Received 5 January 2024, accepted 24 January 2024, date of publication 19 February 2024, date of current version 23 February 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3359644

RESEARCH ARTICLE

An Unmanned Sweeper Path Planning Algorithm for Structured Roads

LEI HE, BAOYUN WANG, YUNSHAN PENG, AND XIUCAI ZHANG
State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China
Corresponding author: Lei He (jlu_helei@jlu.edu.cn)

This work was supported in part by the Project on Research and Development of Intelligent Vehicle Key Technologies and Industrialization Projects Based on New Energy Vehicles under Grant TC210H02S, and in part by the Project on Quantitative Development and Measurement Technology Research of Expected Functional Safety Based on Vehicle-Cloud Collaboration under Grant 20220301012GX.

ABSTRACT This paper presents a path planning algorithm to address the issue of current traversal planning algorithms being unsuitable for lane traversal by unmanned sweepers on structured roads. The algorithm treats lanes in structured roads as nodes and utilizes the A* algorithm to compute the path distances between all lane nodes, constructing a cost matrix, and the pre-traversal lane node sequence is found through the heuristic algorithm. The A* algorithm is then used to search for the shortest path between adjacent pre-traversal lane nodes, and this shortest path lane sequence is added to the pre-traversal node sequence. Pruning optimization is applied to the obtained node sequence, generating the optimal lane traversal node sequence and converting it to the corresponding lanes to achieve the structured road lane traversal. Experimental results in various structured road scenarios indicate that the proposed algorithm can effectively and completely traverse structured roads.

INDEX TERMS Unmanned sweeper, path planning, A* algorithm, LKH-3, pruning optimization.

I. INTRODUCTION

Driverless sanitation sweepers are essential for future cities to improve cleaning efficiency, reduce costs, and decrease reliance on human labor [1], [2]. Establishing lane traversal path planning algorithms for structured roads can improve the intelligence of sanitation transportation systems, which is important for constructing unmanned sweepers and smart cities [3].

The lane traversal path planning algorithm for unmanned sweepers on structured roads is distinct from coverage path planning algorithms. Coverage path planning algorithms are primarily used in indoor cleaning robots [4], agricultural machines [5], etc., focusing on full coverage of the entire area. In the context of structured road sweeping operations, the objective is to traverse all lanes and clean the edges of the road in the shortest path, that is, with the minimum energy consumption. The lane traversal path planning problem can be transformed into a combination problem of finding the shortest path within a discrete domain and traversal optimal path.

Solving the shortest path planning problem requires finding the shortest path from a known start node to a target node with known path information [6]. The path information is usually static, and even if the information changes, intelligent algorithms can perform timely contingency planning. Commonly used algorithms include Dijkstra algorithm, Floyd's algorithm, A* search algorithm, etc. [7], [8], [9], [10]. Dijkstra's algorithm is a classical algorithm used to find the shortest path from a single source in a graph, which is usually applicable to the case of no specific goal node. A* algorithm is highly efficient in finding the shortest path between two specific vertices in a graph, and it is well suited for the path planning problem with known maps, and it is widely used in the field of robotics. Floyd's algorithm can handle directed graphs and graphs with negatively weighted edges, but it has higher computational time complexity than the A* algorithm and is limited in the computation of large-scale graphs.

Solving the traversal optimal path problem is similar to solving the Asymmetric Traveling Salesman

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Gao.

Problem (ATSP). Genetic algorithms, dynamic programming methods, heuristic algorithms, etc., are usually used to solve such problems [11], [12]. However, ATSP is an NP-hard problem, which is a class of problems with roughly exponential time complexity [13]. As the problem size increases, exact algorithms are no longer applicable, when the Lin and Kernighan heuristic family of algorithms is most suitable [14]. The standard Lin-Kernighan heuristic algorithm is one of the classical methods for generating optimal or near-optimal solutions for the symmetric Traveling Salesman Problem (TSP) and belongs to the so-called local search algorithms. Based on this algorithm, Keld Helsgaun designed the LKH algorithm by modifying Lin and Kernighan's heuristic rules for restricting and guiding the search, using 5-opt sequential moves as the basic move components, and applying sensitivity analysis to guide and restrict the search process, which can find the optimal solution of a large-scale TSP problem in a reasonable runtime [15]. The LKH-2 algorithm [16], which is improved on the basis of the LKH algorithm, combined with the transformation method proposed by Jonker and Volgenant [17], can find the optimal solution of the ATSP problem and enhance the efficiency of the algorithm in dealing with the large-scale problems. The LKH-3 algorithm introduces the penalty function to deal with the constraints based on the LKH-2 algorithm, which can efficiently solve the nonstandard ATSP problems [18].

Lanes in structured roads form a network structure interconnected based on certain rules. When unmanned sweepers traverse these lanes, they face numerous constraints, e.g., they have to obey traffic rules, cannot change lanes arbitrarily, turn around to go against traffic, etc. Essentially, this problem falls under the category of traversal routing problems in graph theory [19]. It is similar to the Chinese Postman Problem (CPP), which is usually formulated using the Edmonds and Johnson algorithm [20], but the performance of this algorithm is limited when dealing with large-scale, asymmetric, and graphs that do not satisfy specific properties. In addition, the problem to be solved becomes an NP-hard problem with directed graphs under the condition that vehicles are not allowed to travel against traffic due to the limitation of lane direction.

