Advances in Production Engineering & Management | **ISSN 1854-6250**
Volume 17 | Number 4 | December 2022 | pp 425–438 | Journal home: apem-journal.org
https://doi.org/10.14743/apem2022.4.446 | Original scientific paper

Monte Carlo Tree Search improved Genetic Algorithm for unmanned vehicle routing problem with path flexibility

Wang, Y.D.[a], **Lu, X.C.**[b,*], **Song, Y.M.**[c], **Feng, Y.**[d], **Shen, J.R.**[e]

[a]Beijing Jiaotong University, Shangyuan Village, Haidian District, Beijing, P.R. China
[b]Beijing Jiaotong University, Shangyuan Village, Haidian District, Beijing, P.R. China
[c]Beijing Jiaotong University, Shangyuan Village, Haidian District, Beijing, P.R. China
[d]Beijing Jiaotong University, Shangyuan Village, Haidian District, Beijing, P.R. China
[e]Beijing Capital Agribusiness & Food Group Co. Ltd., Xicheng District, Beijing, P.R. China

A B S T R A C T

With the gradual normalization of the COVID-19, unmanned delivery has gradually become an important contactless distribution method around China. In this paper, we study the routing problem of unmanned vehicles considering path flexibility and the number of traffic lights in the road network to reduce the complexity of road conditions faced by unmanned vehicles as much as possible. We use Monte Carlo Tree Search algorithm to improve the Genetic Algorithm to solve this problem, first use Monte Carlo Tree Search Algorithm to compute the time-saving path between two nodes among multiple feasible paths and then transfer the paths results to Genetic Algorithm to obtain the final sequence of the unmanned vehicles fleet. And the hybrid algorithm was tested on the actual road network data around four hospitals in Beijing. The results showed that compared with normal vehicle routing problem, considering path flexibility can save the delivery time, the more complex the road network composition, the better results could be obtained by the algorithm.

A R T I C L E I N F O

Keywords:
Unmanned vehicle;
Path flexibility;
Vehicle routing problem;
Genetic Algorithm (GA);
Monte Carlo Tree Search algorithm (MCTS);
COVID-19;
Pandemics

**Corresponding author:*
xclu@bjtu.edu.cn
(Lu, X.C.)

Article history:
Received 15 August 2022
Revised 9 October 2022
Accepted 15 October 2022

1. Introduction

The Novel Coronavirus Disease (COVID-19) broke out globally in 2020, and in November 2021, the mutant virus Omicron was discovered and spread rapidly around the world. According to WHO, the strain spreads faster and is more transmissible. By February 2022, the Omicron strain has become a major epidemic strain worldwide. Due to the wide range and high contagiousness of the epidemic, especially the significant characteristics of human-to-human transmission, the Chinese government has always taken strict precautions and implemented closed-off management in high-risk areas to minimize the spread of the epidemic. Thus, unmanned and contactless techniques have become important force in the fight against the epidemic. With the spread of the epidemic, more and more cities, such as Changsha and Shanghai have begun to use unmanned vehicles to distribute medical suppliers to the epidemic areas.

Compared to conventional manned vehicles, unmanned delivery vehicles are far more reliant on road conditions. For technical reasons, unmanned vehicles are less able to adapt to complex road conditions than normal vehicles. Hence the major research on unmanned vehicles usually focus on the design of algorithms based on uncertain traffic situations or obstacle avoidance for unmanned vehicles. However, under circumstance of the epidemic, considering of all technological aspects to plan thorough routes for unmanned vehicles can better meet the requirements of contactless delivery. The traditional vehicle routing problems consider the distribution sequences between demand points solely and ignore the fact that there are frequently multiple alternate paths between demand points in the actual distribution processes. The traffic situations and journey times of various routes are different and have substantial impact on the safety and efficiency of unmanned vehicles.

In reality, usually the more traffic lights need to go through, the more complex information needs to be processed, so this paper incorporates the number of traffic lights in the road network into the vehicle routing problem, considers to select a safer and faster path for unmanned vehicles. Establishing a vehicle routing problem model with paths flexibility (VRP-PF) with the goal of minimizing distribution time for unmanned vehicles.

2. Literature review

Since the unmanned vehicles relies heavily on road conditions and traffic situations, for unmanned delivery vehicles, most researchers take into account the unmanned vehicles' ability of avoiding obstacles and adapting to the traffic conditions.

Hu [1] *et al*. designed a genetic simulated annealing algorithm to solve the unmanned vehicles routing problem with road conditions updates, and adjusted the delivery plan in real time based on the local update strategy for the pre-optimize paths based on the road condition information, and the results showed that the algorithm has more advantages than traditional genetic algorithms under complex road conditions. Guan [2] proposed an unmanned vehicles routing problem model based on traffic situations. To improve the adaptability of unmanned vehicles to road conditions, a DQN local optimization model with heuristic reward and adaptive exploration strategy was proposed. This model could reduce delivery time and increase delivery efficiency, but it did not involve the obstacle avoidance problem of unmanned vehicles and was not tested on real traffic data. Zhu [3] and Han [4] both studied the unmanned vehicle routing problem using deep reinforcement learning, with the difference that Zhu's research targeted the obstacle avoidance problem of unmanned vehicles in dynamic environments, while Han's research considered the objective of minimizing distance and the number of vehicles from the perspective of unmanned fleets. Tavoosi [5] *et al*. designed an improved particle swarm algorithm to process certain and uncertain obstacles to obtain the optimal paths. Based on total driving distance and waiting time, Shi W. *et al*. proposed a multi-objective scheduling model to solve the path conflict problem of automated guided vehicles (AGVs) and used the A* path planning algorithm to search the shortest path of AGV [6]. Erenoglu used the Unmanned Aerial Vehicles (UAV) based 3D city modelling approach to be manage and plan urban areas [7]. In view of many scholars consider the real-time traffic conditions in unmanned vehicles routing problem, Wang [8] *et al*. proposed a routing model for unmanned vehicles in the case of GPS system failure. Additionally, Levy [9] *et al*. considered the unmanned vehicles routing problem under fuel-limited conditions and designed multiple neighbourhood shakes to improve the variable neighbourhood search algorithm, which was able to obtain better results compared with the traditional variable neighbourhood search algorithm but the result was not an optimal solution. Zhao [10] designed a genetic algorithm to solve the unmanned vehicles routing problem considering the charging and switching requirements of unmanned vehicles. It can be seen that the research on unmanned vehicles routing problem mainly focus on obstacle avoidance considering traffic conditions and energy supply of unmanned vehicles. Besides, heuristic algorithms are the main solution approaches used.

