ISSN 2959-6157

Design and Implementation of a Community Waimai Delivery Assistant System Based on Path Optimization Algorithm

Shujia Zhang

Abstract:

With the rapid development of the food delivery industry, how to efficiently complete delivery tasks in complex community environments has become an important challenge for food delivery platforms and delivery personnel. This project focuses on the problem of planning delivery routes for b0oth inside and outside residential areas, and designs and implements a delivery assistant system based on the Internet of Things and path optimization algorithms. The system utilizes the DFRobot IoT platform to achieve real-time communication between Python programs and mobile apps. Through various path optimization algorithms such as genetic algorithm and simulated annealing algorithm, the system automatically generates the optimal delivery path, significantly improving delivery efficiency and reducing delivery costs. The experimental results show that the system has good operability and reliability in practical applications, providing an effective solution for improving the intelligence level of food delivery.

Keywords: path optimization, food delivery, genetic algorithm, simulated annealing algorithm, delivery path planning

1. Introduction

With the rapid development of Internet technology and the growing popularity of e-commerce, delivery service has become an indispensable part of modern urban life. Whether it's daily dining needs or sudden emergency purchases, food delivery services greatly facilitate people's lives. However, with the surge in order volume and the expansion of delivery areas, the efficiency issue of food delivery has become increasingly prominent. Especially in residential communities in cities, delivery personnel often need to shuttle frequently between multiple unit buildings. How to complete the delivery of all orders in the shortest possible time has become a major challenge faced by food delivery companies. This problem is essentially similar to the classic Traveling Salesman Problem (TSP), but it is more complex and diverse due to various limitations and requirements in practical scenarios.

Fig. 1 Image 1related problem scenarios

2. Technology Review

2.1 Greedy Algorithm

Strategically speaking, the greedy method is relatively simple in solving problems. It only makes judgments and choices based on the current information obtained, without considering future outcomes. Once this decision is made, it will not be changed because the greedy method does not consider the overall optimum. Therefore, its selection judgments are only local optima, and not all local selection judgments can always obtain the overall optimum. Relatively speaking, it can obtain approximate optimum solutions. For problems that can only be obtained through brute force exhaustive search, the greedy method is a good way to improve! 2.1.2 Sub heading

When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question.
When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. When receiving the paper, we assume that the corresponding authors grant us the copyright to use. [1]
Code implementation:
In the implementation process, first read the data of each coordinate point from the file, and calculate the Euclidean distance between all points to form a distance matrix. The algorithm starts from the starting point (usually the first point) and selects the nearest unvisited node as the next target each time. The final path connects all nodes together and returns to the starting point to form a closed loop.
advantage:
• Fast calculation speed, suitable for smaller scale problems.
• Simple to implement, easy to understand and expand.
Disadvantages:
• Due to selecting a local optimal solution each time, it is easy to fall into a local optimum and cannot guarantee a global optimal solution.
• For complex scenarios with a large number of nodes, the algorithm performance is poor.

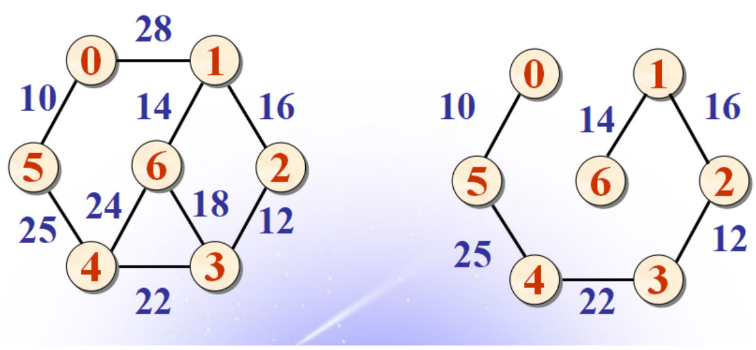

Fig. 2 Greedy Algorithm

2.2 Genetic Algorithm

Genetic algorithm (GA) is an artificial intelligence method that is based on the principle of survival of the fittest in Darwin's theory of biological evolution. It simulates the steps of biological evolution and incorporates concepts such as reproduction, hybridization, mutation, and competition into the algorithm. By maintaining a set of feasible solutions and recombining them, the movement trajectory or trend of feasible solutions in multidimensional space is improved, ultimately leading to the optimal solution. It overcomes the disadvantage of traditional optimization methods being prone to getting stuck in local optima and is a global optimization algorithm. [2]
Genetic algorithm is an optimization algorithm based on natural selection and genetic mechanisms. By performing selection, crossover, and mutation operations on the initial population, genetic algorithms can gradually evolve solutions with higher fitness, that is, solutions with shorter total path distances.
Code implementation:
During the implementation process, the initial population is first generated, with each individual representing a possible path. Then, the population is sorted based on the total distance of the path (i.e. fitness) to retain the optimal individual. Next, a new population is generated through crossover and mutation operations, and iterative evolution continues until the set number of iterations is reached or the fitness of the population no longer significantly improves.
advantage:
• Capable of effectively searching in a larger solution space and finding an approximate global optimal solution.
• Has strong adaptability to complex and multi constraint problems.
Disadvantages:
• Due to the inclusion of random operations, the algorithm results are unstable and may require multiple runs to take the average.
• Algorithm parameters such as population size, crossover rate, and mutation rate have a significant impact on the results and require careful tuning.