All these methods mentioned above are difficult to directly apply to lane traversal path planning on structured roads. Therefore, this paper proposes a lane traversal path planning algorithm for unmanned sweepers based on the connection structure of structured roads to solve the problem of lane traversal path planning for unmanned sweepers on structured roads.

Contributions

There is currently few research on lane traversal path planning for unmanned sweepers applicable to structured roads, and our contribution lies in complementing the gap in this particular scenario.

We have investigated the A* algorithm, the CPP problem, and the TSP problem, and introduced the TSP problem into our study and modified it accordingly. By applying the A* algorithm and the classical solutions to the TSP problem, we have addressed the problem of lane traversal path planning for unmanned sweepers applicable to structured roads, and our study can afterwards be used as a benchmark for this particular area.

II. MODELING THE LANE TRAVERSAL PLANNING PROBLEM FOR UNMANNED SWEEPERS

A. ASSUMPTIONS ON THE ISSUE OF UNMANNED SWEEPERS

Assumption 1: Unmanned sweepers operate at relatively low speeds during sweeping operations and generally work in the early morning hours during periods of low traffic flow when there are few strong interaction scenarios.

Assumption 2: Due to the high velocity air currents generated by the many other vehicles traveling on the roads and the impact of the sprinkler streams, litter waste usually accumulates at the edges of the roads. Therefore, sweepers need to sweep close to the edge of the roads, i.e., along the left or right side of the roads, depending on state regulations. At this point, the focus of the unmanned sweeping on structured roads is to plan the optimal path to traverse all lanes.

Assumption 3: The lane traversal planning problem of unmanned sweepers mainly considers traversal sweeping all lanes at the minimum cost and improving the traversal efficiency (reducing the repeat sweeping rate) as much as possible to reduce energy consumption. Returning to the base station after the sweeping job does not belong to the traversal cost, otherwise, repeating the sweeping of lanes that have already been traversed on the return trip will reduce the traversal efficiency.

B. REPRESENTATION OF THE STRUCTURED ROAD NETWORK

The structured road network based on Lanelets representation is the topologically connected structure of structured roads, as shown in Fig. 1. Lanelets [21] is a data structure for describing geographic information about roads, which is commonly used to model lane geometry, topological relationships, and other related information and is widely used in self-driving path planning. Lanelets lane line geometry is defined by a left boundary and a right boundary. Each boundary corresponds to an interconnected polyline, which simulates the road boundaries of the left and right lanes in real situations, while the polylines implicitly represent the direction of travel. Each lane has adjacent left and right lanes in addition to the preceding and successor lanes. Intersections are also an essential part of the structured road network, as shown in Fig. 2, where the lanes overlap, but each lane represents the corresponding road travel direction.

C. MODELING THE LANE TRAVERSAL PLANNING PROBLEM

The lane traversal planning problem of unmanned sweepers on structured roads refers to planning a complete and feasible path in a given map area so that the unmanned sweepers can

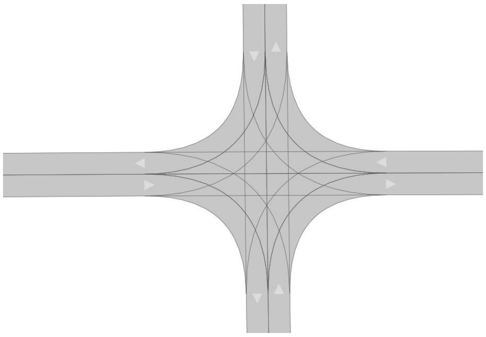

FIGURE 1. Structural representation of Lanelets.

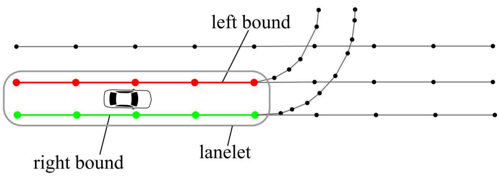

FIGURE 2. Schematic diagram of the intersection.

traverse all the lanes. The lane repetition sweeping rate of the traversal path should be as low as possible so that the total path distance is the shortest to reduce energy consumption. In this paper, each lane is regarded as a node, and the length information of the lane is added to the path distance between lane nodes, at which point the problem is to traverse all lane nodes in a given region of the structured roads and make the traversal path the shortest total distance.

In the structured road network, vehicles can only travel in the direction specified by the road. The shortest path from lane node i to lane node j is often different from the shortest path from j to i, which is because it may be possible to pass from i to j with vehicles traveling straight ahead, but due to the limitation of the direction of lane access, under the condition that vehicles are not allowed to travel against the flow of traffic, it is generally necessary to go through a detour in other lanes in order to return to the i node when traveling from j to i. The problem to be solved at this point is the NP-hard problem with directed graphs.

III. ESTABLISHMENT OF THE LANE TRAVERSAL PLANNING ALGORITHM FOR UNMANNED SWEEPERS

A. GENERATION OF NODE SEQUENCE AND COST MATRIX

This paper converts lanes into node sequences by obtaining the numbers of all n lanes in a specified map area from the structured road network, where one lane has only one direction of travel and one lane corresponds to a unique number, and reassigning each lane a corresponding node sequence number ranging from 1 to n. The lane number corresponds to the node sequence number so that the subsequently generated traversal node sequence can be converted to the corresponding lane, as shown in Fig. 3, and any two nodes in the structured road networks can reach each other, but the lengths of paths in forward and reverse directions may vary.