There are many different variants of the VRP, like Split-Delivery VRP [11], Heterogeneous VRP with Time Windows [12], stochastic VRP [13], VRP with Pickup and Delivery [14], ConVRP (Consistent Vehicle Routing Problem) [15] and EVRP (electric vehicle routing problem) [16] etc. But few research focus on the vehicle routing problem with path flexibility (VRP-PF). This problem was first defined by Huang [17] *et al.* in 2017, and they developed model considering time windows under deterministic and uncertain traffic conditions. The model was solved by CPLEX and obtained the approximate optimal solutions. Liu [18] *et al.* proposed a green vehicles routing model considering the fuel consumption of vehicle acceleration and waiting at traffic lights and established the model with path flexibility to minimizing fuel and other costs. Due to the high complexity of the model, it could only solve instances with 10 demand points. But their research proves that routes planning considering path flexibility can save costs. Wang [19] *et al.* proposed a vehicle routing problem model with path flexibility for electric vehicles, considering the selection of charging stations. Then they designed a variable neighbourhood search algorithm to solve this problem, but they only considered multiple choices of charging stations, multiple paths between the stations are not involved. Guo [20] *et al.* developed a time-dependent bus routing problem model considering traffic congestions with path flexibility, and designed a tabu search algorithm to solve it. Given that the safety of unmanned vehicles is highly dependent on road conditions and the distribution of epidemic supplies are time-efficient, we incorporate the waiting time during traffic lights to measure road conditions and establish a vehicle routing problem model with path flexibility. Then we design a hybrid heuristic algorithm, using Monte Carlo tree search algorithm to improve the Genetic Algorithm to solve the routing problem with path flexibility.

3. Model and algorithm of vehicle routing problem with path flexibility

3.1 Problem description and mathematic model

The vehicle routing problem with path flexibility is defined on a directed graph $G = (V, P)$, where $V = \{0,1,2,\ldots,n\}$ is the set of vertices, $P = \{(i,j,p): i,j \in V, i \neq j, p \in P_{ij}\}$ is the set of paths, and P_{ij} is the optional path from vertex i to vertex j. In the set V, vertex 0 denotes the starting point and the rest vertices denote the demand points. In the set P, the length of the path p_{ij} is d_{ijp}, the number of traffic lights is l_{ijp}. N denotes the set of unmanned vehicles, where all vehicles are homogeneous, and the capacity of the vehicle is C. The vehicle meets a red light with a certain probability S and the waiting time is w. The speed of unmanned vehicle is v_{ijp}, which is related to time and the road area where the unmanned vehicle is located. The objective is to minimise the time cost and find a solution that satisfies the following constraints: (1) the demand at each demand point is satisfied and the demand cannot be split; (2) all vehicles depart from and return to the starting point; (3) minimise the number of traffic lights in the optimal paths. An example diagram of this problem is shown in Fig. 1.

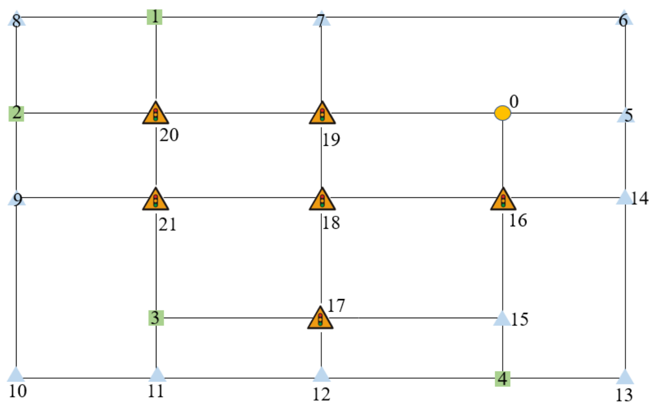


Fig. 1 Schematic diagram of VRP-PF

Assuming the fleet of unmanned vehicles are homogeneous, the yellow dot 0 in Fig. 1 indicates the departure point of all vehicles in the fleet, the green square dots 1,2,3,4 are demand points, the traffic light marker indicates that there is a traffic light at this junction and vehicles may need to wait for a while when the light is red. The blue triangular dots are non-demand nodes in the road network. Suppose that all nodes in the diagram are disconnected, and the unmanned vehicle needs to travel from point 0 to demand point 1, there are multiple paths to choose, such as 0-5-6-7-1, 0-19-7-1 or 0-19-20-1. For the road network with only 22 nodes shown in Fig. 1, there are more than dozens of feasible paths between each two demand points, which greatly increases the complexity of the problem than traditional routing problem.

Unmanned delivery vehicles are low-speed vehicles, and the Implementation Rules for the Management of Unmanned Delivery Vehicles promulgated by the Beijing Economic and Technological Development Area in May 2021 requires that the speed of unmanned delivery vehicles should not exceed 15 km/h. Besides, the speed of unmanned vehicles varies with the traffic conditions in real life. Therefore, we map the traffic indexes crawled from Baidu map to the speeds of unmanned vehicles, and the corresponding table of traffic indexes and speeds is shown in Table 1. The expression of speed is a piecewise function which is shown in Fig. 2.