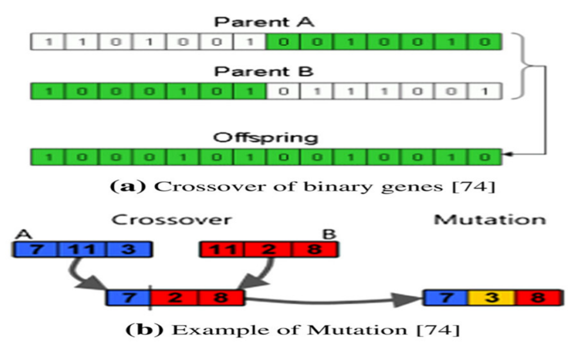

Fig. 3 Genetic Algorithm

2.3 Annealing Algorithm

Simulated Annealing (SA) is a method that simulates the cooling process of classical particle systems in thermodynamics to solve extreme values of planning problems. When the temperature of an isolated particle system decreases at a sufficiently slow rate, the system is approximately in thermodynamic equilibrium and eventually reaches its lowest energy state, the ground state, which is equivalent to the global minimum of the energy function. Due to its ability to effectively solve large-scale combinatorial optimization problems and minimal requirements for planning problems, simulated annealing has become a promising optimization method[3]

The simulated annealing algorithm gradually reduces the temperature of the system by simulating the physical annealing process, in order to find the global optimal solution. This algorithm can avoid getting stuck in local optima during the search process and is suitable for solving complex optimization problems.

Code implementation:

During the implementation process, the initial path is randomly generated and its total distance is calculated. The algorithm generates a new path by randomly swapping two nodes in the path, and accepts the new path if it is better; If the new path is poor, the solution is accepted with a certain probability, and the probability gradually decreases as the temperature decreases. Ultimately, the algorithm converges to a global or approximate optimal solution at low temperatures.

advantage:

• Can effectively escape from local optima and find global optima.

• Low dependence on initial solutions, suitable for complex path optimization problems.

Disadvantages:

• The convergence speed of the algorithm is slow and the computation time is long.

• Parameters such as initial temperature and cooling rate have a significant impact on the results and require careful tuning.

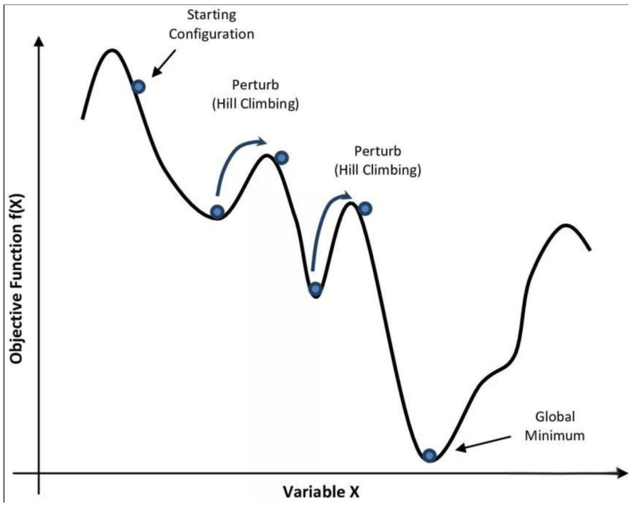

Fig. 4 Annealing Algorithm

3. Code Test

3.1 development environment

Python, Compared to other programming languages, Python is easier to get started with, easy to quickly get started with, and has a large number of third-party library support. It can easily and quickly implement various functions with a small amount of code, and Python language has good calling methods in fields such as artificial intelligence and web crawlers. In addition, the programming tool used in this project is the idle tool that comes with Python. It is installed together with Python and displays different colors for different built-in functions and data types, making it easy to use.

3.2 running results

Experimental objective:

To verify the effectiveness of different path optimization algorithms in practical scenarios.

Experimental steps:

1. Run the greedy algorithm, genetic algorithm, and simulated annealing algorithm separately in Python programs to generate the optimal path for the same building combination.
2. Record the path length and computation time generated by each algorithm.
3. Compare the efficiency and path optimization effects of various algorithms.