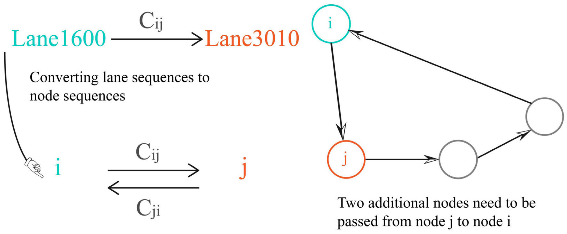

FIGURE 3. Schematic diagram of the conversion of lanes into the sequence of nodes. The diagram converts lanes 1600 and 3010 into nodes i and j. The path length Ci j between the two lanes is found by the A* algorithm.

After converting lane numbers to node sequence numbers, the A* algorithm searches for the shortest path between any two lane nodes in the structured road network and constructs an $n \times n$ cost matrix by using the distance of the shortest path between nodes as weights. For example, lane 1600 and lane 3010 correspond to nodes i, j, respectively, and the A* algorithm searches the distance between i, j as C_{ij}, where C_{ij} denotes the distance from node i to node j (i, j = 1,.., n), then the value of row i and column j of the cost matrix is C_{ij}.

B. SOLVE THE PRE-TRAVERSAL NODE SEQUENCE

The constructed cost matrix contains path distances between any two lane nodes. Subsequently, the pre-traversal node sequence can be searched based on the cost matrix. This sequence is required to traverse all lane nodes while minimizing the traversal cost. It is worth noting that the node sequence generated at this stage just traverses all lane nodes with minimum cost, but there is no direct passage between two neighboring nodes. This problem will be solved in the next section. The process of obtaining this sequence of nodes is the process of solving the ATSP, which can be formulated as follows: define a "cost matrix" $C = (C_{ij})$, where C_{ij} denotes the cost of traversing from node i to j (i, j = 1,.., n), i.e., the cost, and $C_{ij} \neq C_{ji}$. Find the cost of the arrangement of integers from 1 to n ($i_1, i_2, i_3, \ldots, i_n$), minimize the value:

$$C_{i_1 i_2} + C_{i_2 i_3} + \cdots + C_{i_n i_1} \tag{1}$$

It was pointed out above that the A* algorithm is suitable for solving the shortest path planning problem, so the A* algorithm is applied to solve the C_{ij} between all the lanes. Since this paper mainly considers the problem of traversing to sweep all the lanes with minimum cost during the sweeping work of unmanned sweepers, further improvements are made to the ATSP.

The ATSP is defined to find the minimum value of $C_{i_1 i_2} + C_{i_2 i_3} + \cdots + C_{i_n i_1}$, which defines that the ATSP needs to start from node i_1 and eventually return to node i_1, whereas in this paper, there is no expectation to return to the initial node, so a void node 0 is set up, and the cost between the void node and any node is 0. Define the improved ATSP as the cost matrix $C = (C_{ij})$, with C_{ij} denoting the path length from node i to node j (i, j = 0,1,..., n), i.e., the cost. where $C_{ij} \neq C_{ji}$. Essentially, this problem becomes solving for the arrangement of integers from 0 to n ($i_0, i_1, i_2, \ldots, i_n$) that

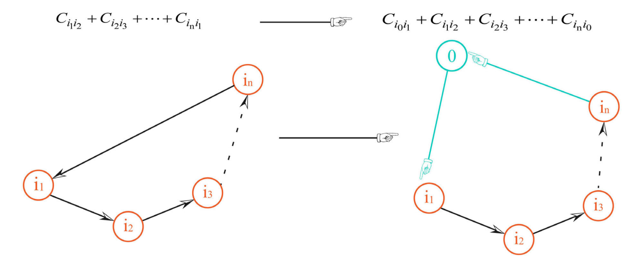

FIGURE 4. **Schematic diagram of adaptive improvements to the ATSP.**

minimizes the value:

$$C_{i_0i_1} + C_{i_1i_2} + C_{i_2i_3} + \cdots + C_{i_ni_0} \tag{2}$$

The node sequence derived from the original definition is i_1, i_2, i_3, …, i_n, i_1, and the improved ATSP node sequence solved is $i_0, i_1, i_2, \ldots, i_n, i_0$, as shown in Fig. 4, i_0 corresponds to void node 0, void node to i_1 and in to void node path cost are 0. The optimal path no longer needs to constitute a closed loop, which can reduce the total path length and the solving complexity, and improve the efficiency of traversal sweeping.

According to the improved ATSP formula, the cost matrix needs to add the cost between the void node and other nodes, and the cost is 0. At this time, the cost matrix becomes $n+1\times n+1$ dimensions. If lane 1600 and lane 3010 correspond to nodes i and j, respectively, and the distance between i and j is C_{ij}, the value of the j+1th column of the i+1th row of the cost matrix is C_{ij}. Entering the improved cost matrix into the LKH-3 solver, the pre-traversal node sequence can be derived.

C. GENERRATE THE TRAVERSAL NODE SEQUENCE

The solved pre-traversal node sequence can traverse all the nodes with minimum cost, but the neighboring nodes of this sequence are often not the precede and successor lane nodes for direct passage. Since the nodes solved by the LKH method do not consider the connectivity between nodes, i.e., lanes, it is necessary to add intermediate nodes between neighboring nodes to make the planned path passable. Therefore, the A* algorithm searches for the shortest passage path nodes between neighboring nodes, i.e., intermediate nodes, between each pre-traversal node. The intermediate nodes are then added to the pre-traversal node sequence to generate the traversal node sequence.