Table 1 Correspondence between traffic index and travel speed

Real-time vehicle speed	Traffic index	Unmanned vehicle's speed
>30	1	15
10-30	2	10
<10	3	5

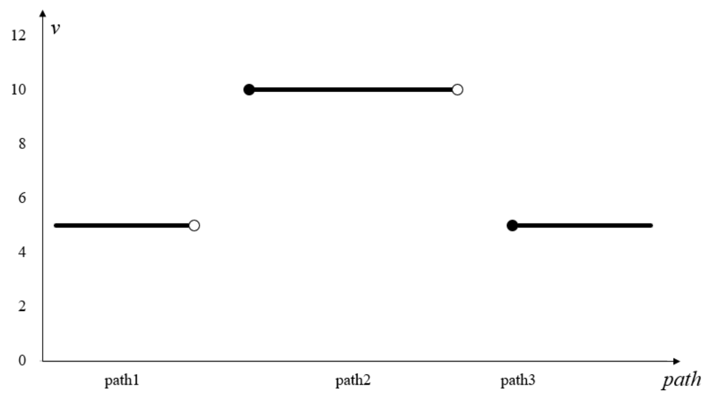


Fig. 2 Speeds of unmanned vehicles on different roads

The optimization objective in this paper is delivery time of unmanned vehicles, as shown in Eq. 1.

$$\min T = \sum_{(i,j)\in V} \sum_{p\in P} \frac{d_{ijp}}{v_{ijp}} x_{ijp}^{k} + Sw_{ijp} l_{ijp} x_{ijp}^{k}, \forall k \in N \tag{1}$$

Define the decision variables and associated parameters as follows.

$$x_{ij}^{k} = \begin{cases} 1, & \text{if arc } (i,j) \text{ } is \text{ on the optimal route} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ijp}^{k} = \begin{cases} 1, & \text{if the vehicle travels path } P_{ijp} \text{ on the arc } (i,j) \\ 0, & \text{otherwise} \end{cases}$$

C_{ijp}^{k} is actual capacity when vehicle k is on arc (i,j,p), and q_i is demand of customer i. Therefore, the question can be formulated as follows.

s.t.

$$\sum_{i\in V, i\neq j} x_{ij} - \sum_{i\in V, i\neq j} x_{ji} = 0, \qquad \forall j \in V \tag{2}$$

$$q_j x_{ij} \leq C_{ij}^{k} \leq (C - q_i) x_{ij}, \forall (i,j) \in V \tag{3}$$

$$\sum_{p\in P} x_{ijp}^{k} = x_{ij}, \qquad \forall (i,j) \in V \tag{4}$$

$$\sum_{p \in P} C_{ijp} = C_{ij}, \qquad \forall (i,j) \in V \tag{5}$$

$$q_j x_{ijp} \leq C_{ijp}^k \leq (C - q_i) x_{ijp}, \forall (i,j) \in V, p \in P \tag{6}$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in V \tag{7}$$

$$C_{ij} \geq 0, \forall (i,j) \in V \tag{8}$$

$$x_{ijp} \in \{0,1\}, \forall (i,j) \in V, p \in P \tag{9}$$

$$C_{ijp} \geq 0, \forall (i,j) \in V, p \in P \tag{10}$$

$$q_i > 0, \forall i \in V \backslash \{0\} \tag{11}$$

$$q_0 = 0 \tag{12}$$

Eq. 2 is the vehicle flow conservation constraint. Constraint Eq. 3 ensures that the volume of unmanned vehicles at each node does not exceed the maximum vehicle capacity. Eq. 4 ensures that each unmanned vehicle selects only one feasible path from node i to node j. Constraints Eqs. 5-6 ensure that supplies are transported on only one feasible path and do not exceed the maximum vehicle capacity. The rest are specific constraints of the variables.

3.2 Monte Carlo Tree Search algorithm (MCTS)

The vehicle routing problem belongs to NP-hard problem, and the vehicle routing problem with path flexibility is more difficult because the decisions to make are not only the routing decision but also the path selection decision depending on the departure time and the congestions in the relevant road network. The problem can be regarded as a two-stage problem with finding the optimal sequence of demand points and the optimal path selection between demand points. Moreover, the sequence of demand points affects path selection between points, and different paths affect the sequence of demand points in reverse. Since in the real road network, roads are crisscrossed and there are often numerous feasible paths between two points, Huang [17] *et al.* used Dijkstra's algorithm to find the shortest path between two points when considering path flexibility. However, the contrasting traffic conditions of different roads and the variational speed on different roads increase the difficulties of using exact algorithms. Hence, we apply the Monte Carlo tree search algorithm (MCTS) to solve the path flexibility of the problem and adopt MCTS to improve the genetic algorithm to solve the whole problem.

MCTS is a method for determining the optimal policy in a given domain. It is a simulation-based search algorithm with a tree structure that combines depth-first search and breadth-first search. Furthermore, it maintains superior results when the search space is huge and is widely used in fields such as games [21-24]. Therefore, MCTS is able to find the acceptable path rapidly for the large datasets. The process of MCTS can be divided into four steps: selection, expansion, simulation, and backpropagation, repeated these four steps until convergence [25].

The selection process is commonly implemented using the Upper Confidence Bound for Tree (UCT) algorithm, which searches and selects the next node to be visited among all the nodes, the formula of UCT is shown as Eq. 13.

$$UCT = \overline{X}_j + 2C_p\sqrt{\frac{2\ln n}{n_j}} \tag{13}$$

where n is the number of times the current parent node has been visited, n_j is the number of times the child node has been visited, C_p is a constant greater than zero, and the value of $\overline{X}_j$ usually between [0,1], [24].

Even though the UCT strategy can provide acceptable outcomes, the rate of convergence is modest. To improve the efficiency, the weight of the node depends on the travel time, and the shorter the travel time is, the higher the weight is and the node is more likely to be selected. Therefore, the pseudo for the algorithm can be written as Algorithm1.

Algorithm1：path search for MCTS

Input：origin node n_o, destination node n_d
Output: optimal path p between n_o and n_d
function PathSearch($Array$, n_o, n_d)
Disconnect n_o from other nodes and initialize trial nodes list L
if $Array[d_{oj}]$!= 0 **then**
  $L \leftarrow$ add n_j to L
end if
while n_i != n_d **do**
  $n_i \leftarrow$ Select a node n_i from L according to the weight
  $p \leftarrow$ add n_i to p
end while
return p
end function

Unlike complex games such as Go (Weiqi), game theory is not involved during the routing search process and therefore there is no need to switch decisions, so we omit the backpropagation process. The routing search is able to find a solution inevitably (if the path is not feasible can be named as a non-feasible solution), so by setting a certain number of iterations we can implement the simulation process in MCTS. By simulating k_{max} times, the final output of the path with the highest weight can be fed back to the genetic algorithm, the flow chart of MCTS algorithm is shown in Fig. 3.

