are mainly divided into environment modeling, route searching, and route smoothing [3]. Environmental modeling abstracts the space we live in, and proper modeling of space is an important basis for route search; route search plans the route based on environmental modeling; route smoothing lies in the feasibility of implementing search route, providing a more reasonable solution.

The core of route planning technology lies in the application and research design of related algorithms [4]. Nowadays, more route planning algorithms are used, such as traditional algorithms, graphics methods and intelligent bionic algorithms. These algorithms have their own advantages. However, for complex global route planning, the above algorithm is not completely applicable because of many constraints on the solution. Therefore, this paper designs an intelligent route planning algorithm based on multi-objective optimization.

2. Multi-objective optimization model of route planning and its research

2.1. Route planning model for autopilot
This article is to solve the problem of route planning in autopilot. It is hoped to find a route planning solution for vehicles through a series of geographically dispersed customer points, that is, to minimize the number of vehicles and the distance travelled [5]. This type of route planning problem has a time window. Let the time window of customer i be \([a_i, b_i]\), \(a_i\) indicates the window opening time, and \(b_i\) indicates the window closing time. This paper explores the vehicle routing problem under hard time window constraints. The vehicle arriving early needs to wait for the customer to open the time window.

Let the vehicle set be \(K\) and the customer point set be \(V\). The journey from customer point i to customer point j is \(c_{ij}\). The time taken from customer point i to customer point j is \(t_{ij}\). The vehicle's stay time at customer point i is \(s_i\). The number of passengers on customer point i is \(p_i\) and the maximum number of people in vehicle k is \(Q_k\).

2.2. Objectives and constraints of vehicle route planning problem model
In this model, there are two decision variables, x and w, \(x_{ijk}\) represents whether the vehicle k can travel from the customer point i to the customer point j, and the \(w_{ik}\) represents the time when the vehicle k arrives at the customer point i. They meet the following const

\[
x_{ijk} = \begin{cases} 
1, & \text{the vehicle travels from the customer point i to the customer point j} \\
0, & \text{the vehicle does not drive from the customer point i to the customer point j} 
\end{cases} \\
a_i < w_{ik} < b_i
\]

We use k cars to serve all customer points, and we need to meet the following goals and constraints:

Goals:
1. Minimize the sum of the distance traveled by vehicles:
\[
\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} x_{ijk} c_{ij}
\]
2. Minimize the number of vehicles. 0 represents the starting point, not fixed:
\[
\min \sum_{k \in K} \sum_{j \in V} x_{0jk}
\]

Constraints:
1. Each customer point is visited and only visited once
\[
\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1 \quad \forall i \in V, \forall i \neq j
\]
2. Meet time window constraints when vehicle k goes from customer point i to customer point j
\[
a_j < w_{jk} = w_{ik} + s_i + t_{ij} < b_j \quad i \neq j, x_{ijk} = 1
\]
(5) Vehicles are not allowed to be overloaded

$$\sum_{i \in V} p_i \sum_{j \in V, j \neq i} x_{ijk} < Q_k \quad \forall k \in K$$

Since we are using a multi-objective optimization algorithm, we also need to meet the following constraints:

$$\begin{align*}
\text{Min} F(y) &= [f_1(y), f_2(y)] \\
h_i(y) &= 0, 0 \leq i \leq l \\
g_j(y) &\geq 0, 0 \leq j \leq k
\end{align*}$$

In this constraint, \(y = [x, w]\) is the decision vector, and \(h_i(x)\) and \(g_j(x)\) are equality constraints and inequality constraints, respectively. Finding the \(y\) that satisfies the constraints makes the objective function reach the optimal expectation.

3. Multi-objective optimization intelligent route planning algorithm

A two-stage hybrid algorithm that is widely used in existing path planning first looks for solutions with the smallest number of vehicles and then considers driving costs to a minimum. When the method based on weight and fitness value is adopted, the two values of vehicle number and distance are evaluated as a unified value. At this time, only one-peak space search must be performed for the path planning problem, and the obtained solution is often not optimal. Therefore, this paper regards the path planning problem as a multi-objective optimization problem.

3.1. Multi-objective optimization of route planning problems

In this paper, the two targets of the number of vehicles and the total distance are considered to be separated in the search space, and there is no need to introduce preferences in the search process to affect the performance of the search and the quality of the obtained solution, which is highly computationally efficient. At the same time, we can get a better solution.