The entire process of generating the cost matrix, solving the pre-traversal node sequence, and adding the intermediate nodes to generate the traversal node sequence is shown in Fig. 5, which illustrates that the node sequence 1, 2, …, n is the initial lane node sequence, use the A* algorithm to search for the path cost between all lanes and adaptively improve the ATSP problem, add the void node 0 and generate the corresponding cost matrix $C=(C_{ij})$; transfer the cost matrix to the LKH-3 solver to generate the pre-traversal node sequence of i_0, i_1, …, i_n, i_0; use the A* algorithm to search for the intermediate nodes i_a, i_b of the neighboring nodes of the pre-traversal sequence and add them to the pre-traversal node sequence to generate the traversal node sequence.

D. PRUNING OPTIMIZATION

The pre-traversal node sequence generated already contains all lane nodes, and adding passable path to generate the traversal node sequence will inevitably result in repetition of lanes. At this point, it is necessary to perform pruning optimization to remove the redundant repeated lanes. Therefore, this subsection analyzes the problem of node redundancy and performs pruning optimization for the traversal node sequence.

Pruning I

In the structured road network, the same sweeping area at an intersection is covered by multiple lanes. When solving the traversal path planning, each lane that overlaps with each other is traversed, resulting in excessive redundancy of lane sweeping at intersections. Therefore, after converting the map into the structured road network, pruning I is performed to filter the overlapping lanes at intersections to improve efficiency.

According to Assumption 2, the unmanned sweeper only needs to sweep along the side of the outermost lanes, so in this paper, the masking operation is performed on the lanes close to the center point of the intersection, and the outermost lanes are not masked. The masked lanes are not involved in the computation only when solving the pre-traversal lane node sequence, but in the process of solving the shortest path with the A* algorithm, the masked lanes are involved in the same way as the normal lanes in order to find out the shortest path between the lanes.

Pruning II

The searched intermediate nodes between neighboring nodes may be duplicated with the pre-traversal nodes located after the neighboring nodes, resulting in redundancy. In this regard, the intermediate nodes are all labeled when solving the intermediate nodes, the duplicated nodes in the subsequent pre-traversal node sequence are found and filtered, and then the intermediate nodes continue to be solved. An example of pruning II is shown in Fig. 6, which illustrates the intermediate nodes i_a, i_b of the pre-traversal sequence nodes i_1 to i_2 overlap with the subsequent pre-traversal nodes, thus filtering the subsequent nodes i_a, i_b.

Pruning III

The intermediate nodes of the neighboring nodes may contain the initial non-0 nodes of the pre-traversed node sequence, at which point filter out the initial non-0 node. Then, the search for the next initial non-0 nodes is looped until the new initial non-0 node is no longer duplicated with the intermediate nodes. An example of pruning III is shown in Fig. 7, which illustrates the intermediate node i_1 of the pre-traversal node sequence i_c to i_d overlaps with the initial non-0 node i_1 of the pre-traversal sequence, so filter out the initial non-0 node i_1.

The traversal node sequence generated after pruning optimization is the optimal traversal node sequence. At this time, the optimal traversal node sequence is converted to the corresponding lane number to obtain the optimal traversal lane sequence.

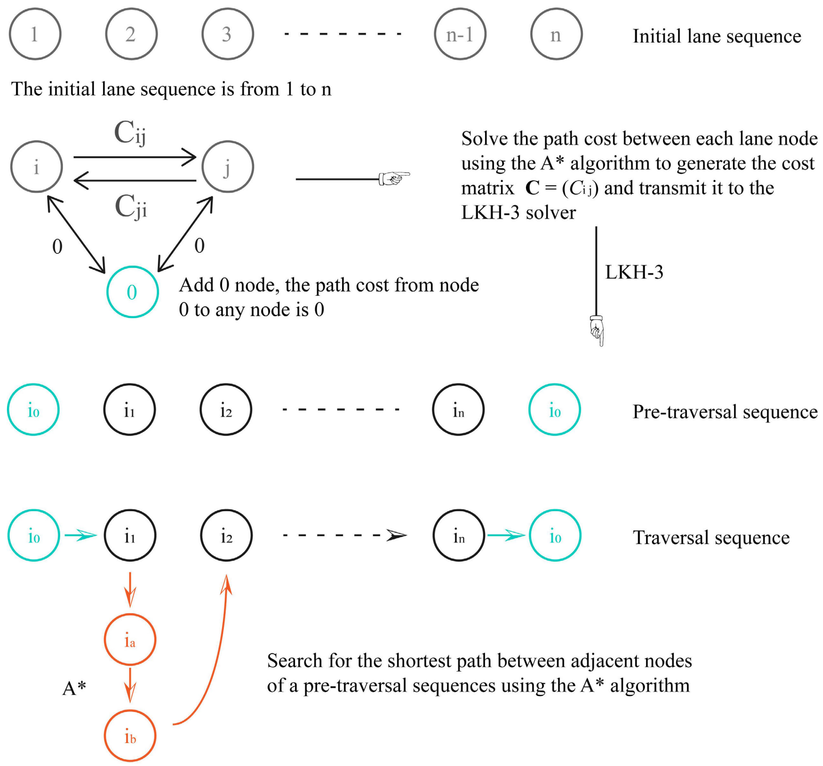

FIGURE 5. Schematic diagram of the traversal node sequence generation process.