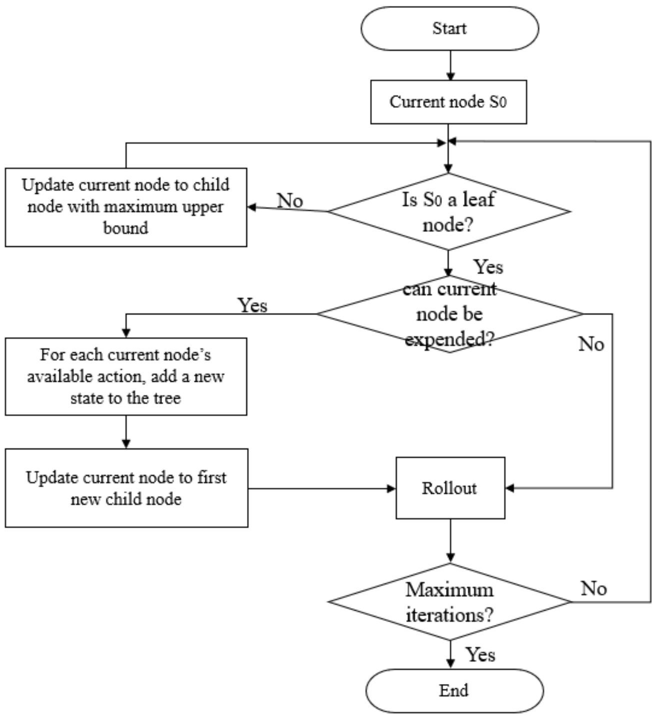


Fig. 3 Flow Chart of MCTS Algorithm

3.3 Monte Carlo Tree Search Improved Genetic Algorithm (MCTS-GA)

The Monte Carlo tree search algorithm can find the optimal path between two nodes effectively, but it cannot solve the vehicle routing problem. In contrast, the genetic algorithm has better search capability and scalability, so we use the framework of the genetic algorithm and combined with the Monte Carlo tree search algorithm to solve the entire problem.

Coding design

Chromosomes are encoded using decimal coding that is frequently used in genetic algorithms in this paper, but each node includes an attribute label. The nodes in the road network are divided into four categories, starting nodes (point 0 in Fig. 1), customer demand nodes (green square nodes in Fig. 1), road nodes without demand (blue triangular nodes in Fig. 1) and traffic lights nodes (nodes with traffic light signs in Fig. 1), which together form a complete road network. In the actual distribution work, the customer demand nodes are the nodes that required to make decisions about the order of distribution for the vehicle routing problem, while the road nodes and traffic lights nodes are the nodes that required to make decisions about whether or not to pass for path flexibility.

Therefore, the attribute labels of the different nodes need to be assigned to the nodes in order for the algorithm to identify the nodes to be processed at different stages when coding. A schematic of the coding scheme is shown in Fig. 4 (using a feasible path in the road network of Fig. 1 as an example).

Fig. 4 shows a complete feasible route for an unmanned vehicle, which includes both the demand nodes to be delivered and all road nodes in the road network to be passed through during delivery. But when computing the vehicle routing problem with genetic algorithm, the code of chromosome can be streamlined to retain only the starting nodes and customer demand nodes, which means the code shown in Fig. 4 only needs to be processed as (0,2,1,0). This route is the delivery route for an unmanned vehicle without path flexibility (assuming that the sum of the demand does not exceed the vehicle capacity). In contrast, in the process of MCTS, the entire road network needs to be processed, that means the path with minimized travel time and traffic lights between each two nodes in the route needs to be found in turn. For example, a feasible path from the starting point to customer demand node 2 can be expressed as (0,19,20,2).

0	19	20	2	8	1	7	6	5	0
Origin	Traffic light	Traffic light	Customer node	Road node	Customer node	Rode node	Rode node	Rode node	Origin

Fig. 4 Schematic of complete route coding scheme

Crossover, mutation and heredity

Both the crossover and mutation operations are directed at the vehicle routing problem, without considering path flexibility. The crossover process uses a two-opt crossover. To avoid duplicate fragments or missing fragments in the offspring generated after the crossover, we first randomly select sample fragments from parent 2 and inserts them into the corresponding positions of the offspring, and then traverses parent 1 and inserts the genes that are not duplicated with the sample fragments therein in turn to form new offspring. The crossover process is shown in Fig. 5 (the crossover process does not consider road nodes and traffic lights nodes, assuming that all nodes except node 0 in Fig. 5 are customer demand nodes).

The mutation process is a two-point mutation, which means that two genes on a chromosome that do not contain the first and last gene are randomly selected to swap positions. To improve the efficiency of the search, an elitist selection strategy is used, whereby the top 2 % of the offspring in each generation are retained and placed directly into the next generation. Besides, the idea of an invasive weed algorithm is involved so that the more adapted individuals produce relatively more offspring.

P1	0	19	20	2	8	1	7	6	5	0
P2	0	6	20	1	7	5	8	2	19	0
Child	0	19	20	1	7	5	8	2	6	0

Fig. 5 Schematic diagram of the crossover process

Adaptability function

After the allocation sequence of customer demand nodes has been generated by the genetic algorithm, the MCTS can be used to obtain the optimal path between the customer demand nodes then the travel time and total length of the path as well as the times of traffic lights passed can be calculated. The individual with the greater the fitness is more retainable, so according to Eq. 1 the fitness function can be expressed as Eq. 14.

$$fitness = {}^{1}/_{T} \tag{14}$$

MCTS-GA algorithm flow

The above describes the process of selecting feasible paths between nodes using the MCTS algorithm and the solution of the vehicle routing problem using genetic algorithm respectively, but the two processes need to be carried out in unison to obtain a complete solution for the problem. The flowchart of the Monte Carlo Tree Search Improved Genetic Algorithm (MCTS-GA) is shown in Fig. 6.