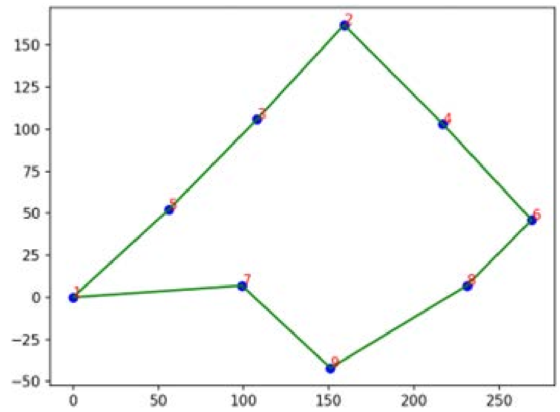

Fig. 5 Experimental results of Genetic Algorithm

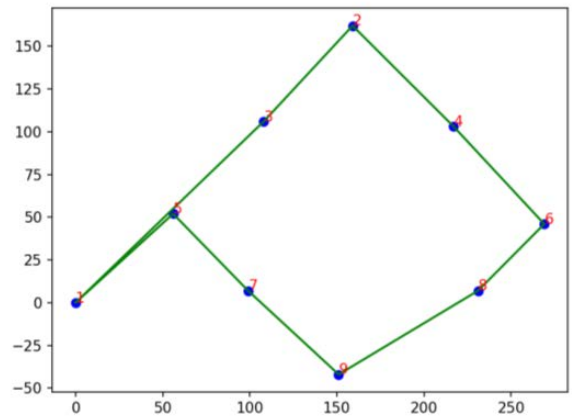

Fig. 6 Experimental results of Greedy Algorithm

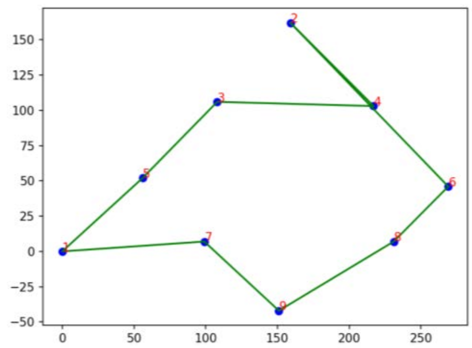

Fig. 7 Experimental results of Annealing Algorithm

Table 1. Program running time

Numble	Algorithm	Time
1	Genetic	0.015275s
2	Greedy	0.000031s
3	Annealing	0.0007s
4	exhaustive method	0.845904s

3.3 test summary

Genetic algorithm performs better in complex scenarios, generating shorter paths quickly and accurately, while greedy algorithm has the fastest computation speed and is suitable for situations with fewer nodes.

4. Current Problems and Future Directions

Although significant achievements have been made in optimizing sales and delivery routes within and outside the community, there are still some limitations and room for improvement in this project. In future research, we plan to conduct in-depth exploration in the following areas:

1. Further improvement of path optimization algorithm: Although the current system can generate better delivery paths, the efficiency and effectiveness of the algorithm may be affected when facing larger or more complex order distributions. In the future, we will explore the combination of more intelligent algorithms to further improve the computational efficiency and path optimization effect of the system.

2. Expansion and application promotion of the system: Currently, this system is mainly used for food delivery scenarios within residential areas. In the future, we can consider promoting the application of the system to other fields, such as express delivery, logistics scheduling, etc., to expand its application scope and market value.

5. Conclusion

This project focuses on the optimization of sales and delivery routes inside and outside the community, and designs and implements a complete set of delivery assistant systems based on the Internet of Things and path optimization algorithms.

The innovation of this project lies in the use of multiple path optimization algorithms (such as genetic algorithm, simulated annealing algorithm, etc.) in this system, which can automatically generate the optimal delivery path based on the actual building location and order situation, solving the problem of manual path planning for delivery personnel in complex environments.

The key and difficult point of this project lies in the selection and implementation of path optimization algorithms. In practical scenarios, different building distributions and order volumes have different requirements for path planning. Choosing the appropriate algorithm and conducting effective optimization is one of the core challenges of this project. Through multiple experiments and optimizations, a path optimization algorithm that can perform well in complex environments was ultimately achieved.

References

[1] Lai Xuewei. Application of Greedy Algorithm in TSP Problem [J]. Journal of Xuchang University, 2017, 36 (2): 4. DOI: 10.3969/j.issn.1671-9824.2017.02.009

[2] Wang Songquan, Cheng Jiaxing. Performance analysis of genetic algorithm and simulated annealing algorithm for solving TSP [J]. Computer Technology and Development, 2009, 19 (11): 4. DOI: CNKI: SUN: WJFZ. 0.2009-11-027

[3] Noy IN F,MeGuinness D L. Ontology development: a guidetoincreasing your first ontology[R]. USA: Srandford University,2001.

[4] Song Congwei Pyrimidine: Python Framework for Object Oriented Genetic Algorithm [J]. Computer Programming Techniques and Maintenance, 2021

[5] Ji Xinshui. Research on Delivery Path Optimization Algorithm Based on Simulated Annealing Algorithm [J] [2024-09-19].

[6] Xu Xin, Shen Bo. A Path Planning Method for TSP Problem: CN201910256712.2 [P] CN109948865A[2024-09-19].