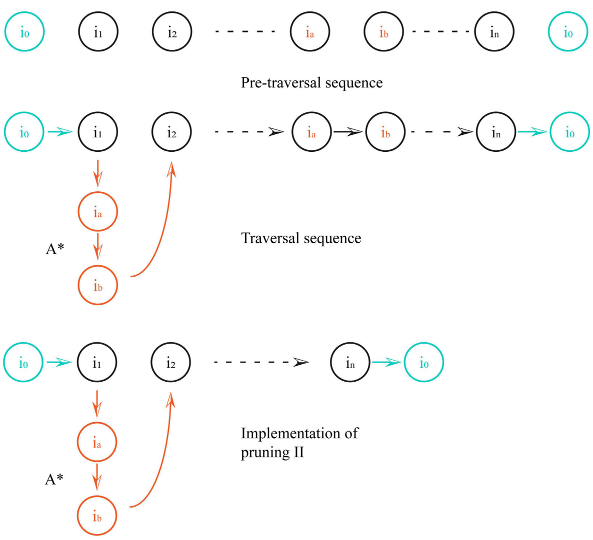

FIGURE 6. Schematic diagram of pruning II.

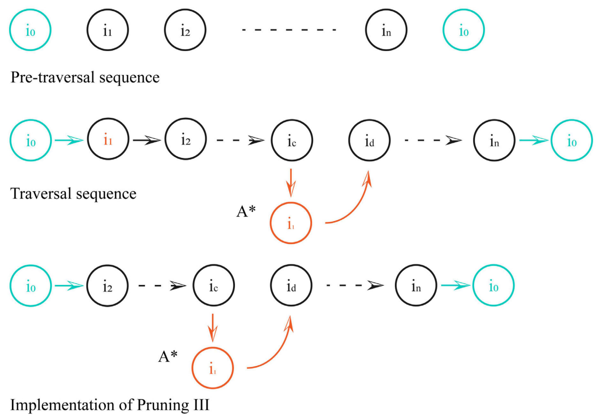

FIGURE 7. Schematic diagram of pruning III.

FIGURE 8. Map of structured roads in the select urban area.

E. OVERALL LOGICAL STRUCTURE OF THE METHODOLOGY

The logical structure of the methodology for solving the unmanned sweeper lane traversal path planning problem on structured roads is presented as follows:

- Read the converted structured road network scene file to obtain the lane information in the topological network and create the start and goal states for path planning.

