Package Delivery via Drone

Radium Zhang
zhan5821@umn.edu

December 21, 2020

Abstract

Through reading many peer-reviewed papers, I have summarized the two models of how people use drones to deliver express delivery. They are the refueling point model and the vehicle-drone composite delivery model. In my proejct, I will use the second model and design the algorithm used in the second stage of drone delivery. The algorithms I have adopted and compared are the informed algorithm, the ant algorithm and the uninformed algorithm Dijkstra Algorithm. I used eight different scenarios to run these two algorithms. By comparing their running time and the relative best path they found, I came to the conclusion that the two algorithms have their own advantages and disadvantages in different application scenarios. Relatively speaking, the applicability of ant algorithm is better and wider.

Introduction

Now we often do online shopping. In the support of a large number of algorithms, machines gradually replace human-beings. My research problem is to use drones to deliver packages, which algorithm will be more efficient and save resources. First of all, we know the common vehicle routing problem: according to the parcels to be delivered (or goods to be collected) and geographically distributed, we can calculate the number of vehicles to be sent from the logistics center and what route to take to deliver (or collect) these packages respectively, so as to save time and effort. We often treat these problems as TSP problems, which widely used in transportation, circuit board circuit design and logistics distribution. Due to the large data scale of this kind of problem, it is often powerless to use an accurate algorithm to solve it. [1]Therefore, scholars focus on using heuristic algorithm to find the approximate optimal solutions, such as simulated annealing algorithm, genetic algorithm, ant colony algorithm and so on, as well as traditional greedy algorithm and dynamic programming. Therefore, in this paper I will build my algorithm to solve this problem – "Drone Package Delivery".

Existing Solution

Research on task allocation based on fuel replenishment point mode

The actual performance constraints such as the range and flight time of the current delivery drones have led to prominent problems such as small coverage and low efficiency. Some scholars have considered establishing suitable fuel replenishment points in the drone delivery network. As early as 2005, in order to improve the efficiency of vehicle delivery, [5]Kuby proposed to replace the traditional road gas station mode with mobile refueling methods. Under the condition of a certain number of mobile refueling points, the flow from the start point to the end point was maximized, so that vehicles could not Continue to run when running out of fuel. Unlike road vehicles operating on a fixed network, UAV delivery will change the location of the UAV's possible flight path, which will affect the acquisition of the optimal flight path. At the same time, the UAV's range performance will also be affected. Determination of the location of the fuel replenishment point. Sundar based on the mid-way replenishment fuel supply model to solve the problem of single drone support multiple demand points, which improved the continuous support capability of drones. On this basis, [5]Kim, Song, Morrison, etc. The model of refueling researched the coordination problem of multiple drone distribution tasks. [3]Boualem proposed the functional relationship between UAV's energy consumption and actual load and flight mode, and optimized the task allocation of UAV delivery of emergency supplies in disaster relief. [6]Insu optimized the location of the UAV fuel replenishment point and the delivery route at the same time, and established a mixed integer programming model covering positioning and task allocation.