After generating the sequence of customer demand nodes by the genetic algorithm, we use the MCTS to find the optimal path between each two nodes and the search result is transferred to the genetic algorithm for population fitness calculation. Then cross and mutation operations are carried out to generate the new population, and the complete path is finally output after repeated iterations until reach the maximum number of iterations.

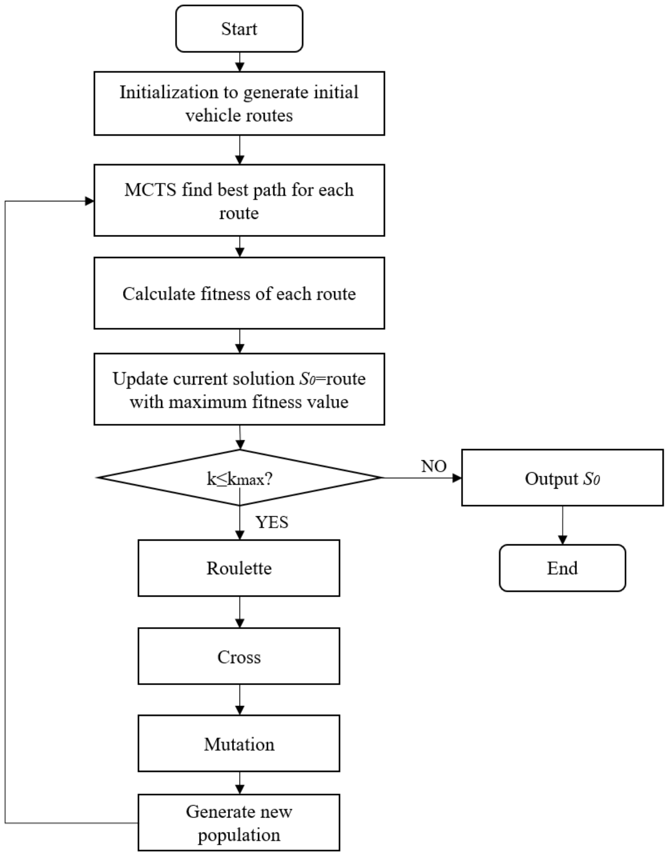


Fig. 6 Flow chart of MCTS-GA algorithm

4. Numerical experiments: Results and discussion

In this section, the algorithm performance is tested by numerical experiments using real road network in Beijing. As adopting unmanned vehicles to delivery epidemic medicine is considered, we choose some of the hospitals in Beijing as the starting points for medicine distribution. The unmanned vehicles start from the hospital to the community, and the road networks around the target hospitals are established as examples for testing.

The algorithm is implemented using Python code in an Intel(R) Core (TM) i5-8250U CPU @ 1.60 GHz and 8GB of RAM. The parameters of algorithm and unmanned vehicles are shown in Table 2, the example information is shown in Table 3.

Table 2 Algorithm parameters and unmanned vehicle performance parameter settings

Parameters	Value
Crossover probability	0.8
mutation probability	0.2
Iterations of GA	100
Iterations of MCTS	500
Probability of encountering a red light	0.5
Red light waiting time	90 seconds
capacity of unmanned vehicle	20 pieces

Table 3 Instance information table

Instance name	Number of customer nodes	Number of road nodes	Number of traffic lights	Total amount of nodes
Hospital23	6	11	5	23
Hospital29	8	17	3	29
Hospital39	11	21	7	39
Hospital58	15	28	14	58

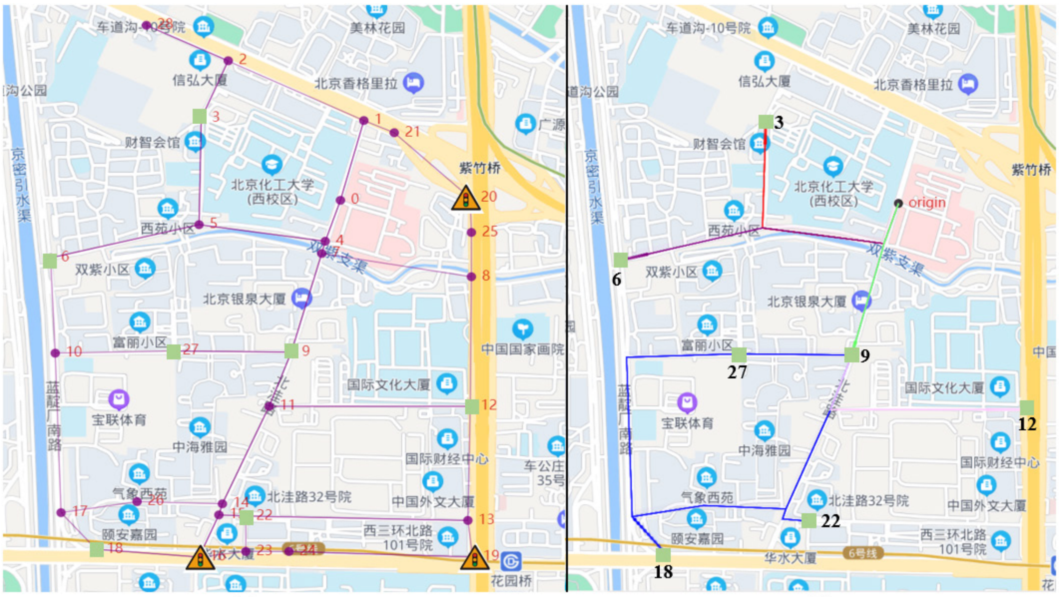
In this paper, three Beijing hospitals are selected as target hospitals, and two small-scale road networks and one medium-scale road network were established as examples. Example Hospital23 is the Sixth Hospital of Peking University, with a road network consisting of 23 nodes; Example Hospital29 is Hospital 466, with a total of 29 nodes in the road network. Example Hospital39 is the Chinese Armed Police General Hospital, with a total of 39 nodes in the road network. Example Hospital58 is the Bayi Children's Hospital, with a total of 58 nodes in the road network. The results of examples are shown in Figs. 7-10 respectively, where the road network is shown in Figs. (a) and the distribution results are shown in Figs. (b), and the green square nodes in Figs. (b) are customer demand nodes. Tables 4-7 show the specific distribution paths of each example.