3.2. Multi-objective algorithm Framework for vehicle routing problem (V-MOEA)

At present, most multi-objective genetic algorithms first construct the set of non-dominated solutions, then select some individuals in the set of non-dominated solutions for cross-mutation, and iterative optimization to obtain a new population. This algorithm is adapted to vehicle routing problem. Combined with the Solomon’s standard problem set, the termination condition is designed to end the loop when \(\text{minError} < 2.1\).

$$\text{minError} = 0.7 \times \frac{\text{minCarNum}}{\text{benchmarkCarNum}} + 1.3 \times \frac{\text{minDistance}}{\text{benchmarkDistance}}$$

\(\text{BenchmarkCarNum}\) and \(\text{benchmarkDistance}\) are the vehicle allocation numbers and the total travel distance of the known optimal solutions abroad[6].

---

Step 1: Read data from the dataset (time window constraints, load constraints, vehicle constraints, customer locations and requirements)

Step 2: Initialization population: FIHS algorithm is used to obtain an excellent feasible solution. The rest of the individuals are randomly generated and feasible.

Step 3: The fitness value is calculated for all individuals.

Step 4: Local search Optimization for population (Probabilistic Local Interchangeable Optimization)

Step 5: All individuals were selected, crossed and mutated to obtain subpopulations. Merge father and son population into new population.

Step 6: According to the new population, the multiple dominant solution set is constructed, and the next iteration population is obtained by truncation.

Step 7: Calculate \(\text{minError}\), if \(\text{minError} < 2.1\), exit this algorithm. Else go step 3
3.2.1. Chromosome representation and construction of initial population. In this paper, the encoding mechanism of natural number string is used to express chromosomes. Each gene bit in the chromosome represents a customer point, and the sequence of natural numbers between gene bits represents the order in which the vehicle reaches the customer point. In constructing the initial population, we first use the forward insertion method (FIHS) to generate a good feasible individual, and the remaining individuals are randomly generated.

- **Forward insertion heuristic search (FIHS)**

  In this paper, a forward heuristic search algorithm is used to obtain the solution [7]. In this method, the initial vehicle of each individual is obtained by a seed cost function. Traversing all unadded vehicles records the insertion cost of each access point at different insertion points. Select the route sequence with the least insertion cost until all access points are added.

  **Seed cost function**: \( \text{seedCost} = -0.3 \times \text{distance}(0,i) + 0.7 \times \text{dueTime}(i) \).

  Distance(0,i) represents the distance from the departure point to the customer point i, dueTime(i) represents the closing time of the customer point i. The lower the seed cost, the longer the seed customer is from the starting point and the earlier the due time is.

- **Insertion cost function**: \( \text{insertCost} = 0.3 \times \text{distance} + 0.7 \times \text{numCar} \).

- **Algorithm**: The design algorithm of FIHS

  Step 1: Customer with the lowest seed cost is supposed to be selected and considered as the seed customer, while the others are supposed to be removed.

  Step 2: FOR \( C_i \in \text{unselected customer} \), Find the best \( C_i \) insertion point for selected customers and record the corresponding insert cost.

  Step 3: IF unable to find the constraint compliant insertion point for the \( C_i \) customer, go step 1. Otherwise, the customer with the least cost will be inserted.

  Step 4: IF all customers are plugged in, exit. Otherwise, go to step 2.

- Feasibility process of individual

  Only one individual can be obtained using the FIHS algorithm. The remaining individuals are randomly generated. Traversing the customer points of random individuals one by one. If the customer point violates the time window constraint or load constraint, a new car is allocated. Since the objective function is the number of vehicles and the distance, the interference of exceeding the number of vehicles can be ignored.

3.2.2. Genetic operator design.

- **Selection operator**

  It is necessary to ensure the fairness of the selection operator and the iterative efficiency of the algorithm. This paper adopts the elite selection strategy and the binary random league strategy hybrid method to obtain the corresponding mating pool population [c1]. First, keep a small number of elite individuals into the mating pool. Then the remaining missing individuals apply equal probability to extract two individuals, and the two of the better individuals are added to the mating pool until the mating pool is selected.

- **Description of Hybrid Operator Design Algorithm**

  In this paper, the idea of designing the route hybrid operator is to comprehensively consider the two aspects of the number of vehicles and time, so as to select the optimal route and check the constraint satisfaction. The specific description of the algorithm is as follows:

- **Algorithm**: The design algorithm of hybrid operator

  Step1: \( \forall \text{ Individuality1 } \in \text{ Pop} \), repeat step 2 to step 8.
Step 2: Individual is crossed with a certain probability Rate. Get Front of Individual1, and Rate = $e^{-\left(1 - \text{Front}\right)/3}$.

Step 3: IF Rate×Random number between 0 and 1 Then go back to Step 1.

Step 4: Select Individual2 from Pop randomly.