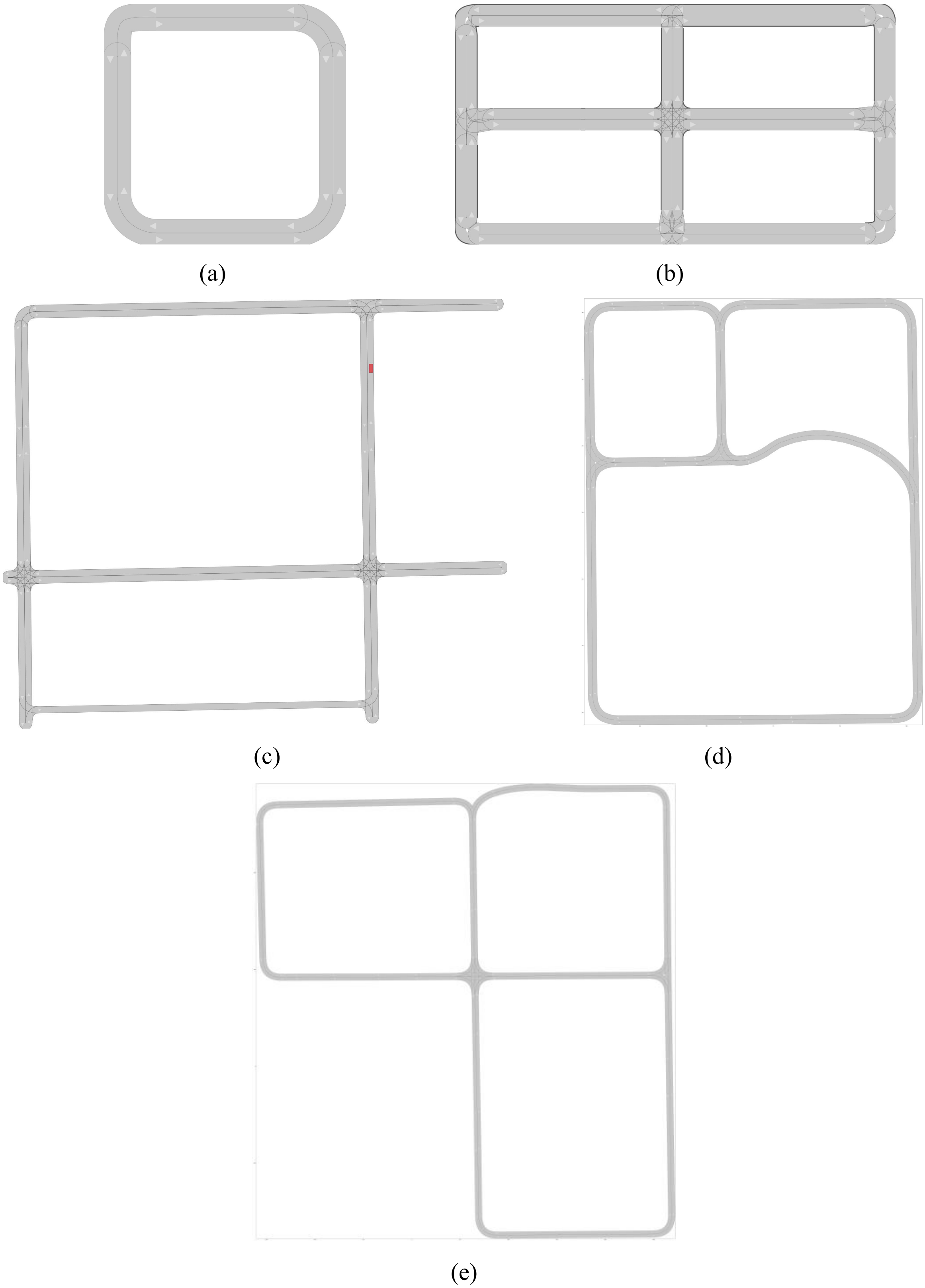

FIGURE 9. 9(a)-9(b) are basic structured road network maps, 9(c)-9(e) are real structured road network maps.

- Convert all lane numbers to node sequence numbers from 1 to n. Construct an index array <lanelet id, node id> of lane numbers and node sequence numbers that are converted to each other.

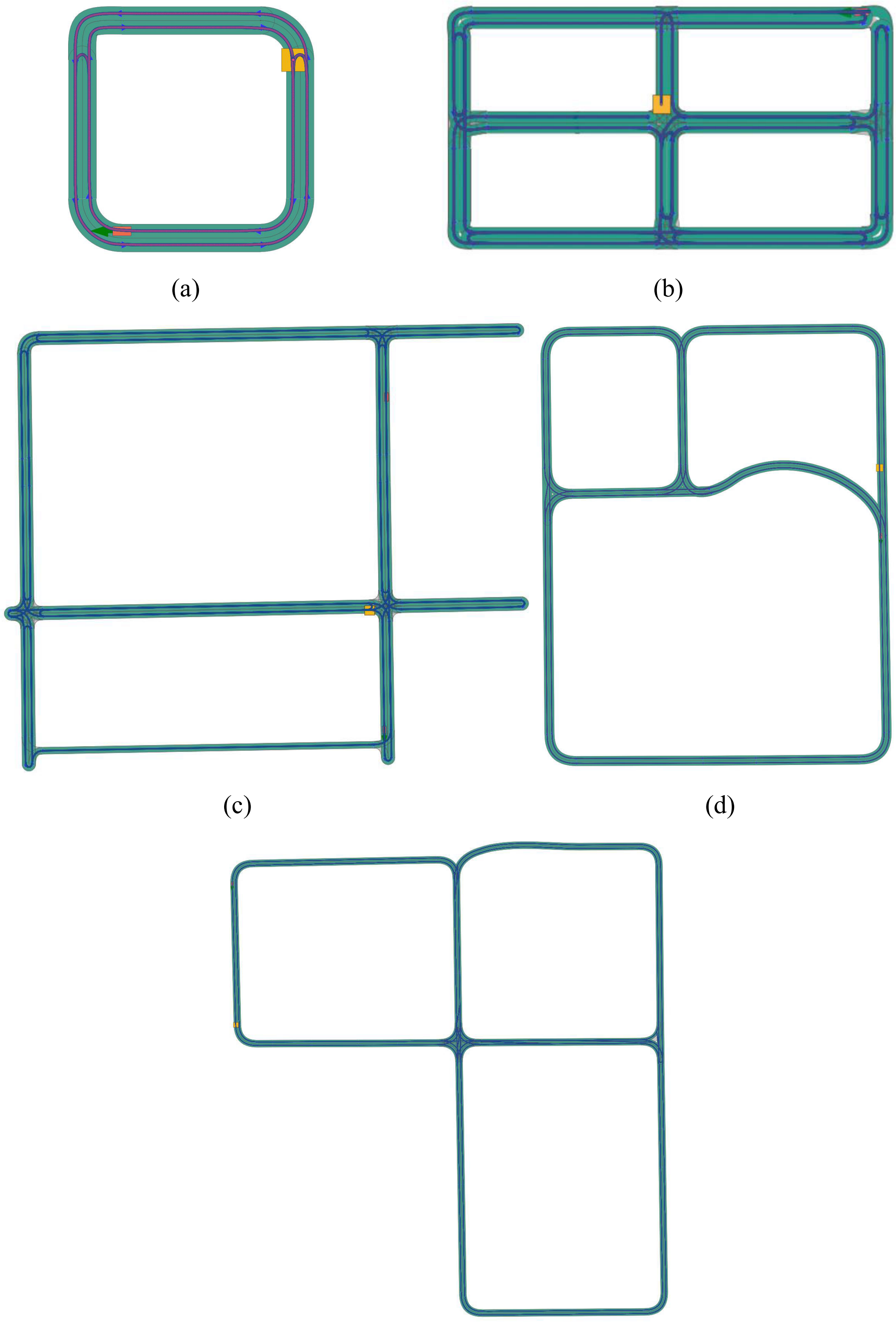

FIGURE 10. **Traversing the planning path visualization graph. The traversed lanes are shown in blue, and the purple line in the center represents the traversal path of the unmanned sweeper.**

- Traverse index <lanelet id, node id>, retrieve all lanes and their successor lane nodes. If the successor lanes are greater than or equal to 2, mask its successor lane nodes near the center point of the intersection to

TABLE 1. Traversal efficiency.