(a) (b)

Fig. 7 Results of Hospital23

Table 4 Detailed distribution paths of Hospital23

Unmanned vehicle	Customer nodes	Load pieces	Load Factor (%)	Distribution path	Times of passing through traffic lights
1	21,13,1	16	80	0,2,21,13,12,1,0	0
2	8,5	20	100	0,2,3,8,9,10,11,6,5,4,3,2,0	3
3	19	15	75	0,2,21,13,14,15,19,15,14,13,21,2,0	2

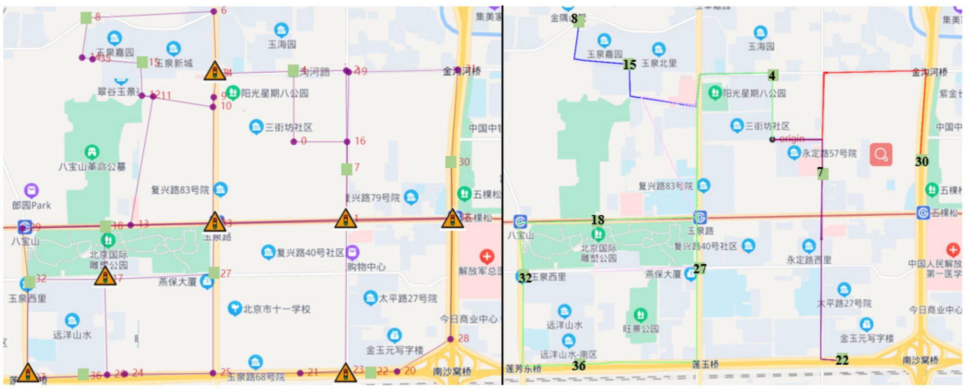


(a) (b)

Fig. 8 Results of Hospital29

Table 5 Detailed distribution paths of Hospital29

Unmanned vehicle	Customer nodes	Load pieces	Load factor	Distribution path	Times of passing through traffic lights
1	18,22,27	20	100	0,4,7,9,11,14,15,22,15,14,26 ,17,18,17,10,27,9,7,4,0	0
2	3	20	100	0, 4, 5, 3, 5, 4,0	0
3	6	18	90	0, 4, 5, 6, 5, 4,0	0
4	9,12	19	95	0, 4, 7, 9, 11, 12, 11, 9, 7, 4,0	0
5	20	20	100	0, 1, 21, 20, 25, 8, 7, 4,0	0
6	28	18	90	0,4,5,3,2,28,2,3,5,4,0	0

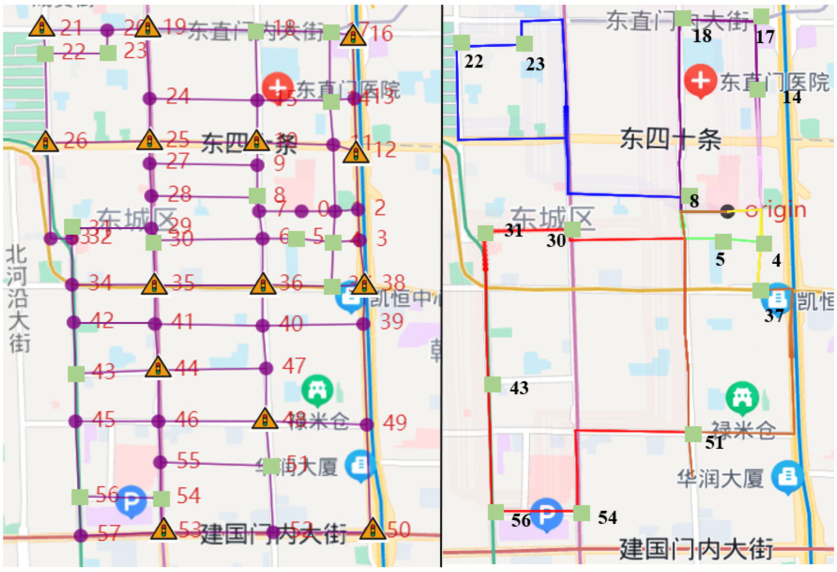


(a) (b)

Fig. 9 Results of Hospital39

Table 6 Detailed distribution paths of Hospital39

Unmanned vehicle	Customer nodes	Load pieces	Load factor (%)	Distribution path	Times of passing through traffic lights
1	8,15	20	100	0,4,3,9,10,12,15,35,14,8,1 4,35,15,12,10,9,3,4,0	0
2	7,30	17	85	0,16,7,16,19,31,30,31,19,1 6,0	0
3	22	18	90	0,16,7,1,23,22,23,1,7,16,0	4
4	4,18,27	19	95	0,4,3,9,10,11,5,18,17,27,5, 9,3,4,0	2
5	24,32,36	16	80	0,4,3,9,5,27,25,24,26,36,3 7,32,29,18,13,5,9,3,4,0	3



(a) (b)

Fig. 10 Results of Hospital58

Table 7 Detailed distribution paths of Hospital58

Unmanned vehicle	Customer nodes	Load pieces	Load factor (%)	Distribution path	Times of passing through traffic lights
1	22,23	19	95	0, 7, 8, 28, 27, 25, 24, 19, 20, 23, 22, 26, 25, 27, 28, 8,7,0	4
2	30,31,43,56,54	20	100	0, 7,6,30,29, 31, 32, 34, 42, 43, 45, 56, 54, 55, 46, 48, 47, 40, 36, 6,7,0	2
3	17,18,8	17	85	0,1,11,14,17,18,15,10,9, 8,7,0	1
4	14	15	75	0, 1, 11,14, 11, 1,0	0
5	5	12	60	0, 7, 6, 5, 4, 1,0	0
6	4,51	17	85	0, 1, 4, 37, 38, 39, 49, 48, 51, 48, 47, 40, 36, 6, 7,0	4
7	37	20	100	0, 1, 4, 37, 4, 1,0	0