Research on task allocation based on vehicle-UAV joint delivery model

In view of the limitation of the range of drones' performance, some scholars have considered the joint delivery of drones and traditional vehicles. The delivery vehicles can also be used as mobile drone bases. The advantages. [7]Murray defined this problem for the first time as the Flying Sidekick Traveling Salesman Problem (FSTSP), constructed a mixed integer linear programming model and designed a heuristic algorithm to solve it. The algorithm is based on the classic TSP problem, and iteratively compares all the task points in the TSP one by one to filter out the demand points suitable for UAV delivery. At the same time, it adjusts the previously generated vehicle routes and evaluates all the solutions until the best is generated. Program. However, due to the large number of iterations, cumbersome process, time-consuming calculation and low efficiency, it is only suitable for solving the situation where the number of demand points is small. [1]Agatz studied the Traveling Salesman Problem with Drone (TSP-D) involving drones. Compared with the FSTSP problem, drones in TSP-D can be launched and recovered at the same location, using partial The algorithm that combines search and dynamic programming to solve the task allocation model. The solution idea is to give priority to the vehicle transportation route, and then gradually compare and screen to determine the task points of the UAV. Therefore, the algorithm is not efficient and is not suitable for solving the situation of a large number of demand points . [3]Bouman et al. improved the above method and proposed an accurate algorithm based on dynamic programming, which can solve the task allocation problem when the number of demand points is large. [3]Ponza discussed the FSTSP problem in depth in his master's thesis, established an improved mixed integer linear programming model, and designed a heuristic algorithm based on simulated annealing to solve it. [10]Savuran set the goal of minimizing the combined delivery mileage of drones and delivery vehicles, using genetic algorithm to solve the problem, and comparing with the nearest neighbor algorithm, the results show that the genetic algorithm is more efficient. [4]Ha studied the traveling salesman problem with the participation of drones, established a task allocation model with the goal of minimizing the time and cost of the drone to complete the delivery task, and designed a heuristic algorithm to solve it. In addition, [11]Wang studied the problem of multi-vehicle and multi-UAV joint delivery. For the first time, this problem was defined as a VRP problem with drone participation (VRP-D). The individual delivery time of vehicles was compared, and a multi-vehicle and multi-UAV joint delivery model with the shortest delivery time as the goal was established. On this basis, [9]Poikonen considered the UAV's range constraints and integrated the delivery cost target. In the solution process, both the cost boundary and the time-consuming boundary were considered. [6]Mourelo built a drone-vehicle hybrid express delivery system to solve the problem of determining the launch position of drones during joint delivery and optimizing the number of drones carried by a single vehicle. In the field of commercial logistics, distribution cost is a key indicator for many companies. [1]Mathew and others are studying the Heterogeneous Delivery Problem with the goal of reducing transportation costs. For demand points that cannot be directly reached by distribution vehicles, [1]Mathew The end of the network uses vehicles to launch delivery drones for the "last mile" delivery, which decomposes the problem into a generalized traveling salesman problem for solution.

My Solution

I will take vehicle-UAV joint delivery model to establish my solution. In my solution, I assume all the packages that need to be delivered are already dropped in the package center. Each city has several package center, the drone will be able to pick up the package from the customer and drop it to the package center, or the drone will be scheduled to deliver the package from the package center. In my solution, we don't need to care about the path from one city to another, as I am trying to solve the shortest path via drone in the city wide.

Dijkstra Algorithm

[8]The Dijkstra algorithm adopts a greedy strategy, declaring an array dis to store the shortest distance from the source point to each vertex and a set of vertices that have found the shortest path: T. Initially, the path weight of the origin S is assigned to be 0 (dis[s] = 0). If there are edges (s, m) that can be directly reached for the vertex s, set dis[m] to w(s, m),

and set the path length of all other vertices (that S cannot directly be reached) to infinity. Initially, the set T has only vertex S.
Then, select the minimum value from the dis array, the value is the shortest path from the source point S to the vertex corresponding to the value, and the point is added to T. Then a vertex is completed.
Then, we need to see if the newly added vertices can reach other vertices and see if the length of the path through this vertex to other points is shorter than that of the source point directly. If so, then replace the value of these vertices in dis. Then, find the minimum value from dis and repeat the above action until T contains all the vertices of the graph.
[10]In my research topic, the ordinary Dijkstra Algorithm usually only studies: how to find the shortest path without considering other application costs and other environmental change factors. Therefore, Dijkstra Algorithm's idea of solving my problem is probably the same as that of solving common problems. [2]Also according to Bekhti's paper, his paper is basically studying different UAVs planning algorithm, which are similar to my working problem. And the Dijkstra Algorithm he built is similar to general problem solution. Hence there is no huge difference of solutions between my problem and general problem using Dijkstra Algorithm.

Ant Algorithm

[6]In my question, I want to design a method of path planning based on ant algorithm, including the following steps: (1) Establish a mathematical model of path optimization, as follows: Let G be a path from starting point 1 to ending point n, path G does not include repeated routes and patrol routes, and the cost of a route is the sum of the weights on this route (2) The path search process is as follows: starting from the starting point, using ant algorithm to search for an optimal path within the search radius, as the drone moves to the next node of the path obtained from the previous search, this node As the starting point of the current search, a path is searched again with nodes within the search radius, and this method is continuously implemented until the destination point is searched.

Approach Description

[8] A multi-drone cooperative path planning method based on ant colony algorithm, including the following steps:

Step 1

Analyze the drone flight environment and establish an environment modeling based on the Voronoi diagram.

(1-1) Determine the flying height of the UAV, intercept the two-dimensional plane terrain information of the flying height, and project the ground threat to the two-dimensional plane terrain information to obtain the ground threat plane terrain.

(1-2) Abstract the ground threat plane terrain and other threat sources as threat point sets $\{x_i\}$.