Experimental Scenario	*length* $_{lane}$(m)	*length* $_{traversal}$(m)	*PTE*
Scenario a	387	544	71.25%
Scenario b	1321	1786	73.99%
Scenario c	2490	3800	65.63%
Scenario d	3120	4752	65.65%
Scenario e	4550	6189	73.51%

filter the duplicate lanes at the intersection, complete pruning I.

- Traverse <lanelet id, node id>, after performing pruning I, to construct the cost matrix $C = (C_{ij})$.
    - Construct n+1 rank matrix, set void node, cost between void node and any node is 0.
    - Use the A* algorithm to solve for the shortest path distance between any two lane nodes as weights in the cost matrix. If lane 1600 and lane 3010 correspond to nodes i and j, respectively, and the distance between i and j is C_{ij}, then the value of the j+1th column of the i+1 row of the cost matrix is C_{ij}.
- Import the cost matrix into the LKH-3 solver to solve for the pre-traversal node sequence.
- Use the A* algorithm to search for the intermediate nodes of the neighboring nodes of the pre-traversal sequence, add them to the pre-traversal node sequence, and perform pruning II in the process of searching.
- Retrieve all lane nodes and perform pruning III to get the optimal traversal node sequence.
- Convert the optimal traversal node sequence to the optimal traversal lane sequence by the index array <lanelet id, node id>.
- Visualize the planning path after obtaining the optimal traversal lane sequence.

IV. EXPERIMENTS

A. EXPERIMENTAL SCENARIO CONSTRUCTION

This paper obtains the map vector data from the open-source website OpenStreetMap (OSM), selects two structured road maps near a busy road section in an urban area, as shown in Fig. 8, and uses the simulation environment CommonRoad [21] to transform them into the road networks with topological connectivity structures. Meanwhile, to validate the method's performance in the basic structured roads, two basic structured road network maps are manually created in CommonRoad for testing, as shown in Figs. 9(a)-9(b). Fig. 9(a) simulates a structured road around a single building with a total road length of 387 m. Fig. 9(b) simulates a structured road with multiple intersections around multiple buildings, with a total road length of 1,321 m. And the four real structured road network maps shown in Figs. 9(c)-9(e) are transformed from the different areas of the structured road maps shown in Fig. 9 through CommonRoad. The total road length of the four structured roads are 2490m, 3120m, and 4550m, respectively.

B. CALCULATION OF TRAVERSAL EFFICIENCY

In this paper, the unmanned sweeper lane traversal efficiency is defined as the sweeping path traversal efficiency, and the closer the traversal path length is to the total length of the sweeping area lanes, the lower the traversal sweeping duplication rate is, and the higher the traversal efficiency is. The path traversal efficiency is expressed through *PTE*, which refers to the ratio of the total length of the sweeping area lanes to the traversal path length.

$$PTE = \frac{length_{lane}}{length_{\mathrm{traversal}}} \tag{3}$$

where $length_{lane}$ is the total length of the sweeping area lanes of the structured roads, and $length_{traversal}$ is the traversal path length.

C. EXPERIMENTATION IN CONSTRUCTED SCENARIOS

This paper tests the proposed algorithm in each of the five scenarios from 9(a)-9(e) to verify its effectiveness. The experiments generate the optimal traversal lane sequences corresponding to the five scenarios, then visualize them and calculate the traversal efficiency based on the total length of the sweeping area lanes and the solved optimal traversal path length in the five test scenarios. The traversal planning path visualization of the experimental results is shown in Figs. 10(a)-10(e), and the traversal efficiency is shown in Table 1.

D. SUMMARY OF EXPERIMENTS

In this paper, map vector data of an urban area is obtained from OSM, and four real structured road network maps are constructed from the selected maps by the simulation environment CommonRoad. Meanwhile, two basic structured road network maps are created manually using CommonRoad to validate the algorithm's performance in the basic structured roads. The experiments were tested in each of the above five scenarios, and the traversal efficiency of the five scenarios in the experiments was around 70%. The traversal planning visualization results showed that the method realized the lane traversal planning for unmanned sweepers.

V. CONCLUSION

The lane traversal path planning algorithm proposed in this paper for unmanned sweepers treats lanes as nodes, solves the shortest path distances between all lane nodes of the structured roads using the A* algorithm, makes adaptive improvements to the ATSP problem and constructs the corresponding cost matrix, inputs the cost matrix to the LKH-3 solver to derive the pre-traversal node sequence, searches for intermediate nodes between the neighboring pre-traversal nodes using the A* algorithm and add them to the sequence to generate the traversal node sequence, and perform pruning optimization throughout the process to generate the optimal traversal node sequence and convert it to the corresponding lane sequence for visualization. The experimental verification of five scenarios proves that the algorithm can effectively and completely solve the unmanned sweeper lane traversal path planning problem on structured roads.

Future work could focus on the improvement of traversal efficiency of unmanned sweepers, trajectory planning and motion control of unmanned sweepers.

REFERENCES

[1] N. Hong, P. Zhu, A. Liu, X. Zhao, and Y. Guan, ''Using an innovative flag element ratio approach to tracking potential sources of heavy metals on urban road surfaces,'' *Environ. Pollut.*, vol. 243, pp. 410–417, Dec. 2018, doi: 10.1016/j.envpol.2018.08.098.

[2] L. Shi, L. Cheng, and L. Jiang, ''Research on campus small-scale unmanned driving cleaning technology,'' *Economist*, vol. 382, no. 12, pp. 220–221, 2020.