Table 8 Comparison of VRP-PF and VRP-SP

Indicator	Instance	VRP-PF	VRP-SP	*D*
Total time (min)	Hospital23	20	23	-3
	Hospital29	44	50	-6
	Hospital39	59	69	-10
	Hospital58	64	79	-15
Total distance (m)	Hospital23	5742	5532	210
	Hospital29	11380	9680	1700
	Hospital39	15188	12352	2836
	Hospital58	16322	15243	1079
Times of passing through traffic lights	Hospital23	5	7	-2
	Hospital29	0	2	-2
	Hospital39	9	21	-12
	Hospital58	11	16	-5

The results illustrate that 3 unmanned vehicles are required for delivery in Example Hospital23, with an average load factor of 85 %. 6 unmanned vehicles are required for delivery in Example Hospital29, with an average load factor of 96 %. 5 unmanned vehicles are required for delivery in Example Hospital39, with an average load factor of 90 % and 7 unmanned vehicles are required for delivery in Example Hospital58, with an average load factor of 86 %. As Hospital23, Hospital39 and Hospital58 contains many traffic lights in the road networks, it is difficult to avoid going through the traffic lights during path selection. In the result of Example Hospital39, the times of passing through traffic lights is obviously reduced. Table 8 shows the results of considering path flexibility compared with considering the shortest path between two points, where VRP-PF is the results considering path flexibility using the algorithm proposed in this paper, VRP-SP is the results without considering path flexibility using Dijkstra algorithm to find the shortest path between two points. Meanwhile, *D* is the optimization difference, where the negative value means our algorithm got better results than Dijkstra algorithm.

From the results, it can be seen that considering path flexibility can reduce the times of passing through traffic lights and improve the delivery efficiency at the expense of increasing the total path length. For the sake of presentation, the reduction in total distance is divided by a thousand to uniform the order of magnitude with other indicators in Fig. 11. It can be seen that in Example Hospital58, total time decreases 15 minutes compared with Dijkstra algorithm while total time only decreases 3 minutes in Example Hospital23. The times of passing through traffic lights reduces 5 and 12 times in Example Hospital58 and Hospital39 respectively while this indicator only reduce 2 times in Hospital29 and Hospital23. The optimization is more obvious in the case of complex road network with more traffic lights.

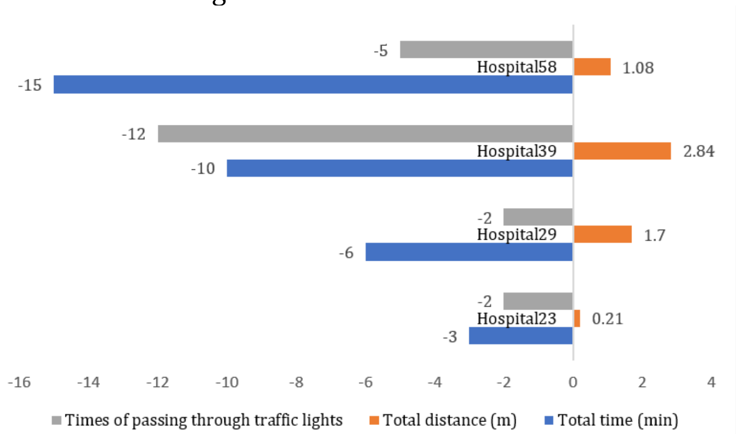


Fig. 11 Comparison of numerical examples

5. Conclusion

The normalization of epidemics has prompted the development of contactless distribution. In this paper, the path flexibility between demand points is considered when dealing with vehicle routing problem for unmanned vehicles. In the case where there are multiple feasible paths between two demand points, the number of traffic lights in the road network is considered from the perspective of driving safety of unmanned vehicles. We build the mathematic model of path flexibility vehicle routing problem with the objective to minimize the distribution time. The MCTS algorithm is used to select the feasible paths between two nodes, and the results are fed back to the genetic algorithm for further optimization to finally determine the complete driving scheme. Finally, we adopt actual road networks in Beijing as examples, obtain solutions under different sizes of road networks, and the results show that the algorithm can select better paths with less driving time for different sizes of instances, and can maintain a high vehicle full load rate. Compared with routing problem that does not consider path flexibility, the results show that considering path flexibility can not only reduce the delivery time but also reduce the times of passing through traffic lights for unmanned vehicles. For more complex road conditions, the better results the algorithm can get.

The algorithm considering path flexibility can not only be used for the routing planning of unmanned vehicles in the distribution of emergency supplies, but also for the path planning of AGV picking and distribution in the warehouse. In the process of vehicle navigation, the algorithm based on Monte Carlo tree search can also flexibly take into account the driver's experience and other factors, and get better results in the congested road sections. Hence the algorithm proposed in this paper can also enrich the navigation algorithms. However, the method requires complete road network information in the process of preliminary data preparation, including node coordinates, connectivity between nodes as well as distances and driving speed of segmented roads, which is the weakness of the algorithm. In addition, for unmanned vehicles, real-time road condition can also be considered to be passed into the algorithm to improve distribution efficiency while improving the ability of unmanned vehicles to cope with complex road conditions and driving safety.

Acknowledgement

We thank the Editor and anonymous referees for their many helpful comments on an earlier version of our paper. This work was supported in part by the National Natural Science Foundation of China under grant numbers 72171016; and the Beijing Logistics Informatics Research Base.

References

[1] Hu, H., Tang, Z.Q., Liu, F.X., Wang, Y.Q., He, X.F., Zhao, J. (2022). Optimization of contactless distribution of medical protective materials under the epidemic, *Chinese Journal of Management Science*, 1-11, doi: 10.16381/j.cnki.issn1003-207x.2020.1925.

[2] Guan, M. (2021). *Research on route planning of unmanned vehicle logistics distribution,* Master's thesis, Beijing University of Posts and Telecommunications, Beijing, China, doi: 10.26969/d.cnki.gbydu.2021.002076.

[3] Zhu, J. (2021). *Research on path planning of unmanned vehicles based on deep reinforcement learning*, Master's thesis, Xi'an University of Technology, Xi'an, China, doi: 10.27391/d.cnki.gxagu.2021.000661.