(1-3) Establish the coordinate system in the plane, obtain the point set of the threat source$\{(x_i, y_i)\}$, and generate the Voronoi diagram.

(1-4) Enter the starting point and end point of the drone, and hence the environment modeling of the Voronoi diagram is completed.

Step 2

Calculate the cost of edges in environment modeling based on Voronoi diagram.

(2-1)Calculate the cost of terrain threats to the edges.$P_{i-F}^{j} = Ke^{-kr_{ij}}$
P_{i-F}^{j} represents the cost of the fixed threat source j to the i-th edge; K is the threat level of the threat source j; k is the artificial agreement coefficient; r_{ij} is the distance from the threat source j to the i-th edge.

(2-2)Calculate the cost to the side of a threat with reconnaissance capability but no attack capability:
$P_{i-F}^{j} = L_i(\frac{1}{d^4_{\frac{1}{8},i,j}} + \frac{1}{d^4_{\frac{3}{8},i,j}} + \frac{1}{d^4_{\frac{5}{8},i,j}} + \frac{1}{d^4_{\frac{7}{8},i,j}})Q_j$
P_{i-R}^{j} is the cost of radar j for the i-th edge; L_i is the length of side i; $d^4_{\frac{1}{8},i,j}$ is the distance from $\frac{1}{8}$ of the i-th side to radar j; Q_j is the transmitting power of radar j.
According to the physics, the formula of Q_j is as follows: $P = \frac{P_t G A_e \delta}{(4\pi)^2 R^4}$.

(2-3)Calculate the cost of a threat with both reconnaissance and attack capabilities:
$P_{i_G}^{j} = B(1-a)p_{ij}$.
P_{iG}^{j} is the threat of missile j to the i-th path; B is the missile's attack capability; $(1-\alpha)$ is the missile hit rate; p_{ij} is the drone's probability of detection on the edge of the bar.

(2-4)Calculate the edge length cost:
$P_{i-L} = \lambda L_i$.
P_{i-L} is the cost of length on side i;λ is the coefficient;$L_i$is the length of the i-th edge.

(2-5)The formula of the total cost of edges:
$P_i = a\sum_{j=1}^{m} Ke^{-kr_0} + b\sum_{j=1}^{n}[L_i(\frac{1}{d^4_{\frac{1}{8},i,j}} + \frac{1}{d^4_{\frac{3}{8},i,j}} + \frac{1}{d^4_{\frac{5}{8},i,j}} + \frac{1}{d^4_{\frac{7}{8},i,j}})Q_j] + c\sum_{j=1}^{r}[B(1-\alpha)p_{ij}] + d\lambda L_i$.
a, b, c, d are constants, satisfying a+b+c+d=1; m is the number of fixed obstacles, n is the number of radars, and r is the number of missiles.

Step 3

Use ant algorithm to plan initial path for drones.

(3-1)Ants start from the initial node, according to the transition probability formula:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta}{\sum_{r \in allowed_j^k} [\tau_{ir}(t)]^\alpha * [\eta_{ir}(t)]^\beta} & j \in allowed_i^k \\ 0 & other \end{cases}$$

Choose a transfer node and add the initial node to the table:

$\eta_{ij}(t)$: Represents the heuristic information on the $< i, j >$ path at time t;

$\eta_{ir}(t)$: Represents the heuristic information on the $< i, r >$ path at time t;

$\eta_{ij}(t) = \frac{1}{J_{ij}}$: The reciprocal of the cost;

$tau_{ij}(t)$: Represents the pheromone concentration on the path of $< i, j >$ at time t;

$tau_{ir}(t)$: Represents the pheromone concentration on the path of $< i, r >$ at time t;

α,β:Respectively represent the weight coefficient of $\tau_{ij}(t)$,$\eta_{ij}(t)$;

$allowed_i^k$: indicates the neighboring point of position i that has not been visited.

(3-2) The ant selects the transfer node according to the transfer probability, and adds the selected transfer node to the table; judges whether the transfer node reaches the end point, if it does not reach the end point, repeat (3-2) until the end point is reached; If it reaches the end point then go to (3-3).

(3-3) Whether the number of iterations reaches a fixed value, if it does not reach the fixed value, go to (3-4) to update the pheromone, if the number of iterations reach a fixed value, go to (3-5) to update the pheromone.

(3-4) Update the path of this cycle according to the pheromone update formula, the number of iterations +1, go to (3-6).