Step 5: Select randomRoute1 from Individual1 randomly. The same, select randomRoute2 from Individual2. ∀ node ∈ randomRoute2 , delete the nodes from Individual1. ∀ node ∈ randomRoute1, delete the nodes from Individual2.

Step 6: Insert randomRoute2 into a random position of Individual1.

Step 7: Insert randomRoute1 into a random position of Individual2.

Step 8: Add Individual1 into newPop and add Individual2 into newPop.

Mutation operator design

In Mutation Operator, the number and species of route nodes is fixed, and the only change is the route and its combination order. Therefore, after comparing most kinds of mutation algorithms like local climbing mutation [8], we used the reverse mutation algorithm and adaptively adjusts the mutation probability of the individual in the same environment according to the front value of the individual.

Algorithm: The algorithm of Mutation operator

Step 1: ∀ (individual k) ∈ population, repeat step 2 to step 10:
Step 2: According to the value of “front” of $P_k$, adjusting the rate of mutation automatically;
Step 3: IF $P_k$ is going to mutate:
   Add $P_k$ to the community ML
Step 4: ∀ (individual i) ∈ ML, repeat step 5 to step 10:
Step 5: Divide $P_k$ into several routes, and select one route “R” from them randomly.
Step 6: Get the node sequence of route R and then reverse it;
Step 7: New sequence $R'$ takes the place of R;
Step 8: Get the new individual P.
Step 9: Get the new PARA_i corresponding to individual P.

3.2.3. Probabilistic Local Interchangeable Optimization. Today’s heuristic algorithms show good results in solving route planning problems, which depend to a large extent on the definition of the neighbourhood and the efficiency of the local search method used. Therefore, an efficient neighbourhood search method is extremely needed. The $\lambda$-interchange local search method [9] has proved to be an effective and efficient method in many problems. The optimization algorithm we designed is based on the idea of a few nodes exchange to obtain the neighbour solution in the $\lambda$-interchange local search method.

The idea of the optimization algorithm is to improve the quality of the solution by exchanging customer points in different vehicle routes of the same individual under different probability conditions. When $\lambda$ is assigned a value of 2, 8 operators are given: (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2). For each operator, customers are always considered sequentially along the route. The improved solution can only be saved if the replacement result results in a reduction in the total cost and still meets the constraints.

Algorithm: Probabilistic Local Interchangeable Optimization

Step 1: The frontier of the current population is obtained by NDSort(), and the leading edge parameters of each body are identified by FrontNO.
Step 2: REPEAT traverses each individual in the population and optimizes the individual
Step 3: Optimization algorithm initialization: If the current individual S is a non-dominated individual, set the local search probability to 1, otherwise set to 0.1, and randomly generate a probability value random_probability, such that 0<=random_probability <=1.

Step 4: IF (random_rate>rate) THEN GOTO Step9

Step 5: REPEAT Traversing the route of the current individual

Step 6: Calculate d according to the λ imported by the algorithm

Step 7: FOR i = 1: d

1. Find the current route location and calculate the current number of operators
2. Optimize the adjacent two routes and find the evaluation value of the generated individual, and get better individuals.

Step 8: UNTIL Complete local optimization of all routes

Step 9: Take the best optimized individual to replace the corresponding individual in the population

Step 10: UNTIL Complete traversal of all individuals in the population

3.2.4. Constructing multilayer nondominated set and truncation method. In this paper, the elite retention strategy is used to preserve the excellent individuals in the parent population, which can speed up the convergence of the algorithm. The first nondominant set constructed is the best individual, which dominates the other individuals in the combined population. The remaining individuals are considered as the next structure set, thus the population is divided into multi-layer non-dominant sets. At this point, the number of individuals in the multilayer non-dominant set exceeds the population size, so it is necessary to truncate it to construct a new generation of population: Firstly, the individuals of the first non-dominant set are selected into the new population to determine whether the number of individuals has reached the population size. If not, the next layer of non-dominant set is selected into the new population. Until the number of selected individuals reaches the upper limit of population size.

4. Test examples and the setting of Experimental Environment

4.1. Test examples and the setting of Experimental Environment

In this paper, the standard problem set of 100 customer points of Solomon is used as a test example to test the performance of V-MOEA in solving route planning problem. Solomon problem set can be divided into six categories: C1, C2, R1, R2, RC1 and RC2. Among them, class C is clustering data, that is, customer points are clustered according to geographical location or time window; The geographic location of customer points of class R is uniformly distributed; The RC class problem is between class C and class R, and it mixes the characteristics of the first two kinds of problems. In addition, for C1C R1 and RC1, the time window of distribution center is narrow and the load capacity of vehicle is small, so a vehicle can only serve a few customer points. The remaining three kinds of problems, the distribution center time window width, vehicle load capacity is large, a vehicle can serve multiple customer points.