[3] China Industrial Information. *2019. Overview and Market Development Prospects of the Street Sweeper Industry in China*. Accessed: Dec. 19, 2019. [Online]. Available: http://www.chyxx.com/industry/201912/819662.html

[4] H. Zhang, W. Hong, and M. Chen, ''A path planning strategy for intelligent sweeping robots,'' in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 11–15, doi: 10.1109/ICMA.2019.8816519.

[5] M. F. Jensen, D. Bochtis, and C. G. Sørensen, ''Coverage planning for capacitated field operations, part II: Optimisation,'' *Biosyst. Eng.*, vol. 139, pp. 149–164, Nov. 2015, doi: 10.1016/j.biosystemseng.2015.07.002.

[6] Y. Zhang, Y. Wen, and H. Tu, ''A method for ship route planning fusing the ant colony algorithm and the A* search algorithm,'' *IEEE Access*, vol. 11, pp. 15109–15118, 2023, doi: 10.1109/ACCESS.2023.3243810.

[7] E. W. Dijkstra, ''A note on two problems in connexion with graphs,'' *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/bf01386390.

[8] P. Hart, N. Nilsson, and B. Raphael, ''A formal basis for the heuristic determination of minimum cost paths,'' *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/TSSC.1968.300136.

[9] B. Bonet and H. Geffner, ''Planning as heuristic search,'' *Artif. Intell.*, vol. 129, nos. 1–2, pp. 5–33, Jun. 2001, doi: 10.1016/s0004-3702(01)00108-4..

[10] J. F. Myoupo and A. C. Fabret, ''A modular systolic linearization of the warshall-floyd algorithm,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 7, no. 5, pp. 449–455, May 1996, doi: 10.1109/71.503769.

[11] D. Whitley, T. Starkweather, and D. Shaner, *The Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination*, 1991.

[12] N. Christofides and S. Eilon, ''Algorithms for large-scale travelling salesman problems,'' *J. Oper. Res. Soc.*, vol. 23, no. 4, pp. 511–518, Dec. 1972, doi: 10.2307/3007964.

[13] M. Grötschel and O. Holland, ''Solution of large-scale symmetric travelling salesman problems,'' *Math. Program.*, vol. 51, nos. 1–3, pp. 141–202, Jul. 1991, doi: 10.1007/bf01586932.

[14] S. Lin and B. Kernighan, ''An effective heuristic algorithm for the traveling-salesman problem,'' *Oper. Res.*, vol. 21, no. 2, pp. 498–516, Oct. 1973, doi: 10.1287/opre.21.2.498.

[15] K. Helsgaun, ''An effective implementation of the Lin–Kernighan traveling salesman heuristic,'' *Eur. J. Oper. Res.*, vol. 126, no. 1, pp. 106–130, Oct. 2000, doi: 10.1016/s0377-2217(99)00284-2.

[16] K. Helsgaun, ''General K-opt submoves for the Lin–Kernighan TSP heuristic,'' *Math. Program. Comput.*, vol. 1, nos. 2–3, pp. 119–163, Oct. 2009, doi: 10.1007/s12532-009-0004-6.

[17] R. Jonker and T. Volgenant, ''Transforming asymmetric into symmetric traveling salesman problems,'' *Oper. Res. Lett.*, vol. 2, no. 4, pp. 161–163, Nov. 1983, doi: 10.1016/0167-6377(83)90048-2.

[18] K. Helsgaun, ''An extension of the Lin–Kernighan–Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems,'' Tech. Rep., Dec. 2017.

[19] Á. Corberán, R. Eglese, G. Hasle, I. Plana, and J. M. Sanchis, ''Arc routing problems: A review of the past, present, and future,'' *Networks*, vol. 77, no. 1, pp. 88–115, Jan. 2021, doi: 10.1002/net.21965.

[20] J. Edmonds and E. L. Johnson, ''Matching: A well-solved class of integer linear programs,'' in *Combinatorial Optimization—Eureka, You Shrink!*, pp. 27–30, 2003, doi: 10.1007/3-540-36478-1_3.

[21] P. Bender, J. Ziegler, and C. Stiller, ''Lanelets: Efficient map representation for autonomous driving,'' in *Proc. IEEE Intell. Vehicles Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 420–425, doi: 10.1109/IVS.2014.6856487.

[22] S. Maierhofer, M. Klischat, and M. Althoff, ''CommonRoad scenario designer: An open-source toolbox for map conversion and scenario creation for autonomous vehicles,'' in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3176–3182, doi: 10.1109/ITSC48978.2021.9564885.

LEI HE received the Ph.D. degree from Jilin University, Changchun, China, in 2011. He is currently a Professor with the College of Automotive Engineering, Jilin University. He is responsible for several state-funded intelligent driving modeling and simulation government projects. He is the author or coauthor of numerous publications in the field of environmental awareness and vehicle control.

BAOYUN WANG is currently pursuing the master's degree in vehicle engineering with the College of Automotive Engineering, Jilin University. His research interests include intelligent vehicle path planning and vehicle trajectory prediction.

YUNSHAN PENG received the master's degree from Jilin University, in 2023. He is currently a Decision Planning Algorithm Engineer with an autonomous driving company. He has been involved in research related to intelligent driving decision planning.

XIUCAI ZHANG is currently pursuing the M.S. degree in automotive engineering with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, China. His research interest includes V2X infrastructure-side multi-target detection.