[4] Han, Y.F. (2021). *Research on distribution path planning of unmanned logistics fleet based on deep reinforcement learning*, Master's thesis, Dalian University of Technology, Dalian, China, doi: 10.26991/d.cnki.gdllu.2021.002725.

[5] Tavoosi, V., Marzbanrad, J., Golnavaz, M. (2020). Optimised path planning of an unmanned vehicle in an unknown environment using the PSO algorithm, *IOP Conference Series: Materials Science and Engineering,* Vol. 671, Article No. 012009, doi: 10.1088/1757-899x/671/1/012009.

[6] Shi, W., Tang, D.B., Zou, P. (2021). Multi-objective automated guided vehicle scheduling based on MapReduce framework, *Advances in Production Engineering & Management,* Vol. 16, No. 1, 37-46, doi: 10.14743/apem2021.1.383.

[7] Erenoglu, R.C., Erenoglu, O., Arslan, N. (2018). Accuracy assessment of low cost UAV based city modelling for urban planning, *Tehnički Vjesnik – Technical Gazette*, Vol. 25, No. 6, 1708-1714, doi: 10.17559/TV-20170904202055.

[8] Wang, B., Misra, S., Sundar, K., Rathinam, S., Sharma, R. (2018). Routing multiple unmanned vehicles in GPS denied environments, In: *AIAA Information Systems-AIAA Infotech @ Aerospace, AIAA SciTech Forum*, 1-10, doi: 10.2514/6.2018-0897.

[9] Levy, D., Sundar, K., Rathinam, S. (2014). Heuristics for routing heterogeneous unmanned vehicles with fuel constraints, *Mathematical Problems in Engineering*. Vol. 2014, Article ID 131450, doi: 10.1155/2014/131450.

[10] Zhao, S.Y. (2021). *Research on path optimization of unmanned ground delivery vehicles in charging and power swapping model,* Master's thesis, Beijing Jiaotong University, Beijing, China, doi: 10.26944/d.cnki.gbfju.2021.003341.

[11] Min, J.N., Jin, C., Lu, L.J. (2019). Split-delivery vehicle routing problems based on a multi-restart improved sweep approach, *International Journal of Simulation Modelling*, Vol. 18, No. 4, 708-719, doi: 10.2507/IJSIMM8(4)CO19.

[12] Liu, M.L., Zhang, C., Wu, Q.L., Meng, B.R. (2021). Vehicle routing problem with soft time windows of cargo transport O2O platforms, *International Journal of Simulation Modelling*, Vol. 20, No. 2, 351-362, doi: 10.2507/IJSIMM20-2-564.

[13] Anuar, W.K., Lee, L.S., Seow, H.-V., Pickl, S.A. (2021). A multi-depot vehicle routing problem with stochastic road capacity and reduced two-stage stochastic integer linear programming models for rollout algorithm, *Mathematics*, Vol. 9, No. 13, Article No. 1572, doi: 10.3390/math9131572.

[14] Amalia, I.S., Bakhtiar, T., Jaharuddin. (2021). Minimizing the handling time in VRP with pickup-and-delivery and time windows, *Journal Conference Series, of Physics,* Vol. 1863, Article No. 012002, doi: 10.1088/1742-6596/1863/1/012002.

[15] Barros, L., Linfati, R., Escobar, J.W. (2020). An exact approach for the consistent vehicle routing problem (ConVRP), *Advances in Production Engineering & Management*, Vol. 15, No. 3, 255-266, doi: 10.14743/apem2020.3.363.

[16] Huang, A.Q., Zhang, Y.Q., He, Z.F., Hua, G.W., Shi, X.L. (2021). Recharging and transportation scheduling for electric vehicle battery under the swapping mode, *Advances in Production Engineering & Management*, Vol. 16, No. 3, 359-371, doi: 10.14743/apem2021.3.406.

[17] Huang, Y., Zhao, L., Van Woensel, T., Gross, J.-P. (2017). Time-dependent vehicle routing problem with path flexibility, *Transportation Research Part B: Methodological*, Vol. 95, 169-195, doi: 10.1016/j.trb.2016.10.013.

[18] Liu, X., Qi, M., Cheng, C. (2017). Green vehicle routing problem with path flexibility: In: *Proceedings of 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, 1037-1041, doi: 10.1109/IEEM.2017.8290050.

[19] Wang, L., Gao, S., Wang, K., Li, T., Li, L., Chen, Z. (2020). Time-dependent electric vehicle routing problem with time windows and path flexibility, *Journal of Advanced Transportation*, Vol. 2020, Article ID 3030197 doi: 10.1155/2020/3030197.

[20] Guo, R., Zhang, W., Guan, W., Ran, B. (2021). Time-dependent urban customized bus routing with path flexibility, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 4, 2381-2390, doi: 10.1109/TITS.2020.3019373.

[21] Sironi, C.F., Liu, J., Winands, M.H.M. (2020). Self-adaptive Monte Carlo tree search in general game playing, *IEEE Transactions on Games*, Vol. 12, No. 2, 132-144, doi: 10.1109/TG.2018.2884768.

[22] Sironi, C.F., Winands, M.H.M. (2018). Analysis of self-adaptive Monte Carlo tree search in general video game playing, In: *Proceedings of 2018 IEEE Conference on Computational Intelligence and Games (CIG)*, Maastricht, Netherlands, 1-4, doi: 10.1109/CIG.2018.8490402.

[23] Kimura, K., Tu, Y., Tanji, R., Mozgovoy, M. (2021). Creating adjustable human-like AI behavior in a 3D tennis game with Monte-Carlo tree search, In: *Proceedings of 2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Orlando, USA, 1-6, doi: 10.1109/SSCI50451.2021.9659551.

[24] Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S. (2012). A survey of Monte Carlo tree search methods, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, 1-43, doi: 10.1109/TCIAIG.2012.2186810.

[25] Zhou, K., Hong, Z., Hu, L. (2019). Cable diameter control based on Monte Carlo tree search, *Computer Engineering and Design*. Vol. 40, No. 8, 2389-2395, doi: 10.16208/j.issn1000-7024.2019.08.048.