(3-5) Update the path in the last few cycles according to the pheromone update formula, the number of iterations +1, go to (3-6).

(3-6) If the number of iterations is greater than the maximum algebra, the search is completed and the shortest path is obtained, otherwise, go to (3-1).

The information update formula is as follows:

$\tau_{ij}(t+h) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & k - thAntPassesOnThe < i, j > Path \\ 0 & other \end{cases}$$

ρ: indicates the pheromone volatilization coefficient;

Q: A constant representing the concentration of pheromone;

L_k: The total length of the path that ant k traverses in this cycle.

Step 4

Judge whether coordination can be achieved by smoothing the initial path of each drone, and perform corresponding operations based on the results.

(4-1) Smooth the angle that does not meet a predetermined angle at the initial path, find the smoothed path length interval.

(4-2) Take the maximum value of the lower limit of the corresponding path length interval of each drone and record it as A, and take the minimum value of the upper limit of the corresponding path length interval of each drone and record it as B.

(4-3) Determine the value of $A - B$.

(4-4) If $A - B \leq 0$, complete the collaboration.

(4-5) If $A - B > 0$, the collaboration cannot be completed.

Experiment Design

As I introduced before, in my project I only consider the delivery of drones in cities. The first city map I used is the map data of the Minneapolis and Saint Paul campuses of the University of Minnesota. At the same time, I divided these two into two control groups. One group is East Bank and West Bank, and the other group is East Bank and Saint paul. The reason for using these two sets of map data is that there are also long-distance and short-distance delivery for delivery by drones within the city. In order to better determine in which case my algorithm is better used, I use Two sets of map data are created. At the same time, we need to ensure that other data cannot be changed which means would not influence our result when we change the algorithm, so in my project, the customer and the package to be delivered are fixed, of course, for each of them has the same origin and destination. And to increase the problem complexity, I also use different package number as a variable to see how these algorithms work in this circumstance. Of course I know that in reality, the appearance of packages and customers is random, but in order to make my experiment more credible and to control variables, I fixed the information of customer and package. At the same time, because my ant algorithm has a certain degree of randomness, ants have different possibilities each time they choose a path. To ensure that I can get the most average answer, I calculated each algorithm 3 times and to find the average of them. Similarly, the algorithm runs on my computer, so there is no algorithm difference caused

by different computer performance. The running environment is MacBook with 1.2GHz Processors/8GB RAM in Mac-OS Catalina 10.15.7 with programming language C++.In my project, I will measure two data. The first data is the running time of the algorithm, which is how long the algorithm takes to get the (relative) best path. The second thing to consider is these The length of the path. Because we know that Ant Algorithm may not guarantee the best path, sometimes it will converge on the local shortest path. We score these two data in a 1:1 ratio, and add them to get a score. This score is the superiority of the algorithm in this scenario. The way I calculate the runtime is using the time class in C++ class, I get the timestamp before algorithm running and get the timestamp after it running. So then the difference of the timestamps will be the time it takes to compute the path.
To add more flavor, I also designed the visualization part of this project. This part uses the web server to create the scene and draw the map of our school. As the picture shows, the pink line is the optimal solution path. For some reason, the $\frac{1}{3}$ of the scene is in shade, for right now I don't know what's wrong. When testing the algorithm, I did not open the visualization, because communication with the web server for visualization may affect the test results.

Analysis

By comparing the experimental groups with different numbers of packages, we can find that when the problem is small, the optimal solution of the problem is obtained by the brute force search algorithm (Dijkstra Algorithm) and the result obtained by the ant algorithm is almost no difference. Through analysis, we know that when we consider that there are n packge centers and n customers, without considering the capacity constraints, the algorithm complexity of Dijkstra Algorithm is: $O(Vertex^2)$. Therefore, there is not much difference

Table 1: Numerical Results of Dijkstra Algorithm and Ant Algorithm

Environment	Num of Packages	Algorithm	Total Score	Route Length	Average Time
East Bank+West Bank	10	Dijkstra	1.32	1.18	0.14
East Bank+West Bank	20	Dijkstra	1.27	1.17	0.10
East Bank+West Bank	10	Ant	42.85	1.18	41.67
East Bank+West Bank	20	Ant	47.06	1.23	45.83
East Bank+Saint Paul	10	Dijkstra	246.18	3.01	243.17
East Bank+Saint Paul	20	Dijkstra	235.66	2.99	232.67
East Bank+Saint Paul	10	Ant	4.09	3.01	1.08
East Bank+Saint Paul	20	Ant	5.55	3.01	2.54