Parameter settings: population size=100, crossover rate, Pc=0.5, variation rate, Pm=0.1.

4.2. experimental result

For each C1 class, C2 class, R1 class, R2 class, rc1 class and rc2 class each select a data set to test. The number of assigned vehicles, the shortest route, the deviation and the maximum number of iterations were recorded three times in each group. Take the average.

Table 1 shows the comparison between the results of each dataset algorithm and the known optimal solution, and the corresponding iterations. The customers of C102 have narrow time window and cluster distribution, while customers of C202 have wide time window and clustering distribution. The deviation of the two data sets (number of vehicles standard deviation / optimal vehicle number + route length standard deviation / optimal route length) is larger than that of other data sets, but the number
of iterations is shorter. For rc102 and rc202, this algorithm has the smallest deviation in the same class, but the number of iterations is opposite.

Table 1. Comparison of data set results

| data set | Min-number of vehicles | shortest distance | known optimal number of vehicles | Distance of known optimal solution | deviation | Max-number of iterations |
|----------|------------------------|-------------------|----------------------------------|-----------------------------------|----------|-------------------------|
| c102     | 10                     | 868.14            | 10                               | 828.94                            | 0.00755301 | 131                     |
| r102     | 17                     | 1518.12           | 17                               | 1486.12                           | 0.00380646 | 198                     |
| rc102    | 12                     | 1583.75           | 12                               | 1554.75                           | 0.00346369 | 194                     |
| c202     | 3                      | 616.56            | 3                                | 591.56                            | 0.00845223 | 145                     |
| r202     | 3                      | 1209.7            | 3                                | 1191.7                            | 0.00356016 | 311                     |
| rc202    | 3                      | 1383.89           | 3                                | 1367.09                           | 0.00299818 | 334                     |

Figure 1. shows the frontier (partial) distribution of the optimal solution obtained by the algorithms in each data set.
Figure 2. Measures the convergence rate of the algorithm under each data set according to minError.

4.3. Experimental analysis and conclusion
As far as the experimental results are concerned, the deviation between the proposed algorithm and the optimal solution is very small. When the customer presents uniform distribution (r class and rc class), the deviation is smaller than that of clustering distribution (class c). However, the convergence speed of the algorithm is not as fast as the clustering distribution.

5. Summary
With the rapid development of autonomous driving technology, the rapid and reasonable path planning solution has become an urgent problem to be solved. Therefore, an intelligent path planning algorithm with high efficiency and excellent results has great significance for optimizing the automatic driving technology. In this paper, the path planning problem in the context of autonomous driving is modeled, the idea of multi-objective is introduced when considering the merits of the evaluation path, and a multi-objective optimization intelligent path planning algorithm is constructed. Through the test and analysis, the proposed algorithm has little deviation from the known optimal solution, and the solution to the r-class problem and the rc-class problem is better than the c-class problem.

The innovation of this paper is that the multi-objective optimization solution is introduced into the path planning, and the influence of vehicle number and distance on the solution is taken into account equally. The efficiency and effect of the proposed solution in this paper are better than many existing algorithms, providing a new solution for the path planning problem of autonomous driving.

References
[1] M I Nasria, T Bektaş, and G Laporte. Route and speed optimization for autonomous trucks. *Computers & Operations Research* **100** 89-101
[2] Y H Lin, L C Huang, S Y Chen, and C M Yu. The optimal route planning for inspection task of autonomous underwater vehicle composed of MOPSO-based dynamic routing algorithm in currents. *Applied Ocean Research* **75** 178-192
[3] Markus, Thomas M B. The Internet of Vehicles or the Second. *ERCIM News* **77** 43-45
[4] J Q Wang, C W Wu, and X J Li. Research on Architecture and Key Technologies of Internet of Vehicles. *Microcomputer Information* **27** 156-158
[5] Z Wu, Y Zhang, Q Wu, and J Wu. Analysis of Autonomous Vehicle Steering System and Route Planning Method. *AISC* **752** 787-793
[6] K Deb, A Pratap, S Agarwal and T Meyarivan A fast and elitist multi-objective genetic algorithm: NSGA-II *Evolutionary Computation* 6 (2002) 182-197

[7] M Liu, J H Zheng and Y B Luo A New Multi-Objective Genetic Algorithm for Vehicle Routing Problem With Time Windows *Xiangtan University*

[8] GEN M and CHENG R Genetic Algorithms and Engineering Design[M] A Wiley-Interscience Publication 266-271.s

[9] I H Osman Meta-strategy simulated annealing and Tabu search algorithms for the vehicle routine problem *Annals of Operations Research* 41(4) 421-451