between the two. Ant Algorithm only needs 71% of the running time of Dijkstra Algorithm algorithm. At the same time, these two sets of data also show that when the package pressure is not too high, there is not much difference between which algorithm is used to calculate the best path of the drone. level. When the number of packages gradually increased, we found that the change curve of Dijkstra's calculation time was very flat, but Ant Algorithm had a big change. This is because, determined by the characteristics of Dijkstra Algorithm, when this algorithm is running, it has already calculated the best path between any two points on the map. Therefore, when we increase the number of packages, the complexity of Ant Algorithm is increased, but the complexity of Dijktra Alogarithm is not actually increased. When comparing the algorithm superiority of long and short distance parcel delivery, we can clearly feel that Ant Algorithm is better than Dijkstra Algorithm. Ant Algorithm can complete the calculation within 1ms, however, the calculation time of Dijkstra Algorithm has reached 200ms. Through analysis, we draw two conclusions:
Conclusion 1: When the delivery distance is short, Dijkstra Algorithm is better than Ant Algorithm. Due to the short delivery distance, the graph vertex included in the map is also less. Dijkstra Algorithm can spend very little time to figure out the best path between each point in the map. Moreover, in the case of short-distance delivery, increasing the number of packages will greatly increase the calculation time of the ant algorithm, but it will not significantly increase the running time of the Dijkstra Algorithm.
Conclusion 2: When the delivery distance is long, Dijkstra Algorithm is obviously inferior to Ant Algorithm. Due to the increase of environmental complexity, the calculation time of Dijkstra Algorithm, which is similar to Brute Force, will also increase.

Conclusion/Summary and future work

As I mentioned in the analysis, Dijkstra Algorithm is very useful in short-distance distribution scenarios, and its running speed compared with Ant Algorithm is not much different. When we increase the delivery distance, the superiority of ant algorithm is reflected. Because it is difficult for us to determine the scope of delivery in daily delivery, the applicability of

Ant Algorithm is better than that of Dijkstra. For future work, first of all, because I used the vehicle-UAV delivery model, in my project I only wrote about the second part of the delivery process, and ignored the part of the truck delivery. Therefore, the first improvement can be to add the truck delivery process. Secondly, since my program is running in docker, I use the time class of C++ to monitor the running time of the algorithm. However, I did not find a better way to measure memory. So another improvement should be to find out how to measure memory usage in docker, so that we can compare the memory usage of different algorithms.

References

[1] N. Agatz, P. Bouman, and M. Schmidt. Optimization Approaches for the Traveling Salesman Problem with Drone. *Transportation science*, 52(4):965–981, 2018.

[2] M. Bekhti, N. Achir, K. Boussetta, and M. Abdennebi. Drone Package Delivery: A Heuristic approach for UAVs path planning and tracking. *EAI endorsed transactions on internet of things*, 3(9):153048, 2017.

[3] P. Bouman, N. Agatz, and M. Schmidt. Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4):528–542, 2018.

[4] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà. On the min-cost Traveling Salesman Problem with Drone. *Transportation Research Part C: Emerging Technologies*, 86:597–621, 2018.

[5] S. H. Kim. Choice model based analysis of consumer preference for drone delivery service. *Journal of Air Transport Management*, 84:101785, 2020.

[6] S. Mourelo Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of industrial engineering and management*, 9(2):374–388, 2016.

[7] C. C. Murray and A. G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015.

[8] H. Ortega-Arranz, D. R. Llanos, and A. Gonzalez-Escribano. *The Shortest-Path Problem: Analysis and Comparison of Methods*, volume 1 of *Synthesis digital library of engineering and computer science*. Morgan & Claypool, San Rafael, California (1537 Fourth Street, San Rafael, CA 94901 USA), 2014.

[9] S. Poikonen and B. Golden. Multi-visit drone routing problem. *Computers & Operations Research*, 113:104802, 2020.

[10] B. M. Sathyaraj, L. C. Jain, A. Finn, and S. Drake. Multiple UAVs path planning algorithms: a comparative study. *Fuzzy optimization and decision making*, 7(3):257–267, 2008.

[11] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu. Routing and Scheduling for Hybrid Truck-Drone Collaborative Parcel Delivery With Independent and Truck-Carried Drones. *IEEE Internet of Things Journal*, 6(6):10483–10495, dec 2019.